www.mientayvn.com

Khi đọc qua tài liệu này, nếu phát hiện sai sót hoặc nội dung kém chất lượng xin hãy thông báo để chúng tôi sửa chữa hoặc thay thế bằng một tài liệu cùng chủ đề của tác giả khác. Tài liệu này bao gồm nhiều tài liệu nhỏ có cùng chủ đề bên trong nó. Phần nội dung bạn cần có thể nằm ở giữa hoặc ở cuối tài liệu này, hãy sử dụng chức năng Search để tìm chúng.

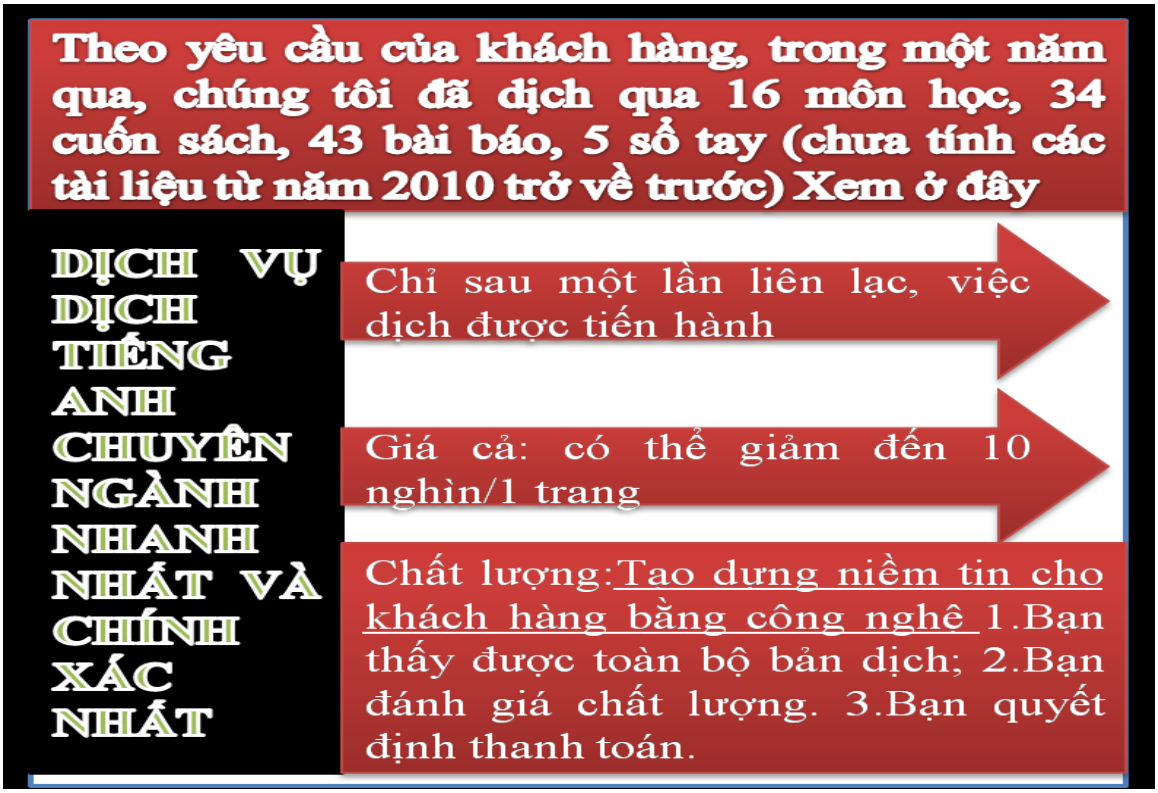
Bạn có thể tham khảo nguồn tài liệu được dịch từ tiếng Anh tại đây:

http://mientayvn.com/Tai_lieu_da_dich.html

Thông tin liên hệ:

Yahoo mail: thanhlam1910_2006@yahoo.com

Gmail: frbwrthes@gmail.com



Giáo trình
Tự học lập trình PHP

Môn học: PHP

Bài 1

Những vấn đề chính sẽ được đề cập trong bài học:

- ✓ *Giới thiệu PHP*
- ✓ *Cấu hình IIS, Apache Web Server*
- ✓ *Cài đặt PHP.*
  - o *Cài đặt PHP.*
  - o *Cấu hình ứng dụng PHP*
- ✓ *Giới thiệu PHP.*
  - o *PHP Script.*
  - o *Ghi chú trong PHP*
  - o *In nội dung bằng PHP*

1. GIỚI THIỆU PHP

PHP viết tắt của chữ Personal Home Page ra đời năm 1994 do phát minh của Rasmus Lerdorf, và nó tiếp tục được phát triển bởi nhiều cá nhân và tập thể khác, do đó PHP được xem như một sản phẩm của mã nguồn mở.

PHP là kịch bản trình chủ (server script) chạy trên phía server (server side) như cách server script khác (asp, jsp, cold fusion).

PHP là kịch bản cho phép chúng ta xây dựng ứng dụng web trên mạng internet hay intranet tương tác với mọi cơ sở dữ liệu như mySQL, PostgreSQL, Oracle, SQL Server và Access.

Lưu ý rằng, từ phiên bản 4.0 trở về sau mới hỗ trợ session, ngoài ra PHP cũng như Perl là kịch bảng xử lý chuỗi rất mạnh chính vì vậy bạn có thể sử dụng PHP trong những có yêu cầu về xử lý chuỗi.

2. CÀI ĐẶT PHP

Cài đặt PHP trên nền Windows thì sử dụng php-4.0.6-Win32.zip, sau khi cài đặt ứng dụng này trên đĩa cứng sẽ xuất hiện thư mục PHP, trong thự mục này sẽ có tập tin php4ts.dll và php.exe cùng với thư mục sessiondata.

Ngoài ra, trong thư mục WINDOW hoặc WINNT sẽ xuất hiện tập tin php.ini, tập tin này cho phép bạn cấu hình cho ứng dụng PHP. Chẳng hạn, khi sử dụng session, PHP cần một nơi để lưu trữ chúng, trong tập tin này mặc định là session.save_path = C:\PHP\sessiondata, nếu bạn cài đặt PHP với thư mục PHP trên đĩa D thì bạn cần thay đổi đường dẫn trong khai báo này.

Tương tự như vậy, khi có lỗi trong trangPHP thì lỗi thường xuất hiện khi triệu gọi chúng, để che dấu các lỗi này thì bạn cần khai báo display_errors = Off thay vì chúng ở trạng thái display_errors = On.

Ngoài ra, trang PHP cũng có thể trình bày một số warning khi chúng phát hiện cú pháp không hợp lý, chính vì vậy để che dấu các warning này thì bạn cũng cần khai báo trạng thái Off thay vì On như assert.warning = Off.

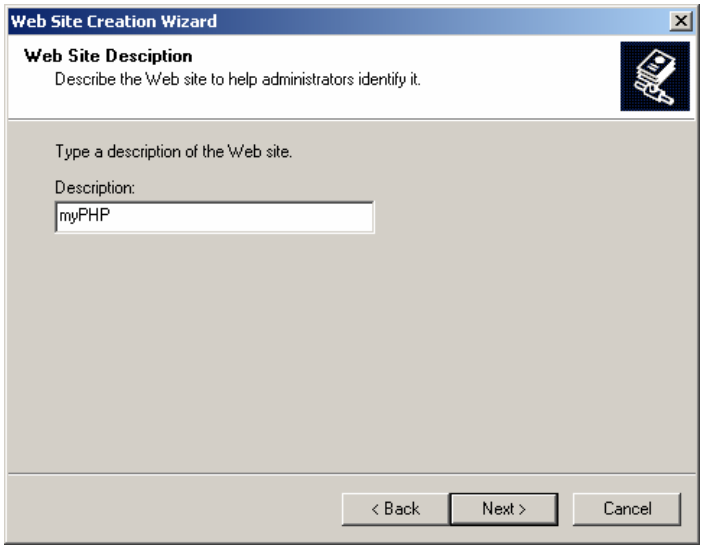
3. CẤU HÌNH ỨNG DUNG PHP

3.1. Cấu hình IIS

Sau khi cài đặt hệ điều hành Windows NT hay 2000 trở về sau, bằng cách khai báo mới một web site hay virtual site trong một site đang có theo các bước như sau:

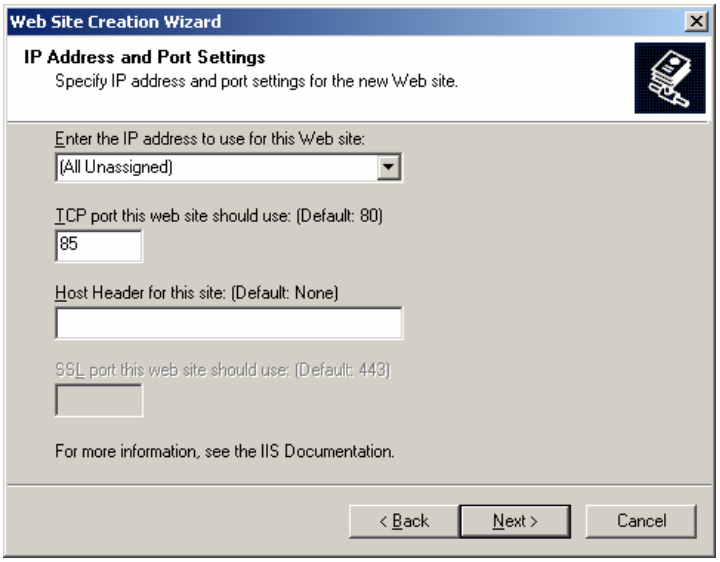
1. Tạo một thư mục có tên myPHP để lưu trữ các tập tin PHP
2. Khởi động IIS (tự động khởi động nếu Windows NT/2000)

3. Chọn Start | Programs | Administrative Tools | Internet Information Server
4. Nếu tạo virtual site thì chọn Default Web Ste | R-Click | New | Virtual Site
5. Trong trường hợp tạo mới Site thì Default Web Ste | R-Click | New | Site
6. Nếu chọn trường hợp 4 thì bạn cung tấp diễn giải của site như hình 1-1



Hình 1-1: Khai báo diễn giải

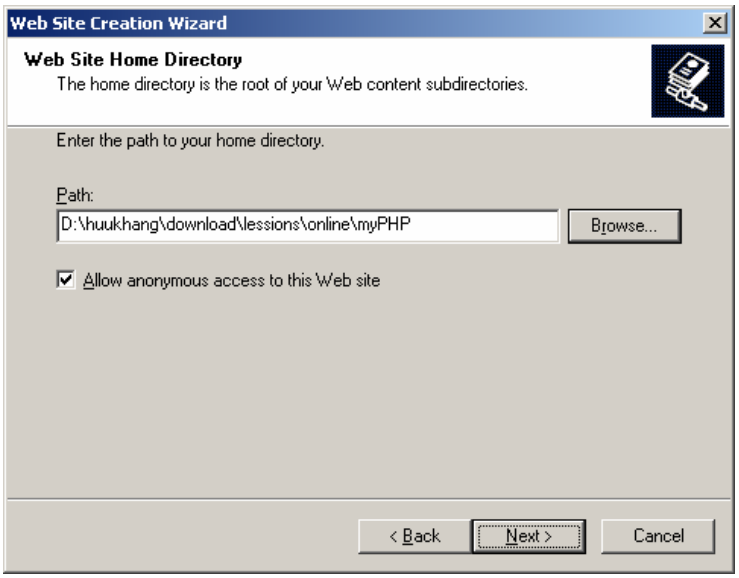
7. Chọn nút Next và khai báo IP và port, trong trường hợp bạn không sử dụng port 80 cho ứng site khác thì chọn giá trị mặc định. Tuy nhiên nếu có nhiều ứng dụng trước đó đã cấu hình trong IIS thì bạn có thể thay đổi port khác, ví dụ chọn port 85 như hình 1-2.



Hình 1-2: Khai báo IP và Port

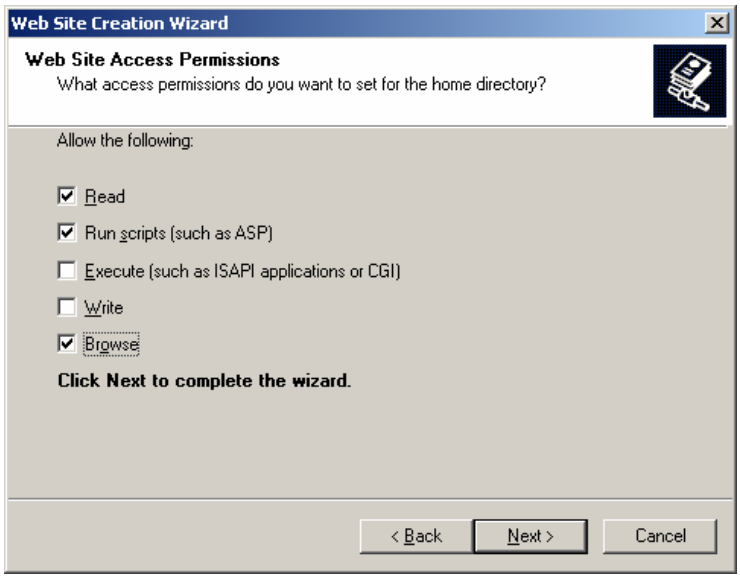
Lưu ý rằng, port 80 là port chuẩn điều này có nghĩa là khi triệu gọi trên trình duyệt bạn không cần gõ port, ví dụ http://localhost/. Đối với trường hợp port khác thì bạn phải gõ tương tự như http://localhost:85/

8. Chọn Next, bạn chọn thư mục của ứng dụng, đối với trường hợp này chúng ta chọn vào thư mục myPHP, chẳng hạn trong trường hợp này chúng ta chọn htư mục myPHP như hình 1-3.



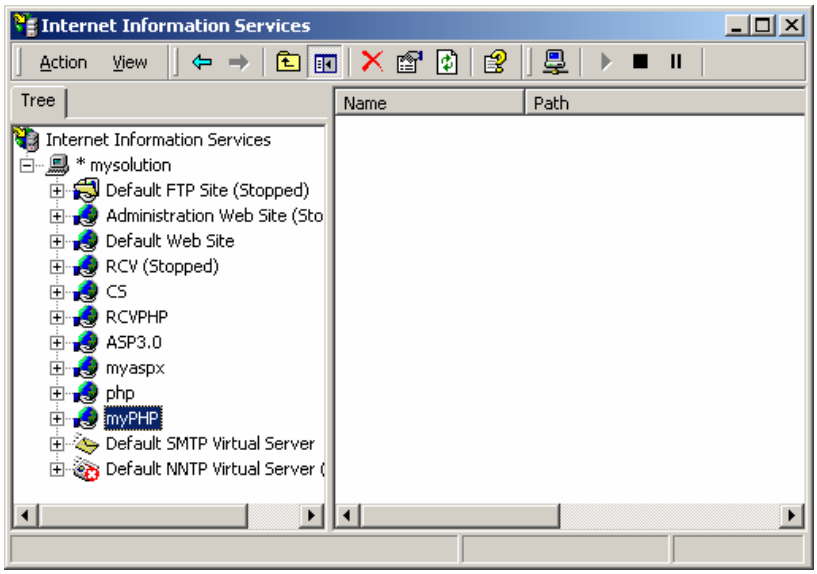
Hình 1-3: Chọn thư mục myPHP

9. Kế đến chọn quyền truy cập web site, trong trường hợp đang thiết kế thì bạn chọn vào Browse. Ngoài ra, nếu bạn cho phép người sử dụng internet có thể thực thi tập tin thực thi từ xa thì chọn vào tuỳ chọn execute.



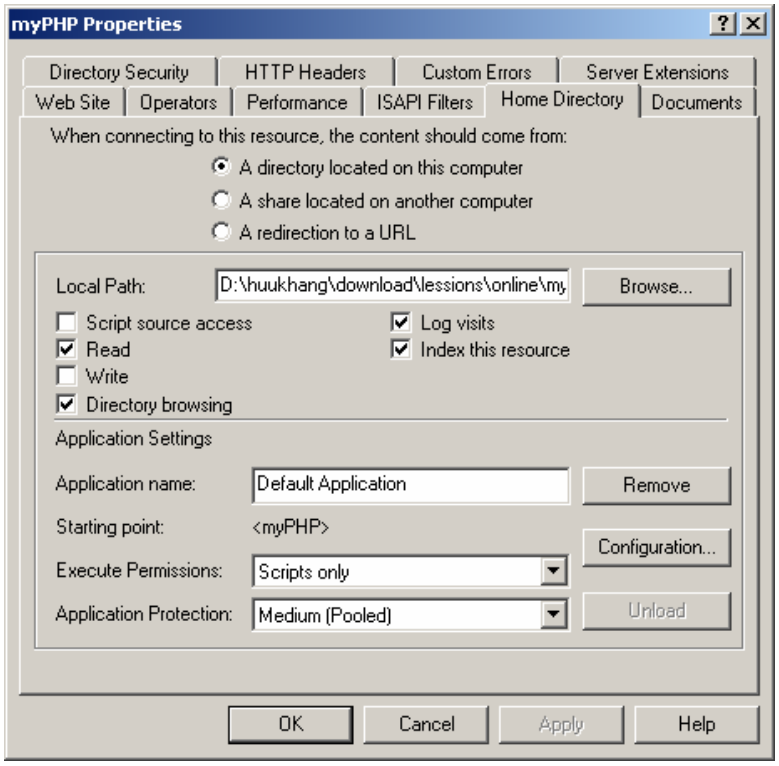
Hình 1-4: Quyền truy cập

10. Chọn Next và Finish, trong cửa sổ IIS xuất hiện ứng dụng có tên myPHP (khai báo trong phần diễn giải) như hình 1-5.



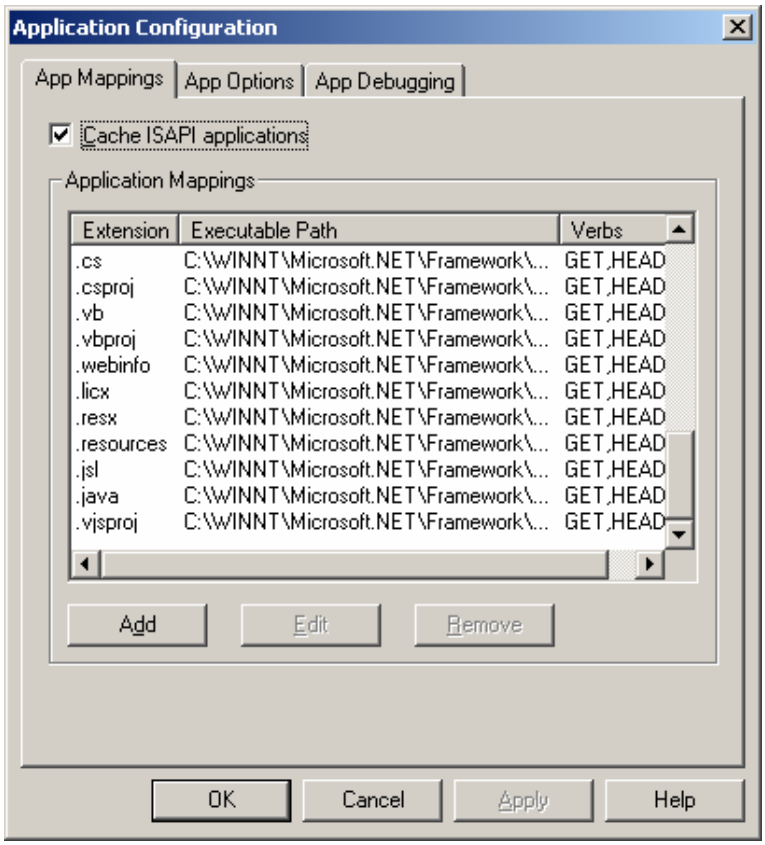
Hình 1-5: Tạo thành công ứng dụng PHP trong IIS

11.Sau khi tạo ứng dụng xong, bạn chọn tên ứng dụng myPHP | R-Click } Properties | cửa sổ xuất hiện như hình 1-5.



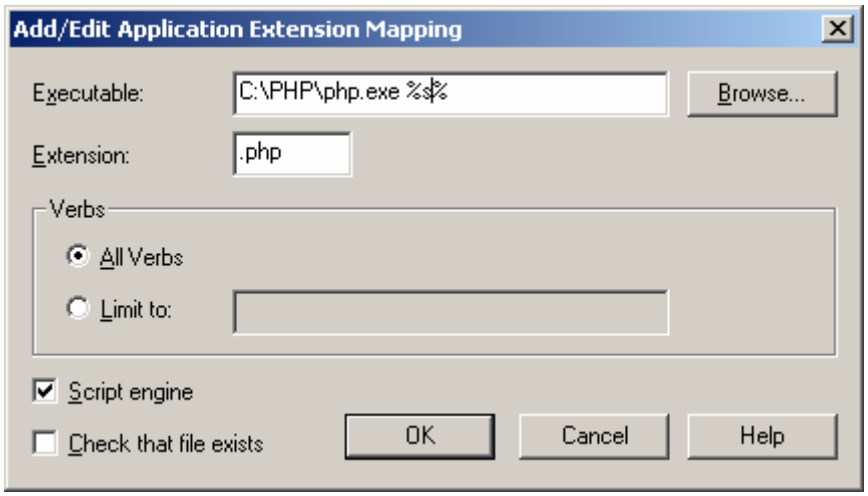
Hình 1-5: Cấu hình PHP trong IIS

12.Bằng cách chọn vào nút Configuration, cửa sổ sẽ xuất hiện như hình 1-6.



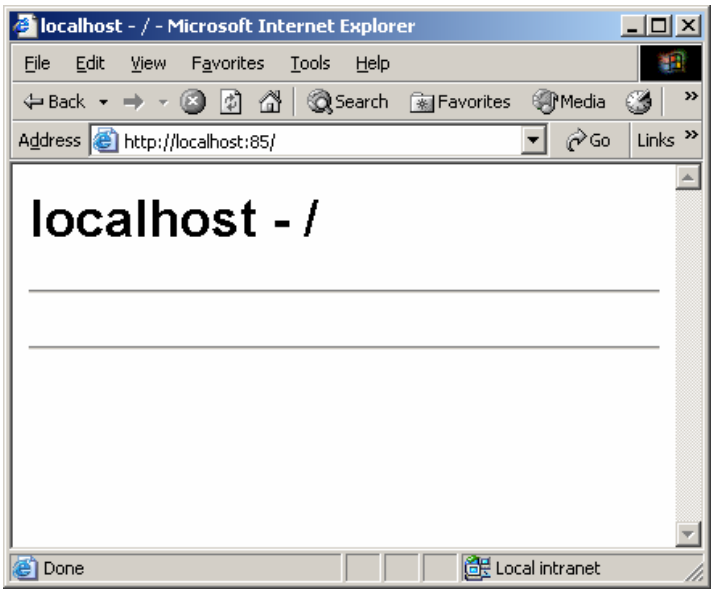
Hình 1-6: Thêm PHP Engine

13.Chọn nút Add, và khai báo như hình 1-7.



Hình 1-7: Khai báo PHP Engine

14.Để kiểm tra úng dụng, bạn mở cửa sổ IE và gõ trên thanh địa chỉ chuỗi như sau: http://localhost:85/ , kết quả xuất hiện như hình 1-8.



Hình 1-8: Ứng dụng PHP đã được khởi động

3.2. Cài đặt Apache Web Servr

Để cài đặt Apache Web Server, bạn theo các bước sau

1. Chep tap tin apache_1.3.22-win32-x86.exe xuong dia cung
2. Chay tap tin nay va cai dat len dia C:\Program Files\, sau khi ket thuc thanh cong phan cai dat Apache, bạn bắt đầu cấu hình ứng dụng PHP.
3. Chép ba dòng lệnh từ tập tin install.txt trong thư mục C:\PHP

```
ScriptAlias /php/ "c:/php/"
AddType application/x-httpd-php .php
Action application/x-httpd-php "/php/php.exe"
```

4. Paste vào tập tin httpd.conf trong thư mục C:\Program Files\Apache Group\Apache\Conf\
5. Chon Start | Programs | Apache HTTP Server | Control Apache Server | Start
6. Viet trang test.php voi noi dung <?echo "hello";?>
7. Chep tap tin test.php vao thu muc C:\Program Files\Apache Group\Apache\htdocs\
8. Sau đó gõ trên trình duyệt http://localhost/test.php

4. GIỚI THIỆU PHP

4.1. Yêu cầu

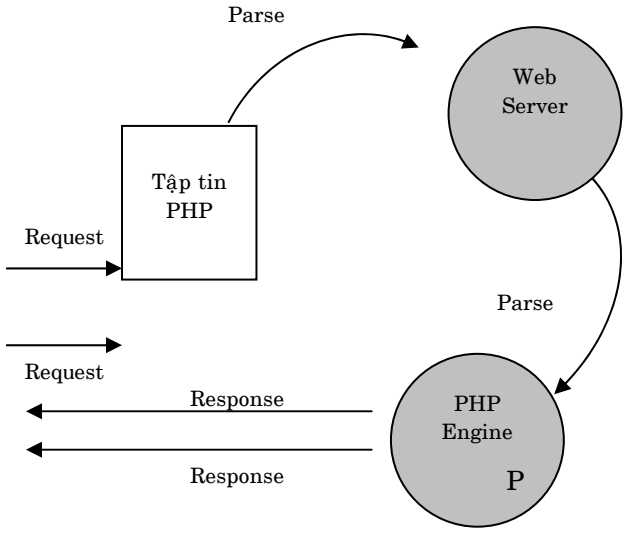
PHP dựa trên cú pháp của ngôn ngữ lập trình C, chính vì vậy khi làm việc với PHP bạn phải là người có kiến thức về ngôn ngữ C, C++, Visual C. Nếu bạn xây dựng ứng dụng PHP có kết nối cơ sở dữ liệu thì kiến thức về cơ sở dữ liệu MySQL, SQL Server hay Oracle là điều cần thiết.

4.2. Giới thiệu

PHP là kịch bản trình chủ (Server Script) được chạy trên nền PHP Engine, cùng với ứng dụng Web Server để quản lý chúng. Web Server thường sử dụng là IIS, Apache Web Server, ...

4.3. Thông dịch trang PHP

Khi người sử dụng gọi trang *PHP*, *Web Server* triệu gọi PHP Engine để thông dịch (tương tự như ASP 3.0 chỉ thông dịch chứ không phải biên dịch) dịch trang *PHP* và trả về kết quả cho người sử dụng như hình 1-9.



Hình 1-9: Quá trình thông dịch trang PHP

4.4. Kịch bản (script)

Nội dung của PHP có thể khai báo lẫn lộn với HTML, chính vì vậy bạn sử dụng cặp dấu giá <?=trị/biểu thức/biến?> để khai báo mã PHP. Chẳng hạn, chúng ta khai báo:

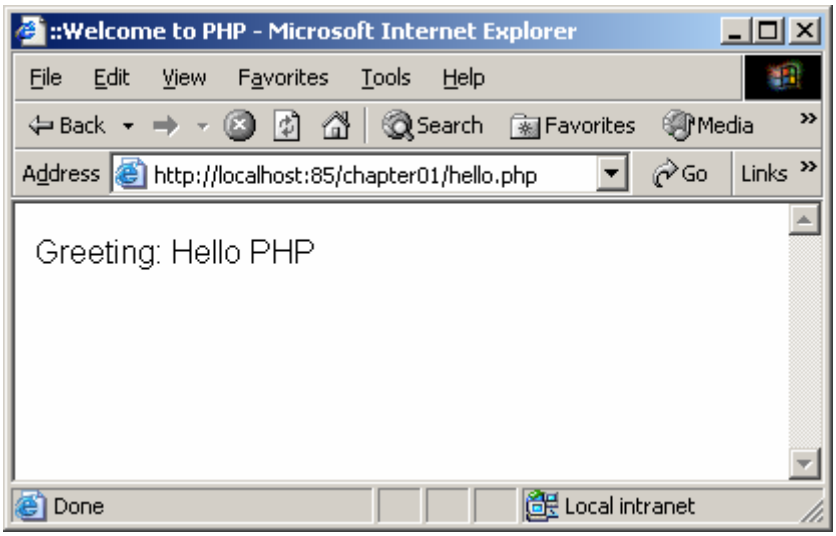
```
<br>
1-Giá trị biến Str: <?=$groupid?>
2-Giá trị biến i: <?=$i?>
3-Giá trị cũ thể: <?=10?>
```

Chẳng hạn bạn khai báo trang hello.php với nội dung như ví dụ 1-1 sau:

Ví dụ 1-1: Trang hello.php

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
  Greeting: <?="Hello PHP"?>
</BODY>
</HTML>
```

Kết quả trả về như hình 1-10 khi triệu gọi trang này trên trình duyệt.



Hình 1-10: Kết quả trang hello.php

Trong trường hợp có nhiều khai báo, bạn sử dụng Scriptlet, đều này có nghĩa là sử dụng cặp dấu trên như <?php Khai báo ?> với các khai báo PHP với cú pháp của C như sau:

```
<?php
     $sotrang=$pagenumber;
     $record=$rownumber;
     $totalRows = 0;
     $paging="";
?>
-Khai báo trên là Scriptlet
Giá trị của paging: <br>
<?= $paging ?>
-Khai báo này là Script
```

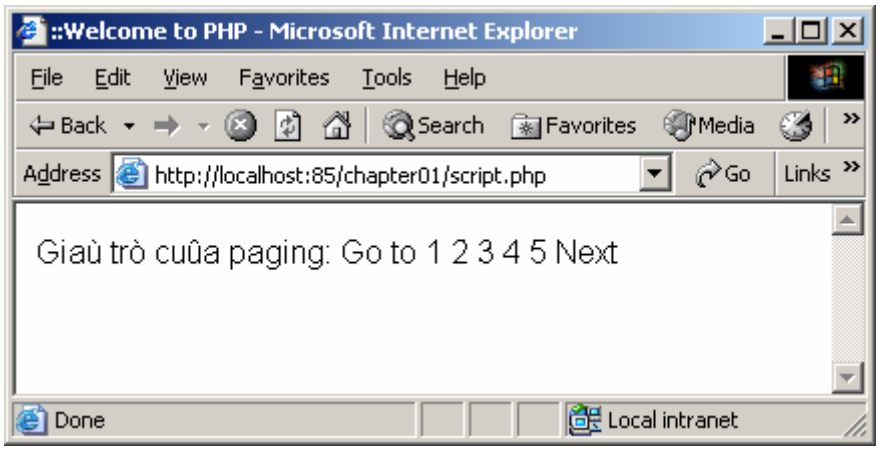
Lưu ý rằng, kết thúc mỗi câu lệnh phải dùng dấu ;

Ví dụ, bạn khai báo đoạn PHP trên trong tập tin script.php như ví dụ 1-2

Ví dụ 1-2: Trang script.php

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
     <?php
     $sotrang=$pagenumber;
     $record=$rownumber;
     $totalRows = 0;
     $paging="Go to 1 2 3 4 5 Next";
     ?>
     Giá trị của paging: <?= $paging ?>
</BODY>
</HTML>
```

Kết quả trả về như hình 1-11 khi triệu gọi trang này trên trình duyệt.



Hình 1-11: Kết quả trang hello.php

Lưu ý rằng, nếu bạn muốn sử dụng script hay scriptlet như ASP thì bạn khai báo trong tập tin php.ini như sau:

```
asp_tags = On
; Allow ASP-style <% %> tags. mặc định là Off
```

Khi đó trong trang PHP, thay vì bạn khai báo

```
<?php
   $sotrang=$pagenumber;
   $record=$rownumber;
   $totalRows = 0;
   $paging="";
?>
```

Thì bạn có thể khai báo như sau:

```
<%
   $sotrang=$pagenumber;
   $record=$rownumber;
   $totalRows = 0;
   $paging="";
%>
```

4.5. Ghi chú trong PHP

Ghi chú trong kịch bản PHP tương tự ngôn ngữ lập trình C, để ghi chú một dòng thì bạn sử dụng cặp dấu /. Chẳng hạn khai báo sau là ghi chú:

```
<?php
      // Khai báo biến để paging
      $sotrang=$pagenumber;
      $record=$rownumber;
      $totalRows = 0;
      $paging="";
?>
```

Trong trường hợp có nhiều dòng cần ghi chú bạn sử dụng cặp dấu /* và */, ví dụ khai báo ghi chú như sau:

```
/*
Khai báo biến để đọc dữ liệu
trong đó totalRows là biến trả
về tổng số mẩu tin
*/
$result = mysql_query($stSQL, $link);
$totalRows=mysql_num_rows($result);
```

Ngoài ra, bạn cũng có thể sử dụng dấu # để khai báo ghi chú cho từng dòng, ví dụ khai báo sau là ghi chú:

```
<?php
     # Khai báo biến để paging
     $sotrang=$pagenumber;
     $record=$rownumber;
     $totalRows = 0;
     $paging="";
?>
```

4.6. In kết quả trên trang PHP

Khác vớ các kịch bản như ASP, JSP, Perl, đối với PHP để in ra giá trị từ biến, biểu thức, hàm, giá trị cụ thể thỉ bạn có thể sử dụng script như trên:

```
Giá trị của paging: <%= $paging %>
```

Tuy nhiên, để sử dụng cú pháp của PHP khi in ra giá trị từ biến, biểu thức, hàm, giá trị cụ thể thì sử dụng khai báo echo như sau:

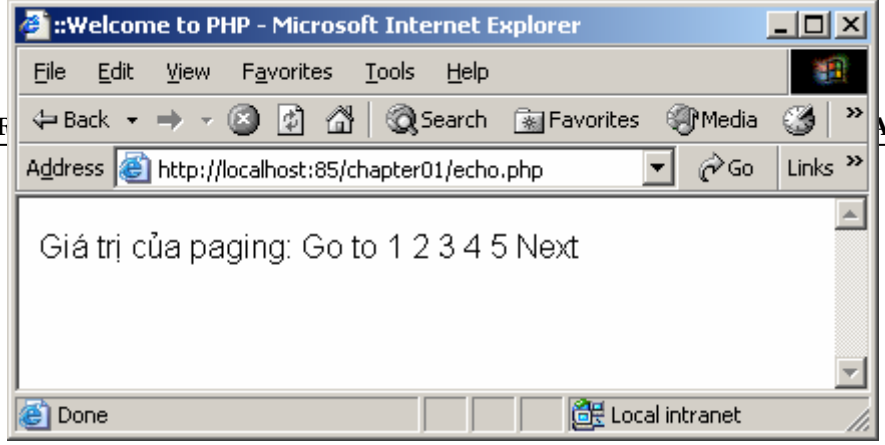
```
<?php
            $stSQLs=”select * from Customers”;
            echo $stSQLs;
?>
```

Chẳng hạn, khai báo echo như ví dụ 1-3.

Ví dụ 1-2: Trang echo.php

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
    <?php
    $sotrang=$pagenumber;
    $record=$rownumber;
    $totalRows = 0;
    $paging="Go to 1 2 3 4 5 Next";
    /*dùng phát biểu echo */
    echo “Giá trị của paging: “;
    echo $paging;
    ?>
</BODY>
</HTML>
```

Kết quả trả về như hình 1-12 khi triệu gọi trang này trên trình duyệt.



Hình 1-11: Kết quả trang hello.php

5. KẾT LUẬN

Trong bài này, chúng ta tập trung tìm hiểu cách cài đặt PHP và Apache Web Server, sau đó cấu hình ứng dụng PHP trong IIS hay sử dụng cấu hình mặc định của chúng.

Ngoài ra, bạn làm quen cách khai báo mã PHP trong trang .php cùng với script hay scriptlet.

Môn học: PHP

Bài 2

Bài học này chúng ta sẽ làm quen và tìm hiểu cú pháp và một số phương thức cơ bản của PHP:

- ✓ *Câu lệnh.*
- ✓ *Kiểu dữ liệu và biến*
- ✓ *Khai báo và sử dụng hằng.*
- ✓ *Dữ liệu mảng*
- ✓ *Chuyển đổi kiểu dữ liệu*

1. KHÁI NIỆM VỀ CÚ PHÁP PHP

Cú pháp PHP chính là cú pháp trong ngôn ngữ C, các bạn làm quen với ngôn ngữ C thì có lợi thế trong lập trình PHP.

Để lập trình bằng ngôn ngữ PHP cần chú ý những điểm sau:

- ❖ Cuối câu lệnh có dấu ;
- ❖ Biến trong PHP có tiền tố là $
- ❖ Mỗi phương thức đều bắt đầu { và đóng bằng dấu }
- ❖ Khi khai báo biến thì không có kiểu dữ liệu
- ❖ Nên có giá trị khởi đầu cho biến khai báo
- ❖ Phải có chi chú (comment) cho mỗi feature mới
- ❖ Sử dụng dấu // hoặc # để giải thích cho mỗi câu ghi chú
- ❖ Sử dụng /* và */ cho mỗi đoạn ghi chú
- ❖ Khai báo biến có phân biệt chữ hoa hay thường

2. KHAI BÁO BIẾN

Khi thực hiện khai báo biến trong C, bạn cần phải biết tuân thủ quy định như: kiểu dữ liệu trước tên biến và có giá trị khởi đầu, tuy nhiên khi làm việc với PHP thì không cần khai báo kiểu dữ liệu nhưng sử dụng tiền tố $ trước biến.

Xuất phát từ những điều ở trên, khai báo biến trong PHP như sau:

- ❖ $variablename [=initial value];

```
$licount=0;
$lsSQL="Select * from tblusers where active=1";
$nameTypes  = array("first", "last", "company");
$checkerror=false;
```

- ❖ Chẳng hạn, khai báo như ví dụ 2-1 (variables.php)

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
```

```
<BODY>
<h4>Variable</h4>
<?php
   $sotrang=10;
   $record=5;
   $check = true;
   $strSQL="select * from tblCustomers";
   $myarr = array("first", "last", "company");
   $myarrs[2];
   $myarrs[0]="Number 0";
   $myarrs[1]="Number 1";
   $myarrs[2]="Number 2";
   echo $myarr[1];echo "<br>";
   echo $myarrs[2];
?>
</BODY>
</HTML>
```

3. KIỂU DỮ LIỆU

Bảng các kiểu dữ liệu thông thường

Boolean	True hay false
Integer	giá trị lớn nhất xấp xỉ 2 tỷ
Float	~1.8e308 gồm 14 số lẽ
String	Lưu chuỗi ký tự chiều dài vô hạn
Object	Kiểu đối tượng
Array	Mảng với nhiều kiểu dữ liệu

3.1. Thay đổi kiểu dữ liệu

Để thay đổi kiểu dữ liệu, bạn có thể sử dụng cách ép kiểu như trong các ngôn ngữ lập trình C hay Java. Chẳng hạn, khai báo ép kiểu như ví dụ 2-2 (box.php):

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Variable</h4>
<?php
     $i="S10A";
     echo $i+10;
     echo "<br>";
     $i="10A";
     $j=(float)$i;
     $j+=10;
     echo $i;
     echo "<br>";
     echo $j;
     echo "<br>";
     $q=12;$p=5;
     echo "Amount: ".(float)$q/$p;
?>
</BODY>
</HTML>
```

Lưu ý rằng, PHP tự động nhận biết giá trị chuỗi đằng sau số sẽ không được chuyển sang kiểu dữ liệu số như trường hợp trên.

Ngoài ra, bạn có thể sử dụng hàm settype để chuyển đổi dữ liệu này sag dữ liệu khác, ví dụ chúng ta khai báo như ví dụ 2-3 (settype.php).

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Change DataType of Variable</h4>
<?php
     $var="12-ABC";
     $check=true;
     echo $var;
     echo "<br>";
     echo $check;
     echo "<br>";
     settype($var,"integer");
     echo $var;
     echo "<br>";
     settype($check,"string");
     echo $check;
?>
</BODY>
</HTML>
```

3.2. Kiểm tra kiểu dữ liệu của biến

Để kiểm tra kiểu dữ liệu của biến, bạn sử dụng các hàm như sau:

is_int để kiểm tra biến có kiểu integer, nếu biến có kiểu integer thì hàm sẽ trả về giá trị là true (1). Tương tự, bạn có thể sử dụng các hàm kiểm tra tương ứng với kiểu dữ liệu là is_array, is_bool, is_callable, is_double, is_float, is_int, is_integer, is_long, is_null, is_numeric, is_object, is_real, is_string. Chẳng hạn, bạn khai báo các hàm này như ví dụ 2-4 (check.php).

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Check DataType of Variable</h4>
<?php
     $sotrang=10;
     $record=5;
     $check = true;
     $strSQL="select * from tblCustomers";
     $myarr = array("first", "last", "company");
     $myarrs[2];
     $myarrs[0]="Number 0";
     $myarrs[1]="Number 1";
     $myarrs[2]="Number 2";
     echo is_array($myarr);
     echo "<br>";
     echo is_bool($record);
?>
</BODY>
```

```
</HTML>
```

3.3. Thay đổi kiểu dữ liệu biến

Khi khai báo biến và khởi tạo giá trị cho biến với kiểu dữ liệu, sau đó bạn muốn sử dụng giá trị của biến đó thành tên biến và có giá trị chính là giá trị của biến trước đó thì sử dụng cặp dấu $$. Ví dụ, biến $var có giá trị là "total", sau đó muốn sử dụng biến là total thì khai báo như ví dụ 2-5 (change.php).

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Change DataType of Variable</h4>
<?php
     $var="total";
     echo $var;
     echo "<br>";
     $$var=10;
     echo $total;

?>
</BODY>
</HTML>
```

3.4. Kiểu Array

Kiễu mảng là một mảng số liệu do người dùng định nghĩa, chúng có cú pháp như sau:

```
$myarrs=array("first", "last", "company");

// mảng bao gồm các kiểu chuỗi
```

hay có thể khai báo như sau

```
$myarr[]=array(3);

$myarr[0]="Number 0";

$myarr[1]="Number 1";

$myarr[2]="Number 2";
```

Thứ tự index trong mảng bắt đầu từ vị trí 0. Chẳng hạn, bạn khai báo mảng một chiều theo hai cách trên như ví dụ 2-6 (array.php).

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Array on demenssion</h4>
<?php
     $myarr[]=array(3);
     $myarr[0]="Number 0";
     $myarr[1]="Number 1";
     $myarr[2]="Number 2";
     echo $myarr[0];
     echo $myarr[1];
```

```
    echo $myarr[2];
    echo "<br>";
    $myarrs=array("first", "last", "company");
    echo $myarrs[2];
?>
</BODY>
</HTML>
```

Nếu như bạn khai báo mảng hai chiều, thì cú pháp khai báo như sau:

```
$myarrs[][]=array(2,3);
```

Chẳng hạn khai báo như ví dụ 2-7 (arrays.php):

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Array two demenssions</h4>
<?php
   $myarrs[][]=array(2,3);
   $myarrs[0][0]="Number 00";
   $myarrs[1][0]="Number 10";
   $myarrs[0][1]="Number 01";
   $myarrs[1][1]="Number 11";
   $myarrs[0][2]="Number 02";
   $myarrs[1][2]="Number 13";
   echo $myarrs[0][2];
   echo "<br>";
?>
</BODY>
</HTML>
```

3.5. Kiểu đối tượng

Để khai báo đối tượng, bạn sử dụng khái niệm class như trong ngôn ngữ lập trình C hay java, ngoài ra phương thức trong PHP được biết đến như một hàm. Điều này có nghĩa là từ khoá là function.

Nếu hàm có tên trùng với tên của class thì hàm đó được gọi là constructor. Chẳng hạn, chúng ta khai báo class và khởi tạo chúng thì tự động constructor được gọi mỗi khi đối tượng khởi tạo, sau đó gọi hàm trong class đó như ví dụ 2-8 (object.php).

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Object</h4>
<?php
class clsA
{
  function clsA()
  {
    echo "I am the constructor of A.<br />\n";
  }
  function B()
```

```
  {
    echo "I am a regular function named B in class A.<br />\n";
    echo "I am not a constructor in A.<br />\n";
  }
}
// Gọi phương thức clsA() như constructor.
$b = new clsA();
echo "<br>";
// Gọi phương thức B().
$b->B();
?>
</BODY>
</HTML>
```

3.6. Tầm vực của biến

Tầm vực của biến phụ thuộc vào nơi khai báo biến, nếu biến khai báo bên ngoài hàm thì sẽ có tầm vực trong trang PHP, trong trường hợp biến khai báo trong hàm thì chỉ có hiệu lực trong hàm đó.

Ví dụ, chúng ta có biến $a khai báo bên ngoài hàm nhưng khi vào trong hàm thì biến $ được khai báo lại, biến này cótầm vựec bên trong hàm. Tương tự như vậy, khi biến $i khai báo trong hàm thì chỉ có tầm vực bên trong hàm cho dù chúng được khai báo lại bên ngoài như ví dụ 2-9 (scope.php).

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Scope of Variable</h4>
<?php
$a = 100;
/* global scope */
function Test()
{
     $i=10;
     $a=10;
     echo "<br>a:=$a";
     echo "<br>i:=$i";
     /* reference to local scope variable */
}
Test();
echo "<br>a:=$a";
$i=1000;
echo "<br>i:=$i";
?>
</BODY>
</HTML>
```

Ngoài ra, để sử dụng biến toàn cục trong hàm, bạn sử dụng từ khoa global, khi đó biến toàn cục sẽ có hiệu lực bên trong hàm. Ví dụ khai báo biến $a bên ngoài hàm, sau đó bên trong hàm Test bạn sử dụng từ khoá global cho biến $a, khi đó biến $a sẽ được sử dụng và giá trị đó có hiệu lực sau khi ra khỏi hàm chứ không gống như trường hợp trong ví dụ scope.php như ví dụ 2-10 (global.php).

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
```

```
</HEAD>
<BODY>
<h4>Scope of Variable</h4>
<?php
$a = 100;
/* global scope */
function Test()
{
     global $a;
     $i=10;
     $a+=10;
     echo "<br>a:=$a";
     echo "<br>i:=$i";
     /* reference to local scope variable */
}
Test();
echo "<br>a:=$a";
$i=1000;
echo "<br>i:=$i";
?>
</BODY>
</HTML>
```

4. HẰNG TRONG PHP

4.1. Khai báo và sử dụng hằng

Hằng là giá trị không thay đổi kể từ sau khi khai báo, bạn có thể sử dụng phát biểu Define để khai báo hằng như sau:

```
define("MAXSIZE", 100);
```

Để sử dụng hằng, bạn khai báo như ví dụ 2-11 (constant.php)

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Constant</h4>
<?php
   define("pi",3.14);
   function Test()
   {
            echo "<br>pi:=".pi;
            echo "<br>pi:=".constant("pi");
   }
   Test();
   echo "<br>pi:=".pi;
   echo "<br>pi:=".constant("pi");
?>
</BODY>
</HTML>
```

4.2. Kiểm tra hằng

Khi sử dụng hằng, mà hằng chưa tồn tại thì bạn sử dụng hàm defined như ví dụ 2-12 sau (defained.php):

```
<HTML>
```

```
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Constant</h4>
<?php
define("pi",3.14);
//define("hrs",8);
function Test()
{
     if(defined("pi"))
       echo "<br>pi:=".pi;
     else
       echo "<br>pi not defined";
     if(defined("hrs"))
       echo "<br>hrs:=".hrs;
     else
       echo "<br>hrs not defined";
}
Test();
?>
</BODY>
</HTML>
```

5. KẾT LUẬN

Trong bài này, chúng ta tìm hiểu cách khai báo hằng, biến và sử dụng hằng biến. Ngoài ra, bạn cũng tìm hiểu cách chuyển đổi kiểu dữ liệu, kiểm tra kiểu dữ liệu, tầm vựec của biến.

Bài 3

PHÉP TOÁN VÀ PHÁT BIỂU CÓ ĐIỀU KIỆN TRONG PHP

Chương này chúng ta sẽ làm quen và tìm hiểu toán tử, phát biểu có điều kiện và vòng lặp của PHP.

Những vấn đề chính sẽ được đề cập trong bài học này

- ✓ *Toán tử.*
- ✓ *Phép gán trong PHP*
- ✓ *Phát biểu có điều khiển.*
- ✓ *Vòng lặp.*

1. KHÁI NIỆM VỀ CÁC TOÁN TỬ TRONG PHP

Khi bạn lập trình trên PHP là sử dụng cú pháp của ngôn ngữ C, C++. Tương tự như những ngôn ngữ lập trình khác, toán tử giúp cho bạn thực hiện những phép toán như số học hay trên chuỗi.

Bảng sau đây giúp cho bạn hình dung được những toán tử sử dụng trong PHP, PHP định nghĩa toá tử toán học, quan hệ, số học, bit và nột số phép toán gán.

Loại toán tử	Toán tử	Diễn giải	Ví dụ
Arithmetic	+	Addition	a + b
	-	Subtraction	a - b
	*	Multiplication	a * b
	/	Division	a / b
	%	Modulus	a % b
Relational	>	Greater than	a > b
	<	Less than	a < b
	>=	Greater than or equal	a >= b
	<=	Less than or equal	a <= b
	!=	Not equal	a != b
	==	Equal	a == b
Logical	!	Not	!a
	&&	AND	a && b
	||	OR	a || b

Assignment	=	Increment and assign	a = b a++ a-- a += b a -= b a *= b a /= b a %= b a |= b a &= b a ^= b a .= b
	++	Decrement and assign	
	--	Add and assign	
	+=	Subtract and assign	
	-=	Multiply and assign	
	*=	Divide and assign	
	/=	Take modulus and assign	
	%=	OR and assign	
	|=	AND and assign	
	&=	XOR and assign	
	^=	Concat and assign	
	.=		
Allocation	new	Create a new object of a class	new A()
Selection	? :	If...Then selection	a ? b : c

2. GIỚI THIỆU TOÁN TỬ

Khi nói đến toán tử, chúng ta luôn liên tưởng đến thứ tự xử lý, cũng như trong toán học, toán tử trong PHP cũng co độ ưu tiên add-subtract-multi-divide.

2.1. Toán tử AND

Khi thực hiện một việc tăng lên giá trị thì bạn sử dụng cú pháp như sau:

$ i=0;$j=0;

j=i++;// i tăng sau khi gán i vào j, chính vì vậy sau khi gán i vào j, j vẫn không thay đổi

j=++i;// i tăng trước khi gán i vào j, chính vì vậy sau khi gán i vào j, j thay đổi.

Ví dụ 3.1: Phép toán AND.

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>AND Operator</h4>
<?php
```

```
  $i=10;
  $j=5;
  $j+=$i++;
  echo "j=$j";
  echo "<br>";
  echo "i=$i";
  echo "<br>";
  $j+=++$i;
  echo "j=$j";echo "<br>";
?>
</BODY>
</HTML>
```

2.2. Toán tử Not: ~ And !

Toán tử ~ đảo nghịch tất cả các bit của tham số, còn toán tử ! đảo nghịch giá trị của giá trị trước đó. Chẳng hạn trong trường hợp này chúng ta sử dụng cho biểu thức hay biến có giá trị boolean.

Ví dụ 3.2: Phép toán ~ and !

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>~, ! Operator</h4>
<?php
  $i=10;
  $j=5;
  $j+=~$i;
  echo "j=$j";
  echo "<br>";
  $j+=~$i++;
  echo "i=$i";
  echo "<br>";
  $j+=++$i;
  echo "j=$j";
  echo "<br>";
?>
</BODY>
</HTML>
```

2.3. Toán tử nhân và chia: * and /

Bạn có thể tham khảo ví dụ sau

Ví dụ 3.3: Phép toán * và /, + và -

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
```

```
</HEAD>
<BODY>
<h4>Multi And Divide Operator</h4>
<?php
    $i=10;
    $j=5;
    echo $i/$j;
    echo "<br>";
    echo $i*$j;
?>
</BODY>
</HTML>
```

2.4. Toán tử modulus: %

Khi chia một số cho một số, bạn cần kết quả là số dư của phép chia đó thì dùng toán tử modulus

Ví dụ 3.4: Phép toán %

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Mod Operator</h4>
<?php
    $i=10;
    $j=7;
    echo $i%$j;
    echo "<br>";
?>
</BODY>
</HTML>
```

2.5. Toán tử quan hệ: >=,>,<,<=,==,!=

Khi cần so sánh kết quả giữa hai toán hạn với nhau, thông thường bạn nghĩ đến phép toán so sánh như là bằng, lớn hơn, nhỏ hơn, ví dụ sau diễn giải cho bạn các toán tử trên:

Ví dụ 3.5: Phép toán >,>=,<,<=,==,!=

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Comparation Operators</h4>
<?php
```

```
    $i=10;
    $j=9;
    echo $i<$j;
    echo "<br>";
    echo $i!=$j;
?>
</BODY>
</HTML>
```

2.6. Toán tử && và ||

&& là toá tử and trong só học, || là toán tử or trong số học. Hai toán tử này rất thường dùng trong khi lập trình trên PHP, ví dụ dưới đây diễn giải cho bạn đầy đủ hai toán tử này. Chú ý rằng khi sử dụng toán tử đều có kèm phát biểu có điều kiện.

Ví dụ 3.6: Phép toán && và ||

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Logic Operators</h4>
<?php
    $b=true;
    $j=3;
    if (($j>=3) && ($b!=true))
    {
      echo "result is true";
    }
    if(($j<3) || ($b==true))
      echo "result is false";
?>
</BODY>
</HTML>
```

2.7. Toán tử ?:

Toán tử này thay thế cho phát biểu có điều kiện if...else, khi bạn cần lấy kết quả theo điều kiện nào đó, nếu có thể không cần phát biểu if-else, thì hãy thay thế bằng toán tử ?:, cú pháp của chúng như sau:

str1=str2.equals(”khang”)?”Welcome to PHP”:”Good bye PHP”;

Ví dụ 3.7: Phép toán ?:

```
<HTML>
```

```
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Selection Operators</h4>
<?php
    $str1="Pham Huu Khang";
    $str2 ="Khang";
    $str1=(str1==str2)?"Welcome to PHP":"Good bye PHP";
    echo "result is ".$str1;
?>
</BODY>
</HTML>
```

3. PHÉP GÁN

Khi gán một giá trị hay biến vào một biến trong PHP, bạn phải dùng đến phép gán, nhưng trong PHP cũng giống như trong C thì có những phép gán được đơn giản hoá hay nói đúng hơn là chuẩn hoá để rút gọn lại trong khi viết.

3.1. Phép gán thông thường nhất như sau:

$j=i;

$str1 =” Hello!”;

$b=true;

3.2. Phép gán thêm một giá trị là 1

$k=0;

$k++;

3.3. Phép gán chuỗi

$strX="Hello";

$strX.=” world”;

$strX.=”ABCc”.$x;

3.4. Phép gán thêm một với chính nó giá trị

$k=0;$j=1;

$k+=$j;

tương tự như vậy chúng ta có $k*=2, nghĩa là $k=$k*2

4. PHÁT BIỂU CÓ ĐIỀU KIÊN

Các phat biểu có điều kiện như :

- IF (điều kiện) { câu lệnh; }
- IF (điều kiện) { câu lệnh; }ELSE { câu lệnh; }

- IF (điều kiện) { câu lệnh; }ELSEIF { câu lệnh; }
- switch (điều kiện)

```
{
   case Value1
        câu lệnh1;
        break;
   }
```

- While (điều kiện)
- Do - While (điều kiện)
- Break
- Continue

4.1. Phát biểu IF (điều kiện) { câu lệnh; }

Sử dụng phát biểu if để chọn lọc kết quả khi điều kiện đúng, ví dụ như sau:

Ví dụ 3.8: Phát biểu IF

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>IF Statement</h4>
<?php
    $b=true;
    $j=3;
    if(($j>=3) &&($b!=true))
      echo "result is true";
    if(($j<3) ||($b==true))
      echo "result is false";

?>
</BODY>
</HTML>
```

4.2. Phát biểu IF (điều kiện) { câu lệnh; }ELSE { câu lệnh; }

Sử dụng phát biểu if để chọn lọc kết quả khi điều kiện đúng, và xuất ra kết quả khi điều kiện sai, ví dụ như sau:

Ví dụ 3.9: Phát biểu IF - ELSE

```
<HTML>
<HEAD>
```

```
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>IF ELSE Statement</h4>
<?php
    $b=true;
    $j=3;
    if ($j>3)
      echo "result is true";
    else
    {
      $j++;
      echo "result is $j";
    }
?>
</BODY>
</HTML>
```

4.3. Phát biểu ELSEIF

Phát biểu elseif là phần của phát biểu if else nhiều nhánh, khi có nhiều điều kiện chọn lựa thì bạn sử dụng elseif, cú pháp của chúng như sau:

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>ELSEIF Statement</h4>
<?php
    $b=true;
    $j=3;
    if ($j>3)
      echo "result is true";
    elseif ($j=0)
    {
      $j++;
      echo "result is $j";
    }
    else
    {
      $j--;
      echo "result is ". $j--;
    }
?>
</BODY>
</HTML>
```

4.4. Phát biểu Switch (điều kiện)

Phát biểu switch là phần của phát biểu elseif nhiều nhánh, khi có nhiều điều kiện chọn lựa thì bạn sử dụng switch, cú pháp của chúng như sau:

```
Switch(điều kiện)
  {
    case Value1
        câu lệnh1;
        break;
    case Value2
        câu lệnh2;
        break;
    …
      default:
        câu lệnh default;
  }
```

Break: dùng để thoát ra khỏi switch khi thoả một case nào đó trong switch, default: khi không có bất kỳ giá trị nào thoản trong các case thì giá trị cuối cùng là defaule statement

Ví dụ 3.10: Phát biểu Switch

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>SWITCH Statement</h4>
<?php
    $j=3;
    $j=date("w");
    $str="";
    switch($j)
       {
     case 0:
      $str="Today is  Sunday";
      break;
     case 1:
      $str="Today is Monday";
      break;
     case 2:
      $str="Today is Tueday";
      break;
     case 3:
      $str="Today is Wednesday";
      break;
     case 4:
      $str="Today is Thursday";
```

```
        break;
       case 5:
        $str="Today is Friday";
        break;
       case 6:
        $str="Today is Saturday";
        break;
       default:
        $str="Today is Sunday";
        break;
      }
      echo $str;
?>
</BODY>
</HTML>
```

4.5. Phát biểu While(điều kiện)

Phát biểu while thực thi những câu lệnh trong while khi điều kiện có giá trị true.

Ví dụ 3.11: Phát biểu While

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>While Statement</h4>
<?php
    $j=10;
    while($j>0)
    {
      echo $j."<br>";
      $j--;
    }
?>
</BODY>
</HTML>
```

4.6. Phát biểu For

Phát biểu for dùng cho vòng lặp có giới hạn cho trước, cú pháp có dạng như sau:

Ví dụ 3.12: Phát biểu For

```
<HTML>
<HEAD>
```

```
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>FOR Statement</h4>
<?php
    for($j=1;$j<=10;$j++)
    {
      echo $j."<br>";
    }
?>
</BODY>
</HTML>
```

4.7. Phát biểu do while

Phát biểu do while cho phép duyệt và kiểm tra điều kiện sau phát biểu thứ nhất, điều này có nghĩa là ít nhất một phát biểu được thực hiện.

Ví dụ 3.13: Phát biểu Do While

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Do While Statement</h4>
<?php
    $j=10;
    do
    {
      echo $j."<br>";
      $j--;
    }while($j>0)
?>
</BODY>
</HTML>
```

Phát biểu exit cho phép thóat ra khỏi phát biểu điều kiện khi thoả điều kiện nào đó.

Ví dụ 3.14: Phát biểu exit

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Exit Statement</h4>
<?php
    $j=10;
    do
```

```
    {
      if($j==3) exit;
      echo $j."<br>";
      $j--;
    }while($j>0)
?>
</BODY>
</HTML>
```

5. TÓM TẮT

Trong bài học này chúng tôi giới thiệu đến cho các bạn các phép gán, các toán tử, đồng thời giúp cho các bạn hiểu thêm vào các phát biểu có điều kiện như while, for, switch, ….

Môn học: PHP

Bài 4

*Bài học này chúng ta sẽ làm quen với biến form và hai phương thức **$HTTP_POST_VARS** và **$HTTP_GET_VARS** của PHP:*

- ✓ *Biến form.*
- ✓ *Phương thức $HTTP_GET_VARS*
- ✓ *Phương thức $HTTP_POST_VARS*

1. BIẾN FORM

Biến form trong PHP được biết đến như một loại biến, thay vì khai báo thì biến đó chính là tên của thẻ nhập liệu trong trang submit hay tham số trên querystring.

1.1. Biến form từ form được submit với phương thức POST

Trong trang bạn submit đến, nếu khai báo tên của thẻ nằm trong thẻ form có tên là xyz thì biến form được định nghĩa là $xyz.

Chẳng hạn, bạn khai báo báo thẻ form trong trang submit.php như ví dụ 4-1.

Ví dụ 4-1: Khai báo thẻ form

```
...
<form action=ex1-1.php method=post>
<tr>
  <td>Name</td><td>:<input type=text name=fullname></td>
</tr>
<tr><td>Gender</td>
  <td>:<input type=radio value=M name=gender> Male
  <input type=radio value=F name=gender> Female</td>
</tr>
<tr><td> </td>
  <td><input type=submit value=Submit></td>
</tr>
</form>
...
```

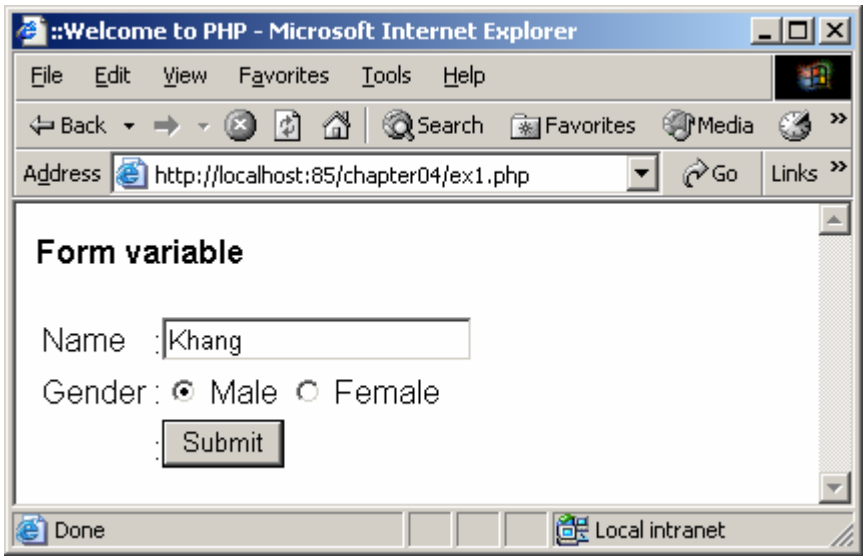
Khi người sử dụng nhập giá trị vào phần Name và chọn giới tính Male hay Female như hình 4-1, nếu nhấn nút submit thì trang ex1-1.php sẽ triệu gọi, trong trang này bạn có thể lấy giá trị nhập từ trang ex1.php bằng cách sử dụng biến form như ví dụ 4-1-1.

Ví dụ 4-2: Dùng biến form

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Form variable</h4>
<table>
<tr><td>Name</td>
<td>
  :<?=$fullname?>
```

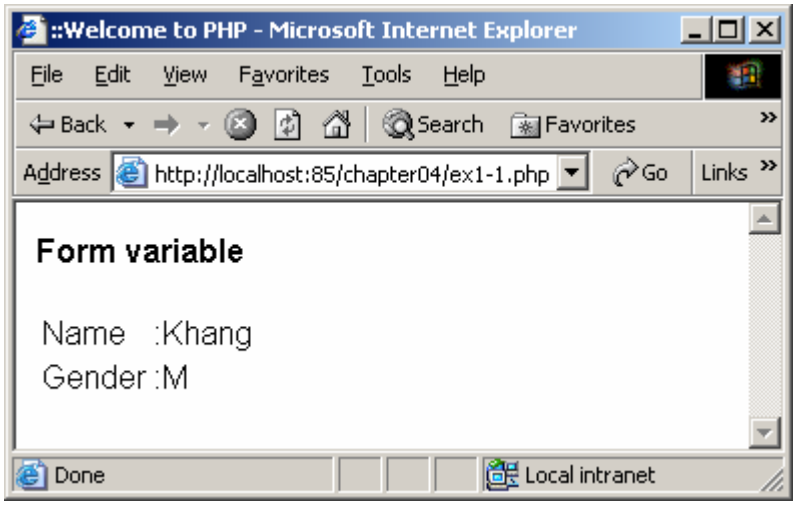
```
</td></tr>
<tr><td>Gender</td>
<td>
  :<?=$gender?>
</td></tr>
</table>
</BODY>
</HTML>
```

Trong đó, $fullname và $gender là tên của hai thẻ input trong trang ex1.php, trong trường hợp này chúng ta sử dụng phương thức POST cho form.



Hình 4-1: Nhập liệu

Kết quả trả về như hình 4-1-1.



Hình 4-1-1: Kết quả lấy từ trang submit bằng biến form

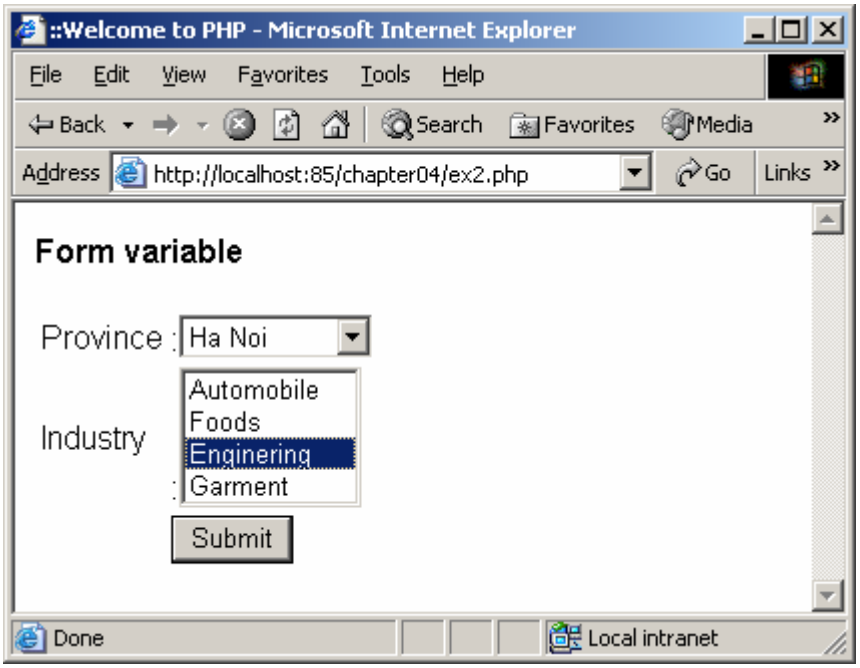
1.2. Biến form từ form được submit với phương thức GET

Nếu bạn sử dụng phương thức GET trong thẻ form, bạn có thể lấy giá trị của các tham số trên chuỗi QueryString bằng biến form. Ví dụ khai báo thẻ form có hai tuỳ chọn như ví dụ 4-2 với phương thức GET trong thẻ form.

Ví dụ 4-2: Khai báo thẻ form

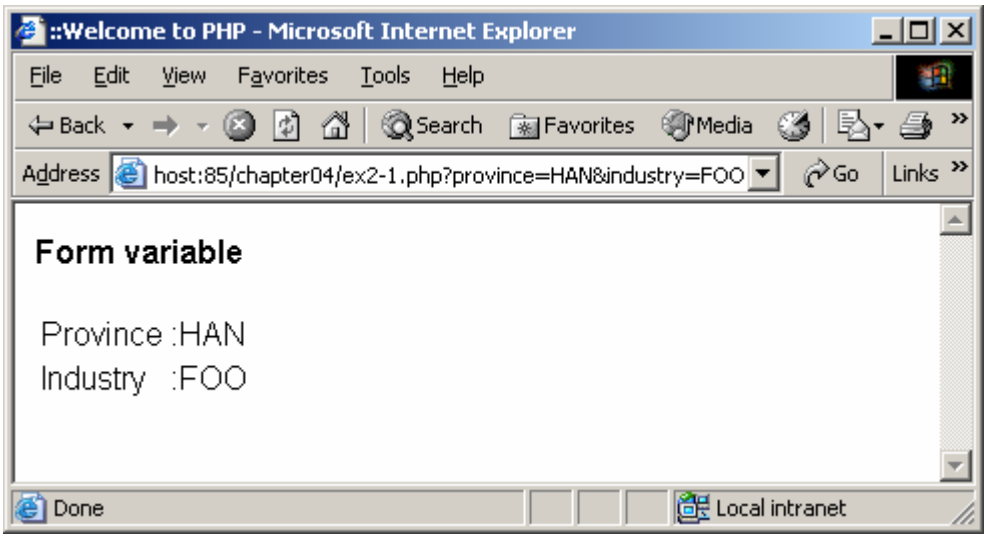
```
...
<form action=ex2-1.php method=get>
<tr><td>Province</td>
<td>
   :<select name=province>
   <option value=HAN>Ha Noi</option>
   <option value=HCM>Ho Chi Minh</option>
   <option value=HUE>Hue</option>
   </select>
</td></tr>
<tr><td>Industry</td>
<td>
   :<select name=industry multiple>
   <option value=AUT>Automobile</option>
   <option value=FOO>Foods</option>
   <option value=ENG>Enginering</option>
   <option value=GAR>Garment</option>
   </select>
</td></tr>
<tr><td> </td>
<td><input type=submit value=Submit></td></tr>
</form>
...
```

Khi triệu gọi trang ex2.php trên trình duyệt, người sử dụng chọn giá trị trong hai tuỳ chọn Province và Industry như hình 4-2.



Hình 4-2: Phương thức GET

Nếu nhấn Submit thì hai giá trị chọn sẽ được truyền lên trên QueryString với hai tham số là tên của thẻ select. Ví dụ trong trường hợp này kết quả trả về như hình 4-2-1.



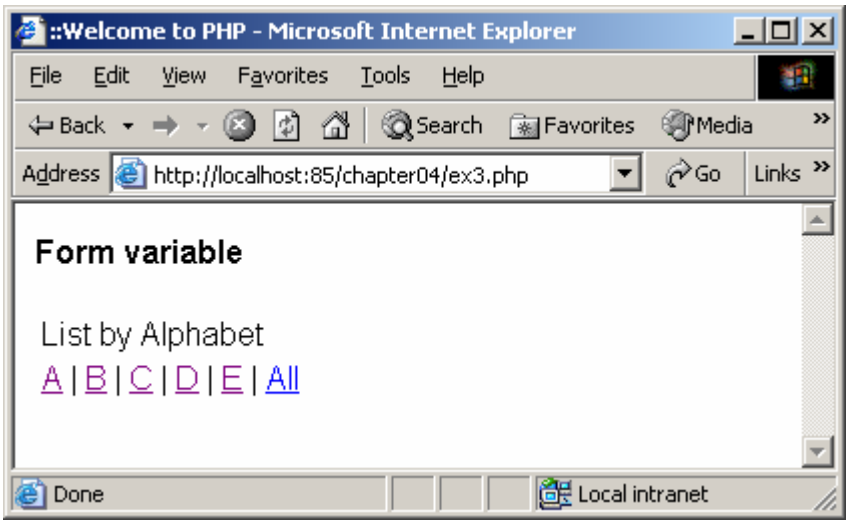
Hình 4-2-1: Biến form với phương thức GET

Trong đó, hai tham số và giá trị tương ứng là ex2-1.php?province=HAN&industry=FOO, bằng cách sử dụng biến form bạn có thể lấy được giá trị này như ví dụ 4-2-1.

Ví dụ 4-2-1: Khai báo thẻ form

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Form variable</h4>
<table>
<tr><td>Province</td>
<td>
  :<?=$province?>
</td></tr>
<tr><td>Industry</td>
<td>
  :<?=$industry?>
</td></tr>
</table>
</BODY>
</HTML>
```

Đối với trường hợp bạn không sử dụng thẻ form như hai trường hợp trên, chúng ta cũng có thể lấy giá trị từ chuỗi QueryString bằng biến form. Chẳng hạn, bạn khai báo trang chop phép người sử dụng chọn ký tự để liệt kê danh sách khách hàng theo ký tự đó như hình 4-3.



Hình 4-3: Chọn ký tự

Bằng cách khai báo các thẻ <a> bạn định nghĩa 24 ký tự như hình trên với tham số al có giá trị tương ứng:

```
<tr><td>
<a href="ex3.php?al=A">A</a> |
<a href="ex3.php?al=B">B</a> |
<a href="ex3.php?al=C">C</a> |
<a href="ex3.php?al=D">D</a> |
<a href="ex3.php?al=E">E</a> |
<a href="ex3.php?al=">All</a>
</td></tr>
```

Khi người sử dụng chọn một ký tự thì sử dụng biến form là tên của tham số (al), bạn có thể lấy được giá trị của ký tự đang chọn:

```
<tr><td>Select:<?=$al?></td></tr>
```

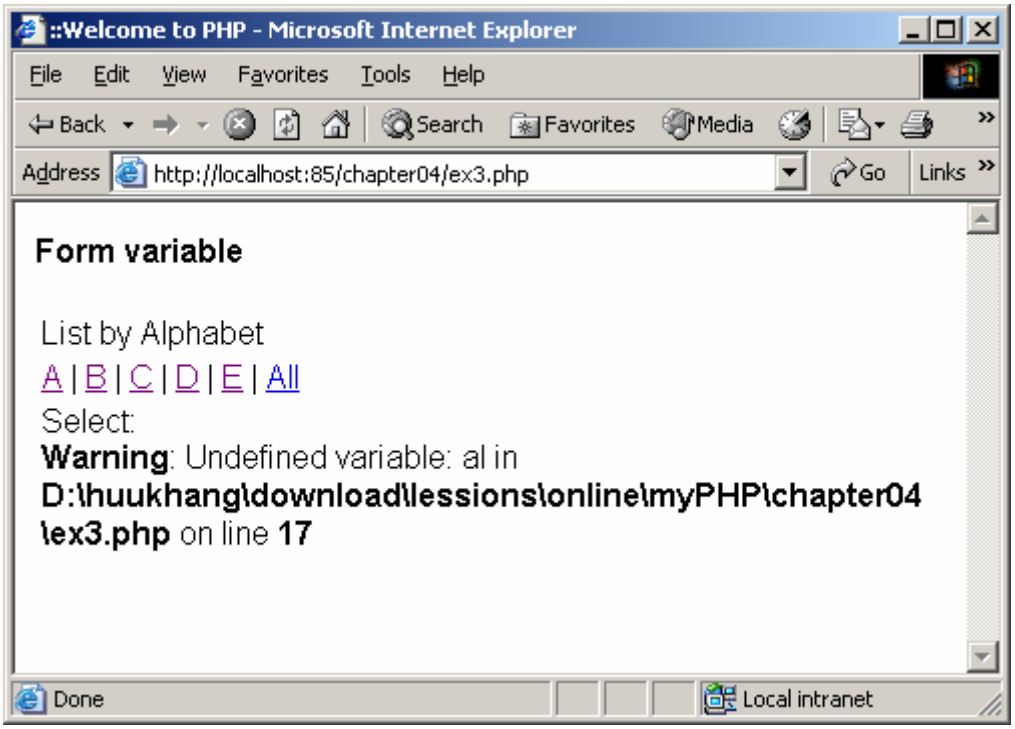
Tuy nhiên, lần đầu tiên triệu gọi trang này mà không có tham số trên QueryString, khai báo biến form sẽ phun ra lỗi như hình 4-3-1.

Để tránh trường hợp này, bạn sử dụng hàm isset để kiểm tra biến tồn tại hay không, nếu tồn tại thì bạn sử dụng biến form này. Ví dụ đối với trường hợp này chúng ta khai báo như ví dụ 4-3.

Ví dụ 4-3: Sử dụng biến form

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Form variable</h4>
<table>
<tr><td>List by Alphabet</td></tr>
<tr><td>
```

```
<a href="ex3.php?al=A">A</a> |
<a href="ex3.php?al=B">B</a> |
<a href="ex3.php?al=C">C</a> |
<a href="ex3.php?al=D">D</a> |
<a href="ex3.php?al=E">E</a> |
<a href="ex3.php?al=">All</a>
</td></tr>
<?php
if(isset($al))
  {
  ?>
     <tr><td>Select:<?=$al?></td></tr>
  <?php
  }
  ?>
</table>
</BODY>
</HTML>
```



Hình 4-3-1: Lỗi phát sinh

Chú ý rằng, khi sử dụng biến form bạn không nên khai báo biến cùng tên với các tham số hay tên của thẻ nhập liệu trong trang triệu gọi trước đó. Nếu không thì giá trị trả về là giá trị của biến thường thay vì biến form.

2. PHƯƠNG THỨC $HTTP_GET_VARS

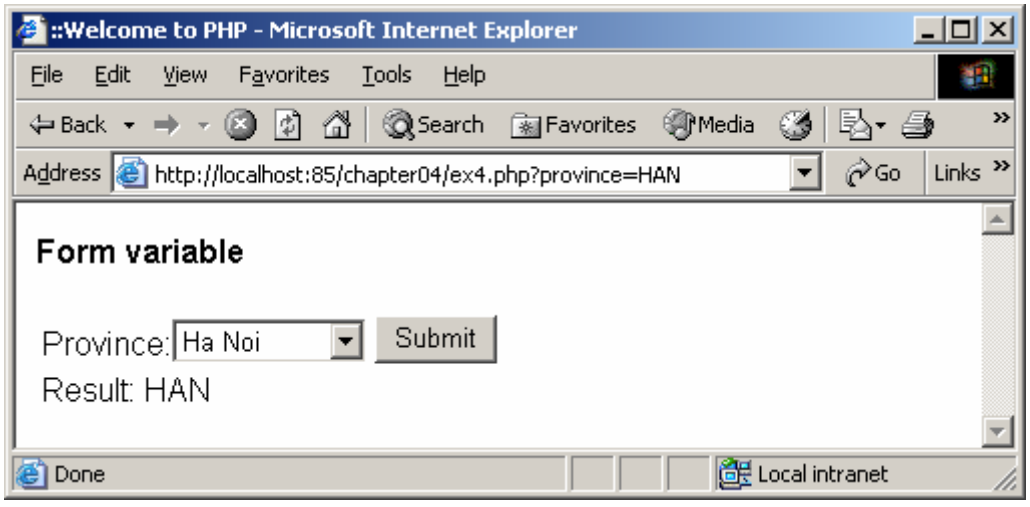
Ngoài cách sử dụng biến form trong trường hợp lấy giá trị từ tham số của QueryString, bạn có thể sử dụng hàm $HTTP_GET_VARS. Ví dụ, chúng ta khai báo trang PHP như ví dụ 4-4.

Ví dụ 4-4: Sử dụng $HTTP_GET_VARS

```
<HTML>
```

```
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Form variable</h4>
<table>
<form action=ex4.php method=get>
<tr><td>Province:<select name=province>
     <option value=HAN>Ha Noi</option>
     <option value=HCM>Ho Chi Minh</option>
     <option value=HUE>Hue</option>
     </select>
<input type=submit value=Submit></td></tr>
</form>
<tr><td>
<?php
     if(isset($HTTP_GET_VARS["province"]))
     {
      $result=$HTTP_GET_VARS["province"];
      echo "Result: ".$result;
     }
?>
</td></tr>
</table>
</BODY>
</HTML>
```

Lưu ý rằng, nếu bạn không sử dụng hàm isset để kiểm tra province tồn tại hay không thì trang php sẽ phun lỗi trong trường hợp lần đầu tiên gọi đến trang ex4.php mà không submit. Tuy nhiên, nếu bạn submit trang này thì kết quả trả về như hình 4-4.



Hình 4-4: Dùng $HTTP_GET_VARS

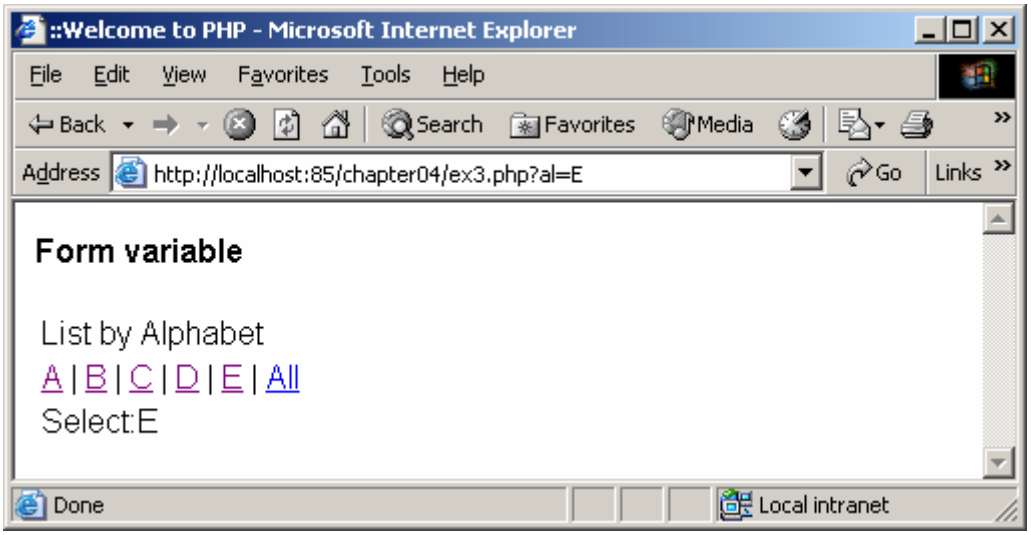
Tương tự như vậy trong trường hợp bạn không sử dụng thẻ form mà giá trị lấy từ chuỗi QueryString bằng cách sử dụng $HTTP_GET_VARS như ví dụ 4-5.

Ví dụ 4-5: Sử dụng $HTTP_GET_VARS

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
```

```
<BODY>
<h4>Form variable</h4>
<table>
<tr><td>List by Alphabet</td></tr>
<tr><td>
<a href="ex3.php?al=A">A</a> |
<a href="ex3.php?al=B">B</a> |
<a href="ex3.php?al=C">C</a> |
<a href="ex3.php?al=D">D</a> |
<a href="ex3.php?al=E">E</a> |
<a href="ex3.php?al=">All</a>
</td></tr>
<?php
if(isset($HTTP_GET_VARS["al"]))
     {
     ?>
     <tr><td>Select:<?=$HTTP_GET_VARS["al"]?></td></tr>
     <?php
     }
?>
</table>
</BODY>
</HTML>
```

Kết quả trả về như hình 4-5.



Hình 4-5: Sử dụng $HTTP_GET_VARS

3. PHƯƠNG THỨC $HTTP_POST_VARS

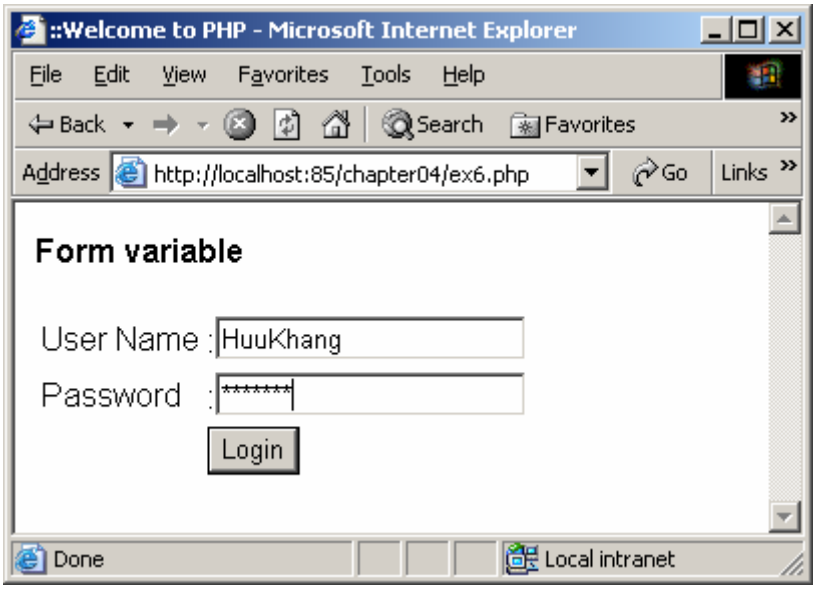
Tương tự như $HTTP_GET_VARS nhưng $HTTP_POST_VARS cho phép bạn lấy giá trị lấy từ các thẻ nhập liệu của thẻ form trong traang submit trước đó. Ví dụ, bạn khai báo trang nhập liệu như ví dụ 4-6.

Ví dụ 4-5: Khai báo form với phương thức POST

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
```

```
<BODY>
<h4>Form variable</h4>
<form action=ex7.php method=post>
<table>
<tr><td>User Name</td>
<td>
     :<input type=text name=username>
</td></tr>
<tr><td>Password</td>
<td>
     :<input type=password name=password>
</td></tr>
<tr><td> </td>
<td><input type=submit value=Login></td></tr>
</table>
</form>
</BODY>
</HTML>
```

Khi người sử dụng nhập username và password như hình 4-6 và nhấn nút Login.



Hình 4-6: Đăng nhập

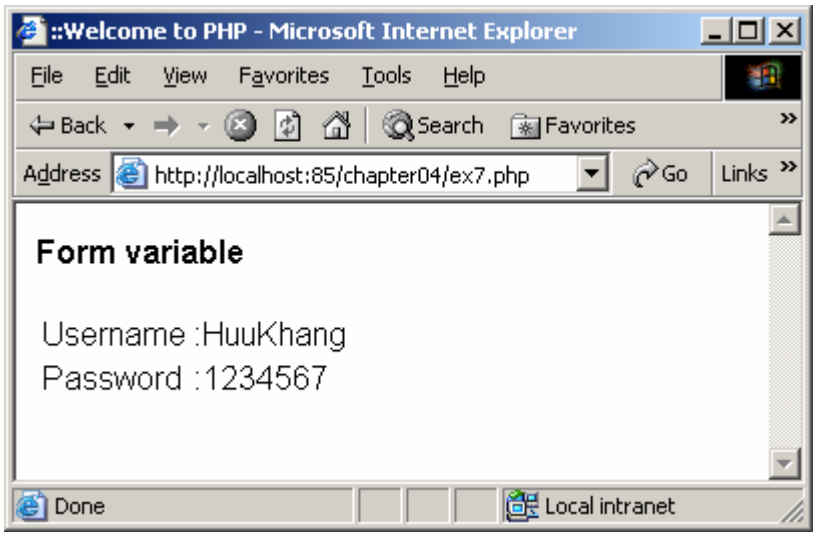
Bằng cách sử dụng $HTTP_POST_VARS để lấy giá trị username và password như ví dụ 4-7.

Ví dụ 4-5: Sử dụng $HTTP_POST_VARS

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Form variable</h4>
<?php
if(isset($HTTP_POST_VARS["username"]))
{
     ?>
```

```
        <table>
        <tr><td>Username</td>
        <td>:<?=$HTTP_POST_VARS["username"]?></td></tr>
        <tr><td>Password</td><td>
        :<?=$HTTP_POST_VARS["password"]?></td></tr>
        </table>
    <?php
    }
    ?>
    </BODY>
    </HTML>
```

Kết quả trình bày như hình 4-7.



Hình 4-7: Dùng $HTTP_POST_VARS

4. KẾT LUẬN

Trong bài này, chúng ta tìm hiểu cách sử dụng biến form và hai phương thức $HTTP_POST_VARS, $HTTP_GET_VARS. Ngoài ra, bạn cũng tìm hiểu cách kiểm tra biến tồn tại hay không bằng hàm isset().

Chú ý rằng, khi sử dụng biến form bạn tránh trường hợp khai báo biến cục bộ hay toàn cục trong tang PHP cùng tên với thẻ nhập liệu của form trước đó submit đến hay tham số trên querystring.

Môn học: PHP

Bài 5

Bài học này chúng ta sẽ làm quen với đối tượng Session và một số đối tượng khác:

- ✓ *Đối tượng Session.*
- ✓ *Đối tượng khác*

1. ĐỐI TƯỢNG SESSION

Trong PHP4.0 đối tượng Session được xem như một đối tượng cho phép bạn truyền giá trị từ trang PHP này sang PHP khác. Để sử dụng Session, bạn khai báo thư mục được lưu trữ dữ liệu do đối tượng nay ghi ra.

Session được sinh ra và được biến mất khi người sử dụng huỷ chúng, thời gian sống của chúng đã hết hoặc người sử dụng đóng trình duyệt.

Chẳng hạn, trong trường hợp này chúng ta sử dụng thư mục C:\PHP\sessiondata được khai báo trong tập tin php.ini.

```
session.save_path = C:\PHP\sessiondata
```

Ngoài ra, khi muốn sử dụng Sessoin thì bạn phải khởi tạo chúng. Để khởi tạo Session bạn có thể khởi tạo trong trang PHP mỗi khi truy cập hay gán giá trị cho Session.

```
session_start();
```

Tuy nhiên, bạn có thể cấu hình trong trang php.ini (1 là start).

```
session.auto_start = 0
```

1.1. Nhận dạng Session

Mỗi phiên làm việc được tạo ra từ Web Server thì sẽ có một nhận dạng duy nhất có giá trị là chuỗi do trình chủ Web tạo ra. Điều này có nghĩa là mỗi khi người sử dụng triệu gọi trang Web của Web Site lần đầu tiên thì phiên làm việc sẽ được tạo ra, khi đó một nhận dạng được cấp cho phiên làm việc đó.

Để lấn giá trị nhận dạng của Session do trình chủ Web cấp phát bạn sử dụng cú pháp:

```
$x= session_id();
```

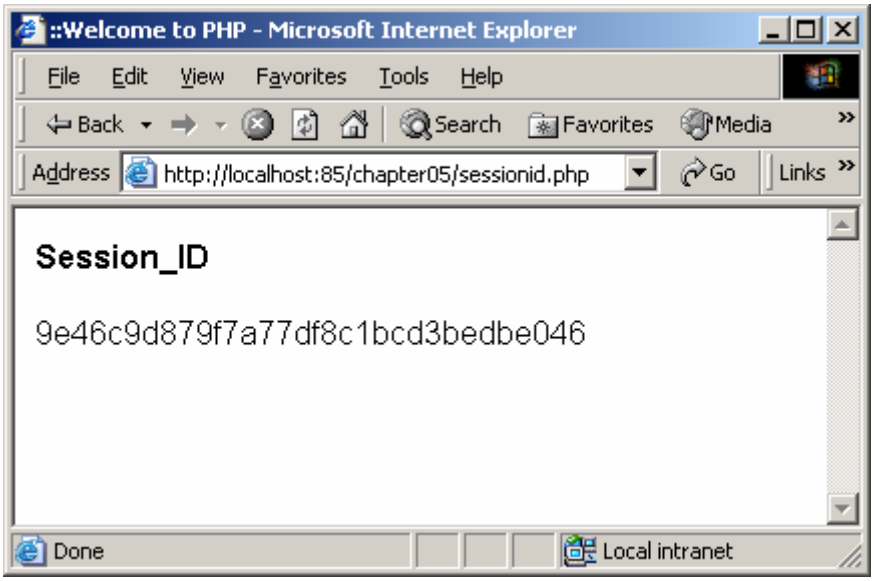
Chẳng hạn, bạn khai báo báo để lấy giá trị session_id trong trang sessionid.php như ví dụ 5-1.

Ví dụ 5-1: Nhận dạng session

```
<?php
  session_start();
?>
<HTML>
```

```
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Session_ID</h4>
<?php
  $sessionid=session_id();
  echo $sessionid;
?>
</BODY>
</HTML>
```

Mỗi người sử dụng truy cập đến Web Site sẽ có một nhận dạng khác như như hình 5-1.



Hình 5-1: Nhận dạng duy nhất

1.2. Khai báo Session

Khi muốn khai báo biến session, bạn phải sử dụng hàm session_register có cú pháp như sau:

```
session_register("sessioname");
```

Khi muốn khởi tạo session, bạn có thể gán giá trị cho session này như gán giá trị cho biến trong PHP, sau đó sử dụng hàm trên để đắng ký.

```
$sessioname=value;
session_register("sessioname");
```

Trong trường hợp có nhiều session, bạn có thể sử dụng hàm session_register để đăng ký cùng một lúc nhiều session như sau:

```
$sessioname1=value1;
$sessioname2=value2;
```

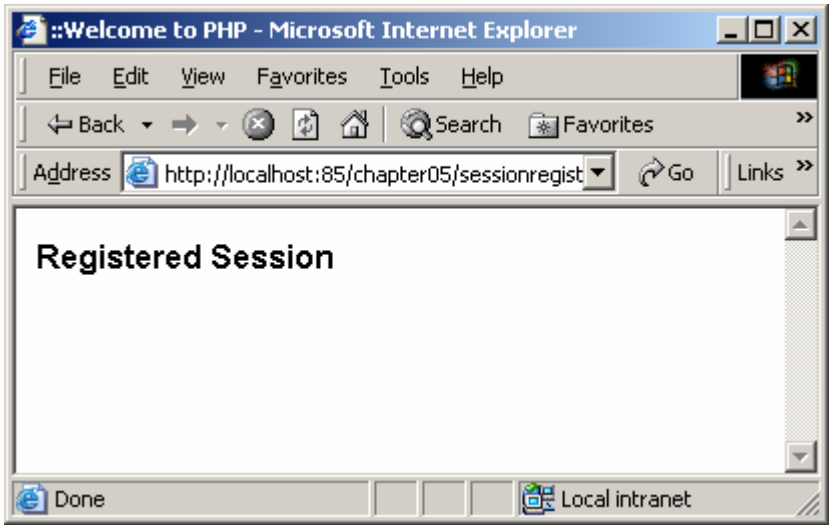
```
$sessioname3=value3;
session_register("sessioname1","sessioname2","sessioname3");
```

Chẳng hạn, trong trường hợp này chúng ta khai báo trang sessionregister.php và đăng ký 3 sessoin có tên userid, email và fullname như ví dụ 5-2 sau:

Ví dụ 5-2: Đăng ký session

```
<?php
  session_start();
?>
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Registered Session</h4>
<?php
  $userid="123";
  $email="test@yahoo.com";
  $fullname="Nguyen Van Ba";
  session_register("userid");
  session_register("email","fullname");
?>
</BODY>
</HTML>
```

Kết quả trả về như hình 5-2.



Hình 5-2: Đăng ký Session

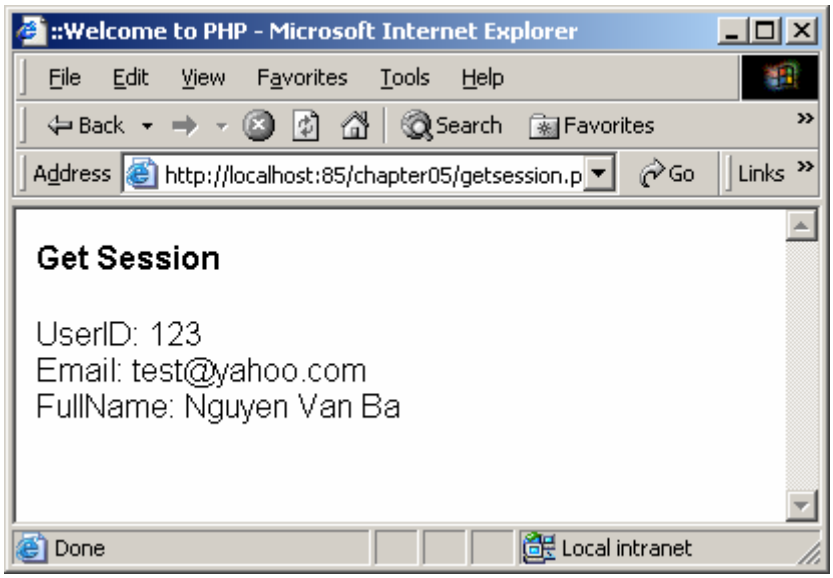
1.3. Lấy giá trị từ session

Sau khi khai báo khởi ạo một số session với giá trị tương ứng của session đó, bạn có thể truy cập các biến session này để lấy giá trị torng trang PHP khác. Chẳng hạn, chúng ta khai báo trang getsession.php để lấy các session của PHP vừa khai báo trong ví dụ trên như ví dụ 5-3.

Ví dụ 5-3: Lấy giá trị từ session

```
<?php
  session_start();
?>
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Get Session</h4>
<?php
  echo "UserID: ". $userid."<br>";
  echo "Email: ".$email."<br>";
  echo "FullName: ".$fullname;
?>
</BODY>
</HTML>
```

Khi triệu gọi trang getsession.php trên trình duyệt bạn trình bày giá trị của session userid, email và fullname như hình 5-3.



Hình 5-3: Lấy giá trị của session

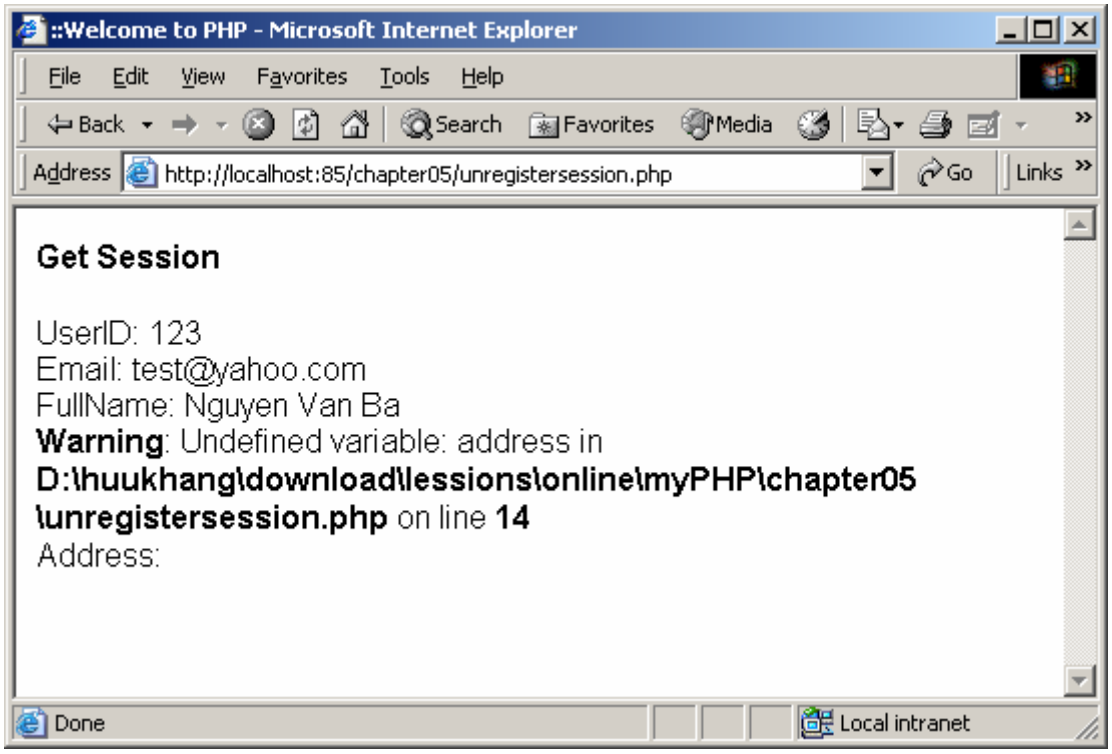
Tuy nhiên, trong trường hợp bạn truy cập một biến session chưa khởi tạo trước đó thì l64i sẽ phát sinh. Ví dụ trong trường hợp này chúng ta truy cập biến session có tên $address như ví dụ 5-4.

Ví dụ 5-4: Truy cập session chưa tồn tại

```
<?php
  session_start();
?>
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
```

```
<h4>Get Session</h4>
<?php
   echo "UserID: ". $userid."<br>";
   echo "Email: ".$email."<br>";
   echo "FullName: ".$fullname;
   echo "Address: ".$address;
?>
</BODY>
</HTML>
```

Khi triệu gọi trang unregistersession.php trên trình duyệt thì lỗi phát sinh như hình 5-4.



Hình 5-4: Lỗi phát sinh

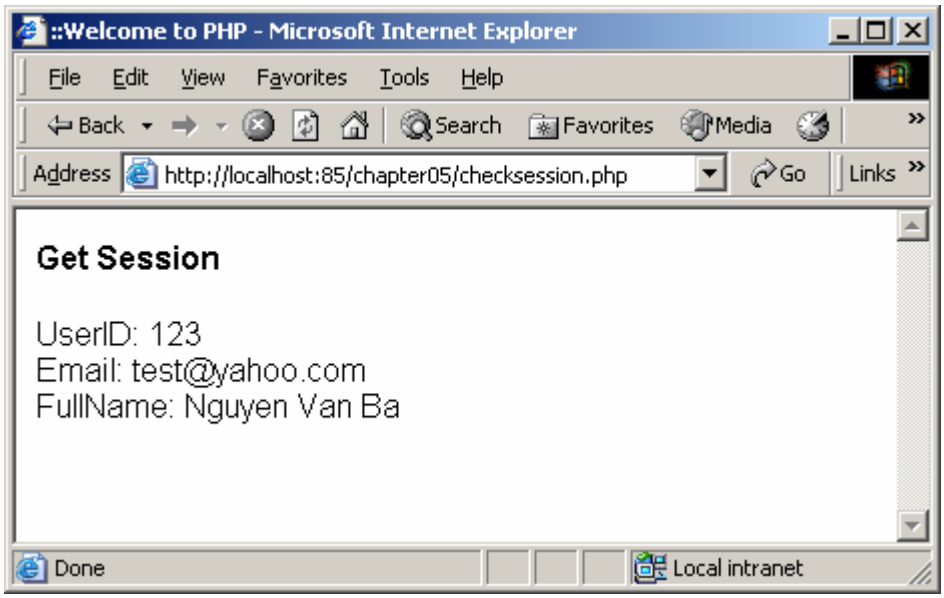
Để kiểm tra session đó có tồn tại hay chưa bạn sử dụng hàm session_is_register trong trang checksession.php. Đối với trường hợp này chúng ta cần kiểm tra 4 session trước khi truy cập đến chúng như ví dụ 5-5.

Ví dụ 5-5: Kiểm tra session

```
<?php
   session_start();
?>
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Get Session</h4>
```

```
<?php
  if(session_is_registered("userid"))
   echo "UserID: ". $userid."<br>";
  if(session_is_registered("email"))
   echo "Email: ".$email."<br>";
  if(session_is_registered("fullname"))
   echo "FullName: ".$fullname;
  if(session_is_registered("address"))
   echo "Address: ".$address;
?>
</BODY>
</HTML>
```

Khi triệu gọi trang checksession.php thì kết quả sẽ trình bày như hình 5-5.



Hình 5-5: Không có lỗi phát sinh

Chú ý rằng, khi sử dụng đến session, bạn phai khởi động chúng bằng session_start() nếu không thì phải khai báo trong php.ini.

1.4. Huỷ session

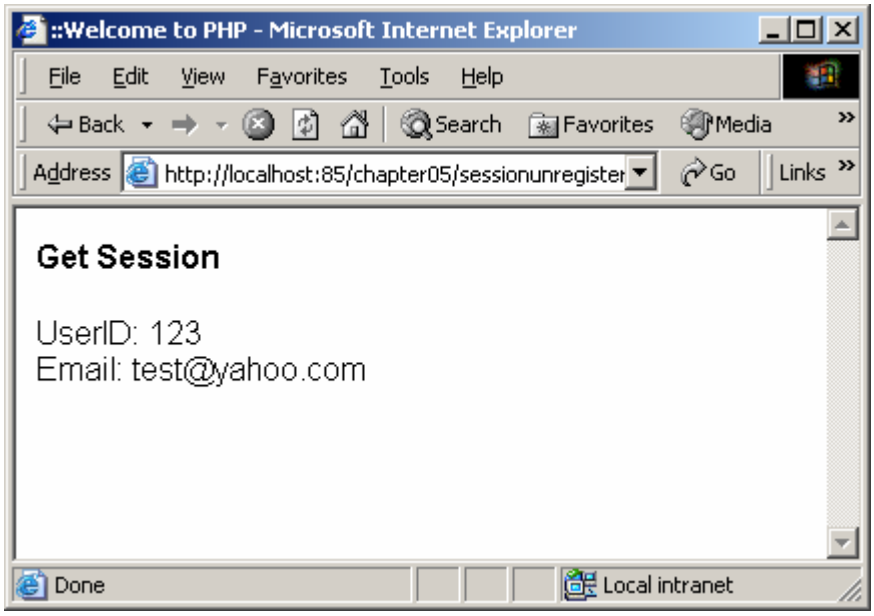
Khi không có nhu cầu sử dụng session nữa thì bạn sử dụng hàm session_unregister để loại session đó. Chẳng hạn, trong trường hợp này chúng ta muốn loại bỏ session có tên là fullname bạn khai báo trong trang sessionunregister.php như ví dụ 5-6.

Ví dụ 5-6: Loại bỏ một Session

```
<?php
  session_start();
?>
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
```

```
<h4>UnRegister Session</h4>
<?php
   session_unregister("fullname");
   if(session_is_registered("userid"))
    echo "UserID: ". $userid."<br>";
   if(session_is_registered("email"))
    echo "Email: ".$email."<br>";
   if(session_is_registered("fullname"))
    echo "FullName: ".$fullname;
   if(session_is_registered("address"))
    echo "Address: ".$address;
?>
</BODY>
</HTML>
```

Khi triệu gọi trang sessionunregister.php trên trình duyệt thìkết quả trả về như hình 5-6.



Hình 5-6: Loại bỏ session

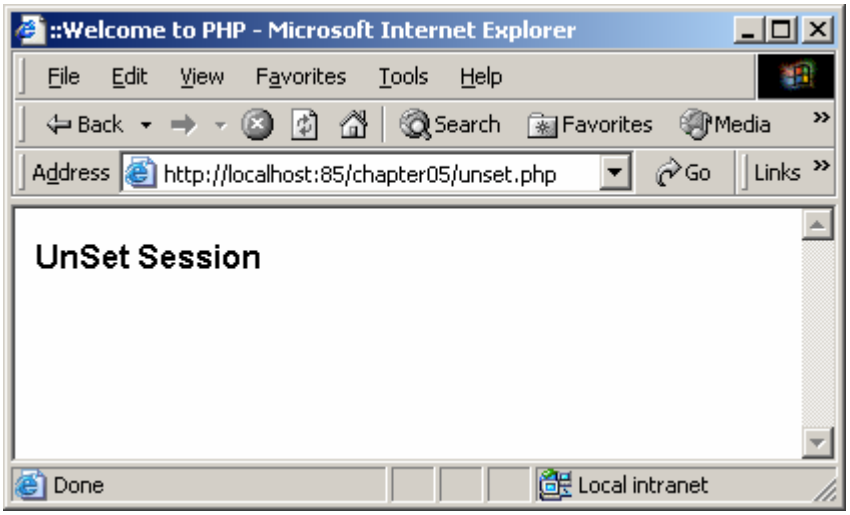
Trong trường hợp loại bỏ tất các session đang tồn tại thì sử dụng hàm session_unset(). Ví dụ dùng hàm này để lạoi bỏ session và dùng hàm sessin_destroy để huỷ tất cả session đó khai báo trong trang unset.php như ví dụ 5-7.

Ví dụ 5-7: Xoá tất cả session

```
<?php
    session_start();
?>
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>UnSet Session</h4>
<?php
    session_unset();
    session_destroy();
    if(session_is_registered("userid"))
```

```
        echo "UserID: ". $userid."<br>";
       if(session_is_registered("email"))
        echo "Email: ".$email."<br>";
       if(session_is_registered("fullname"))
        echo "FullName: ".$fullname;
       if(session_is_registered("address"))
        echo "Address: ".$address;
?>
</BODY>
</HTML>
```

Kết quả trả về như hình 5-7.



Hình 5-7: Huỷ session

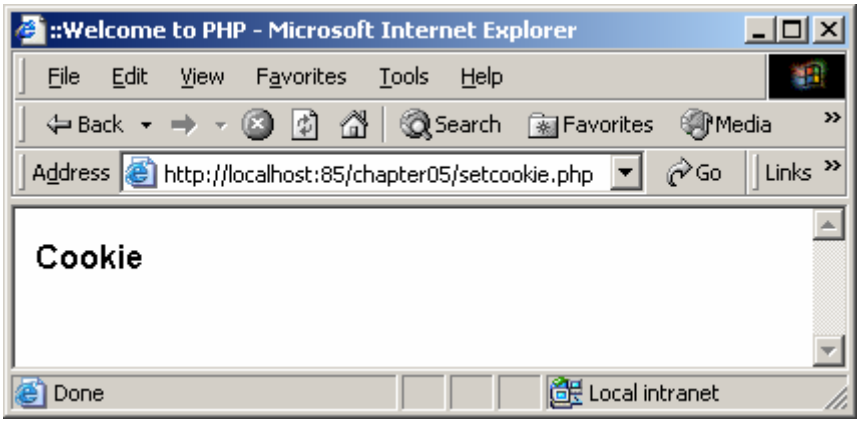
2. COOKIE

Cookie được xem như session, tuy nhiên chúng lưu trữ thông tin trên trình khách. Để sử dụng Cookie, bạn sử dụng hàm setcookie để gán giá trị như ví dụ 5-8.

Ví dụ 5-8: Gán giá trị cho cookie

```
<?php
    setcookie("huukhang","Computer Learning Center");
?>
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Cookie</h4>
</BODY>
</HTML>
```

Khi người sử dụng triệu gọi trang setcookie.php kết quả trả về như hình 5-8.



Hình 5-8: Đắng ký cookie

Ngài ra, bạn có thể gán giá trị cookie bằng session. Chẳng hạn, chúng ta sử dụng hàm session_set_cookie_params để gán cookie như ví dụ 5-9.

Ví dụ 5-9: Gán cookie bằng session

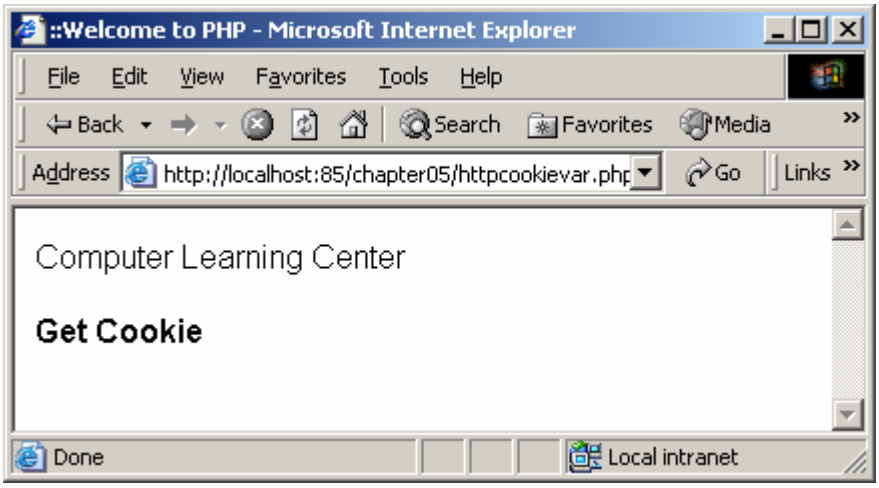
```
<?php
     session_start();
     $myvalue="Online Recruitment";
     session_set_cookie_params($myvalue);
?>
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Session-Cookie</h4>
</BODY>
</HTML>
```

Bằng cách sử dụng $HTTP_COOKIE_VARS để lấy giá trị của cookie trước đó trong trang httpcookievar.php như ví dụ 5-10.

Ví dụ 5-10: Sử dụng $HTTP_COOKIE_VARS

```
<?php
     echo $HTTP_COOKIE_VARS["huukhang"];
?>
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Get Cookie</h4>
</BODY>
</HTML>
```

Kết quả trình bày như hình 5-10.



Hình 5-10: Dùng $HTTP_COOKIE_VARS

Bằng cách sử dụng hàm session_get_cookie_params để lấy giá trị của cookie trước đó trong trang sessiongetcookie.php như ví dụ 5-11.

Ví dụ 5-11: Sử dụng session_get_cookie_params

```
<?php
     session_start();
     $myvalue= session_get_cookie_params();
     echo $myvalue[1];
?>
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Get Cookie</h4>
</BODY>
</HTML>
```

3. KẾT LUẬN

Trong bài này, chúng ta tìm hiểu cách sử dụng biến session và cookie.

Môn học: PHP

Bài 6

Bài học này chúng ta sẽ làm quen cách khai báo hàm, chèn tập tin và tập tin dùng chung:

- ✓ *Cách khai báo hàm.*
- ✓ *Xây dựng tập tin định dạng nội dung*
- ✓ *Tập tin dùng chung*

1. KHAI BÁO HÀM TRONG PHP

Hàm do người sử dụng định nghĩa cho phép bạn xử lý những tác vụ thường lặp đi lặp lại trong ứng dụng.

Để khai bao hàm, bạn sử dụng từ khoá function với cú pháp tương tự như sau:

```
function functioname($parameter)
{
    return value;
}
```

Trong trường hợp hàm không có giá trị trả về thì hàm được xem như thủ tục. Ngoài ra, bạn có thể khai báo tham số tuỳ chọn bằng cách gán giá trị mặc định cho tham số. Ví dụ chúng ta khai báo:

```
function functioname($parameter1, $parameter2=10 )
{
    return value;
}
```

Đối với trường hợp này thì tham số $parameter1 là tham số bắt buộc và tham số $parameter2 là tham số tuỳ chọn, khi gọi hàm nếu không cung cấp tham số cho $parameter2 thì tham số này có giá trị là 10.

Ví dụ, bạn khai báo trang function.php có hàm getResult nhận hai số và phép toán sau đó tuỳ thuộc vào phép toán hàm trả về kết quả. Nếu người sử dụng không cung vấp phép toán thì mặc định là phép toán +.

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Function</h4>
<?php
     function getResult($number1, $number2,$operator="+")
     {
       $result=0;
       switch($operator)
       {
         case "+":
           $result=$number1+$number2;
           break;
         case "-":
```

```
            $result=$number1-$number2;
            break;
          case "*":
            $result=$number1*$number2;
            break;
          case "/":
            if($number2!=0)
                  $result=$number1/$number2;
            else
                  $result=0;
            break;
          case "%":
            if($number2!=0)
                  $result=$number1%$number2;
            else
                  $result=0;
            break;
        }
        return $result;
      }
      echo "result of default operator: ".getResult(10,20);
      echo "<br>";
      echo "result of * operator: ".getResult(10,20,"*");
?>
</BODY>
</HTML>
```

Nếu muốn định nghĩa function không có giá trị trả về, bạn có thể khai báo trong trang void.php như ví dụ sau:

```
...
      function calloperator()
      {
        echo "result of default operator: ".getResult(10,20);
        echo "<br>";
        echo "result of * operator: ".getResult(10,20,"*");
      }
            calloperator();
      ?>
</BODY>
</HTML>
```

Trong trường hợp truyền tham số như tham biến, bạn sử dụng ký hiệu & trước tham số, chẳng hạn chúng ta khai báo hàm có tham biến có tên average như trong trang reference.php như sau:

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP</TITLE>
</HEAD>
<BODY>
<h4>Function</h4>
<?php
      function getAmount($quantity, $price,&$average)
      {
        $result=0;
        $result=$quantity*$price;
        $average=$result*6/12;
        return $result;
      }
      $bq=0;
      echo "result is : ".getAmount(10,20,$bq);
      echo "<br>";
```

```
        echo "result of Average is : ".$bq;
        echo "<br>";
        function getAmounts($quantity, $price,$average)
        {
           $result=0;
           $result=$quantity*$price;
           $average=$result*6/12;
           return $result;
        }
        $bq=0;
        echo "result is : ".getAmounts(10,20,$bq);
        echo "<br>";
        echo "result of Average is : ".$bq;
?>
</BODY>
</HTML>
```

Trong trường hợp trên thì hàm getAmount có tham số $average là tham biến còn hàm getAmounts có tham số $average là tham trị, và kết quả trả về của biến $bq khi gọi hàm getAmount là 100 trong khi đó giá trị của biến này trong hàm getAmounts là 0.

2. XÂY DỰNG TẬP TIN ĐỊNH DẠNG NỘI DUNG

Khi trình bày nội dung trên trang *HTML* hay trang *PHP*, để thống nhất định dạng chuỗi trong thẻ *body* hay thẻ *div* chẳng hạn bạn cần khai báo thẻ *style* trong thẻ *<head>*.

```
<style>
   A {
     COLOR: #003063;
     TEXT-DECORATION: none
   }
   A:hover {
     COLOR: #003063;
     TEXT-DECORATION: underline
   }
   A:link {
     FONT-WEIGHT: bold;
     COLOR: red;
     TEXT-DECORATION: none
   }
   A:visited {
     FONT-WEIGHT: bold;
     COLOR: black;
     TEXT-DECORATION: none
   }
   .title {
     FONT-WEIGHT: normal;
     FONT-SIZE: 22px
   }
   .text{
     FONT: 11px Arial, Helvetica, sans-serif
   }
</style>
```

Trong đó, A tương ứng với liên kết (chuỗi trong thẻ <a>) có định dạng ứng với trường hợp liên kết, di chuyển con chuột, chọn liên kết.

```
   A {
     COLOR: #003063;
     TEXT-DECORATION: none
   }
   A:hover {
```

```
    COLOR: #003063;
    TEXT-DECORATION: underline
  }
  A:link {
    FONT-WEIGHT: bold;
    COLOR: red;
    TEXT-DECORATION: none
  }
  A:visited {
    FONT-WEIGHT: bold;
    COLOR: black;
    TEXT-DECORATION: none
  }
```

Chẳng hạn, chúng ta khai báo trang *PHP* với nội dung được áp dụng với kiểu định dạng khai báo trong thẻ *style* như vú dụ 6-1.

Ví dụ 6-1: Khai báo thẻ style

```
<%@ page contentType="text/html; charset=UTF-8" %>
<html>
<head>
<title>Style trong PHP</title>
<style>
A {
     COLOR: #003063;
     TEXT-DECORATION: none
}
A:hover {
     COLOR: #003063;
     TEXT-DECORATION: underline
}
A:link {
     FONT-WEIGHT: bold;
     COLOR: red;
     TEXT-DECORATION: none
}
A:visited {
     FONT-WEIGHT: bold;
     COLOR: black;
     TEXT-DECORATION: none
}
.title {
     FONT-WEIGHT: normal;
     FONT-SIZE: 22px;
     COLOR: #003063;
}
.text{
     FONT: 11px Arial, Helvetica, sans-serif
}
</style>
</head>
<body>
     <h4>Style Tag</h4>
     <TABLE cellSpacing=0 cellPadding=0
          width="100%" border=0>
     <TR>
        <TD vAlign=top class=title>
        *** Quản Trị SQL Server 2000 ***          </TD>
     </TR>
     <TR>
        <TD class=text>
        <div align=justify>
        Tìm hiểu cách cài đặt, cấu hình, quản trị,
        backup & restore, import & export, thiết
```

```
            kế, lập trình, tự động hoá tác vụ quản trị,
            bản sao dữ liệu, bảo mật và chống thâm nhập
            dữ liệu bằng.
            <b>SQL Injection</b>.</div>
            </TD>
        </TR>
        <TR><TD><hr size=1 color=red></TD</TR>
        <TR><TD>Welcome to
        <a href="www.huukhang.com" class=>
        www.huukhang.com</a></TD
        </TR>
      </TABLE>
</body>
</html>
```

Khi triệu gọi trang *style.PHP* trên trình duyệt, nội dung của trang *web* được định dạng theo thẻ *style* như hình 6-1.

Hình 6-1: Áp dụng thẻ style

Tương tự như vậy khi bạn muốn thống nhất nội dung trong những thẻ khác của một trang *web* thì khai báo một định dạng trong thẻ *style*. Tuy nhiên, khi đặt tên trùng với thẻ *HTML*, mọi thẻ đó trong trang sẽ cùng chung một định dạng. Chẳng hạn, bạn khai báo định dạng cho thẻ *td* như sau:

```
TD {
     FONT: 10px Arial, Helvetica, sans-serif
}
```

Mọi nội dung trình bày trong thẻ *td* sẽ có định dạng như trên. Nếu bạn muốn có định dạng khác thì khai báo thuộc tính *class* cho thẻ *td* đó, ví dụ sử dụng định dạng khác cho thẻ *td*:

```
<td class=text>ABC</td>
```

Thay vì chuỗi *ABC* sẽ có định dạng là *FONT: 10px Arial*, *Helvetica, sans-serif* thì chúng sẽ có định dạng của *FONT: 11px Arial, Helvetica, sans-serif*.

Chú ý rằng, trong mỗi trang *web* bạn phải khai báo thẻ *style* và định nghĩa thống nhất cho các thẻ. Khi có sự thay đổi bạn phải thay đổi trong mọi trang *web*. Để sử dụng chung cho mọi trang *web* trong ứng dụng, bạn cần xây dựng một tập tin *style*, tập tin được biết đến với tên gọi *custom style sheet* (*css*).

Bất kỳ trang *web* nào trong ứng dụng, muốn áp dụng kiểu định dạng trong tập tin *css* thì khai báo liên kết tập tin *css* bằng thẻ *link*.

Ví dụ, chúng ta khai báo tập tin *style.css* bao gồm các định dạng như ví dụ 6-2.

Ví dụ 6-2: Khai báo tập tin css

```
A {
     COLOR: #003063;
     TEXT-DECORATION: none
}
A:hover {
     COLOR: #003063;
     TEXT-DECORATION: underline
}
A:link {
     FONT-WEIGHT: bold;
     COLOR: red;
     TEXT-DECORATION: none
}
A:visited {
     FONT-WEIGHT: bold;
     COLOR: black;
     TEXT-DECORATION: none
}
.title {
     FONT-WEIGHT: bold;
     FONT-SIZE: 14px;
     COLOR: #003063;
}
.text{
     FONT: 11px Arial, Helvetica, sans-serif
}
```

Sau đó trong trang *PHP*, bạn khai báo liên kết tập tin này bằng thẻ *link*, nếu muốn áp dụng định dạng này trong mỗi thẻ *HTML* bạn sử dụng thuộc tính *class* như khai báo định dạng của thẻ *style* ngay trong trang đó như ví dụ 6-3.

Ví dụ 6-3: Khai báo sử dụng tập tin css

```
<html>
<head>
<title>
     Welcome to Link Style Sheet File
</title>
<LINK href="style.css" rel=stylesheet>
     <META http-equiv=Content-Type
     content="text/html; charset=utf-8">
</head>
<body>
     <h4>Style File</h4>
     <TABLE cellSpacing=0 cellPadding=0
          width="100%" border=0>
```

```
        <TR>
           <TD vAlign=top class=title>
           *** Quản Trị SQL Server 2000 ***          </TD>
        </TR>
        <TR>
           <TD class=text>
           <div align=justify>
           Tìm hiểu cách cài đặt, cấu hình, quản trị,
           backup & restore, import & export, thiết
           kế, lập trình, tự động hoá tác vụ quản trị,
           bản sao dữ liệu, bảo mật và chống thâm nhập
           dữ liệu bằng.
           <b>SQL Injection</b>.</div>
           </TD>
        </TR>
        <TR><TD><hr size=1 color=red></TD</TR>
        <TR><TD>Welcome to
        <a href="www.huukhang.com" class=>
        www.huukhang.com</a></TD
        </TR>
      </TABLE>
   </body>
   </html>
```

Triệu gọi trang *includestyle.php* trên trình duyệt như hình 6-3, màu và kích thước *font* cùng với kiểu chữa của nội dung không thay đổi so với *style.php*, bởi vì phần thẻ *style* được tách ra thành tập tin *style.css*, sau đó dùng thẻ *link* để liên kết tập tin *css* này vào trang *PHP* trở lại.

Hình 6-3: Liên kết tập tin css

Chú ý rằng, nếu khai báo thuộc tính class trong thẻ <*table*> thì những nội dung trong thẻ <*table*> sẽ có định dạng *theo* định dạng khai báo trong thuộc tính *class*. Tương tự, nếu khai báo thuộc tính *class* trong thẻ <*tr*> thì nội dung trong thẻ <tr> sẽ có định dạng giống như định dạng khai báo trong thông tin *class*.

3. THỐNG NHẤT KÍCH THƯỚC CỦA MOI TRANG PHP

Khi xây dựng ứng dụng *web* chuyên nghiệp, điều đầu tiên bạn nên quan tâm là sự thống nhất về kích thước của các phần trên trang *web*. Điều này có nghĩa là khi người sử dụng thay đổi trang *web* khi duyệt, phần *top, left, right, bottom* có kích thước như nhau.

Để làm điều này, bạn chia trang *web* ra thành 5 phần: *top*, *left*, *right, body* và *bottom*.

Phần top thường trình bày các thuộc tính như quản cáo (*baner*), *logo* (biểu tượng của công ty), *menu* (thực đơn của ứng dụng) và một số thông tin khác.

Phần *left* là thông tin về các *menu* phụ hay còn gọi là *menu* của *menu* chính, bên cạnh *menu* con này trang *web* thường có các liên kết về liên hệ, quảng cáo, *mailing list* (đăng ký *email*), gởi đến bạn bè (*send to friend*), ...

Đối với phần *right*, thường là phần giới thiệu về các thông đặc biệt và quảng cáo, chẳng hạn đối với ứng dụng bán sách, phần *right* thường là danh sách các nhóm sách bán chạy, sắp phát hành, ...

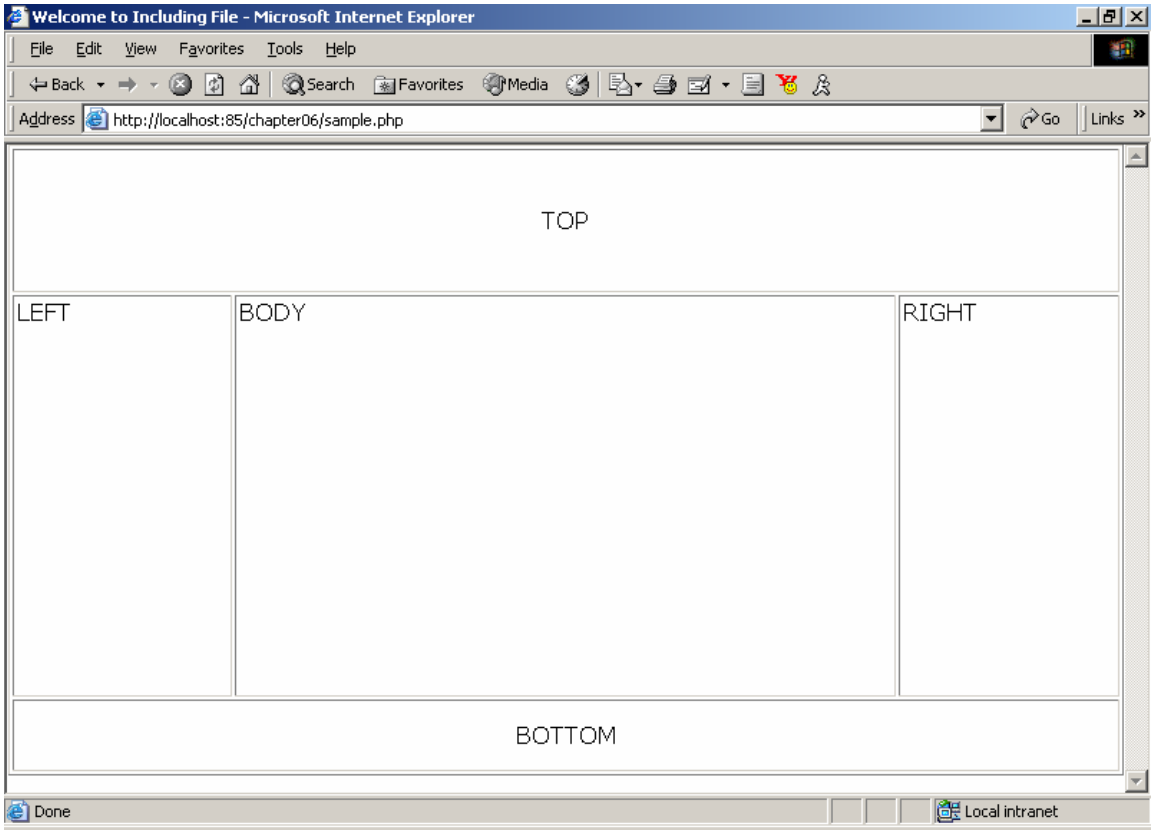
Phần *bottom* thường thông tin liên lạc của công ty, chủ nhân của *web site* và bản quyền. Ngoài ra, phần *bottom* đôi khi là danh sách các *menu* con khác.

Tóm lại, tuỳ thuộc vào ý tưởng thiết kế mỗi phần như trên bao gồm các thuộc tính mà nhà thiết kế cần trình bày sao cho phù hợp. Tuy nhiên, phần *body* là phần trình bày nội dung chính của mỗi trang *web*. Ngoài ra, tuỳ vào từng trường hợp cụ thể, trang *web* có thể không có phần *left* và *right*.

Như vậy, chúng ta sẽ chia trang *web* ra thành 5 phần, phần *body* chính là phần chính của trang *web* đó, còn 4 phần còn lại được chèn vào khi có nhu cầu.

Chẳng hạn, có những trang *web* do thông tin trình bày trong phần *body* nhiều, nên cần không gia lớn hơn, bạn có thể không cần sử dụng hai phần *left* và *right*.

Để làm điều này, trước tiên chúng ta thiết kế trang *sample.php* có 5 phần như hình 6-3.



Hình 6-3: Trang sample.php

Lưu ý:

- *Tạo một table gồm 3 hàng 3 cột và khai báo border=1 để dễ canh lề sau đó bạn có thể khai báo lại thuộc tính này bằng 0.*
- *Phần top và bottom là một hàng và merge 3 cột thành 1.*
- *Bên trong mỗi phần có thể có một hay nhiều thẻ table khác.*
- *Có thể không có phần left và right nhưng bắt buộc phần top và bottom phải có.*
- *Bạn có thể sử dụng chiều rộng của table theo kích thước tương đối (%) hay số chỉ định, đối với màn hình 600*800 thì chiều rộng thường sử dụng là 780, khi người sử dụng chọn độ phân giải của màn hình lớn hơn thì kích thước của table này không thay đổi, trong khi đó nội dung sẽ phủ đầy màn hình khi bạn khai báo kích thước theo 100%.*

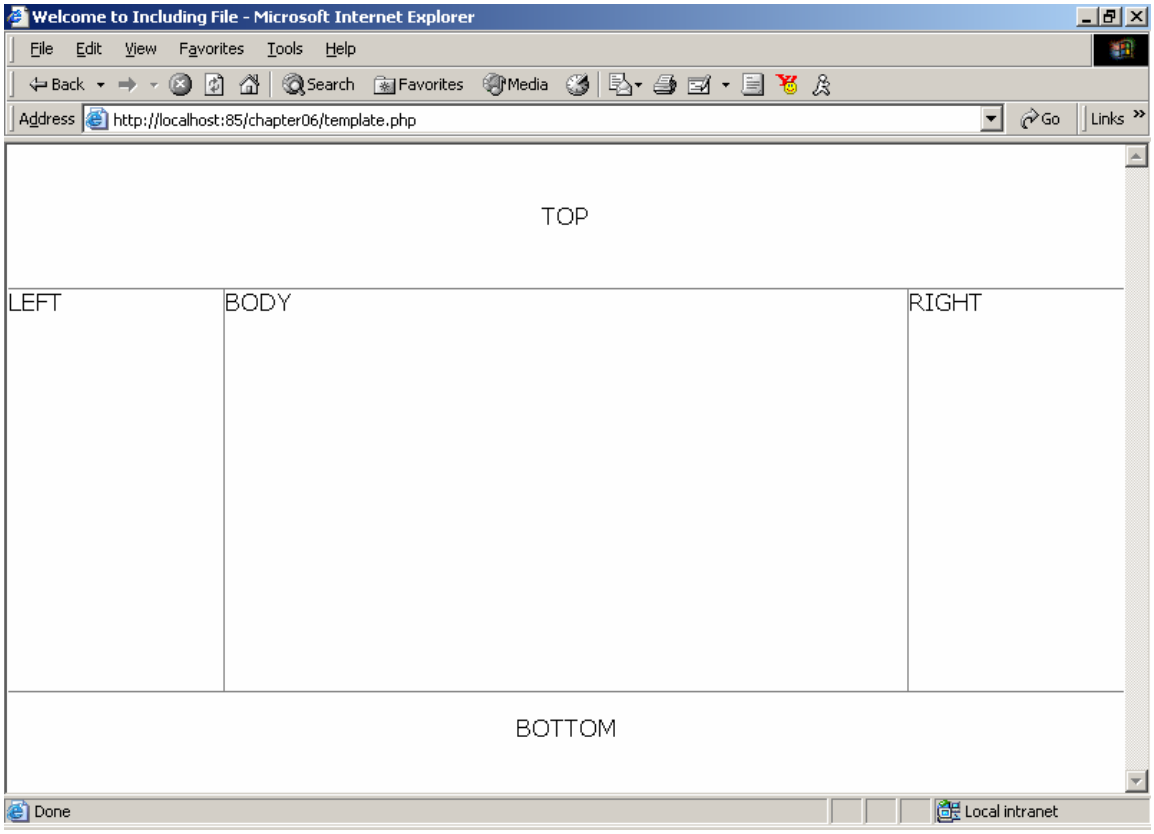
Để có giao diện như trang *sample.php* như trên, bạn có thể khai báo như ví dụ 6-3.

Ví dụ 6-3: Nội dung trang sample.PHP

```
<html>
<head>
<title>
     Welcome to Including File
</title>
<LINK href="style.css" rel=stylesheet>
     <META http-equiv=Content-Type
     content="text/html; charset=utf-8">
</head>
<body bottomMargin=0 leftMargin=0
     topMargin=0 rightMargin=0>
```

```
        <TABLE cellSpacing=2 cellPadding=2
            width="778" border=1 align=center>
         <TR HEIGHT="100">
           <TD  Align=center colspan=3>
             TOP
           </TD>
         </TR>
         <TR HEIGHT="280">
           <TD vAlign=top width="20%">
                LEFT
           </TD>
           <TD vAlign=top width="60%">
                BODY
           </TD>
           <TD vAlign=top width="20%">
                RIGHT
           </TD>
         </TR>
         <TR HEIGHT="50">
         <TD colspan=3 align=center>
              BOTTOM
         </TD
         </TR>
        </TABLE>
    </body>
    </html>
```

Trong trường hợp bạn muốn có đường phân cách giữa mỗi phần bằng *image*, bạn có thể khai báo lại trang *sample.php* có 5 hàng và 5 cột như *template.php* như hình 6-4.



Hình 2-4: Phân cách có viềng

Để trình bày trang *tempale.PHP* như hình 6-4, bạn khai báo nội dung trang này như ví dụ 6-4.

Ví dụ 6-4: Khai báo template.php

```
<html>
<head>
<title>
     Welcome to Including File
</title>
<LINK href="style.css" rel=stylesheet>
     <META http-equiv=Content-Type
     content="text/html; charset=utf-8">
</head>
<body bottomMargin=0 leftMargin=0
     topMargin=0 rightMargin=0>
     <TABLE width="778" border=0 cellSpacing=0
      cellPadding=0 align=center>
     <TR HEIGHT="100">
       <TD Align=center colspan=5>
          TOP
       </TD>
     </TR>
     <!---Khai báo đường phân cách--->
     <TR HEIGHT="1">
       <TD colspan=5 bgcolor=gray></TD>
     </TR>
     <TR HEIGHT="280">
       <TD vAlign=top width="150">LEFT</TD>
       <!---Khai báo đường phân cách--->
```

```
        <TD bgcolor=gray width="1"></TD>
        <TD vAlign=top width="476">BODY</TD>
        <!---Khai báo đường phân cách--->
        <TD bgcolor=gray width="1"></TD>
        <TD vAlign=top width="150">RIGHT</TD>
      </TR>
      <!---Khai báo đường phân cách--->
      <TR HEIGHT="1">
        <TD colspan=5 bgcolor=gray></TD>
      </TR>
      <TR HEIGHT="50">
      <TD colspan=5 align=center>
        BOTTOM
      </TD
      </TR>
     </TABLE>
</body>
</html>
```

Sau đó tách trang *template.php* này thành 5 trang khác nhau được đặt tên tương ứng là *top.htm, left.htm, right.htm* và *bottom.htm*, trong đó phần *body* tương ứng với trang *templates.php*.

Để khai báo chèn tập tin trong trang *PHP*, bạn sử dụng cú pháp như sau:

```
<?php
include(”filename”);
?>
```

Hay

```
<?php
require(”filename”);
?>
```

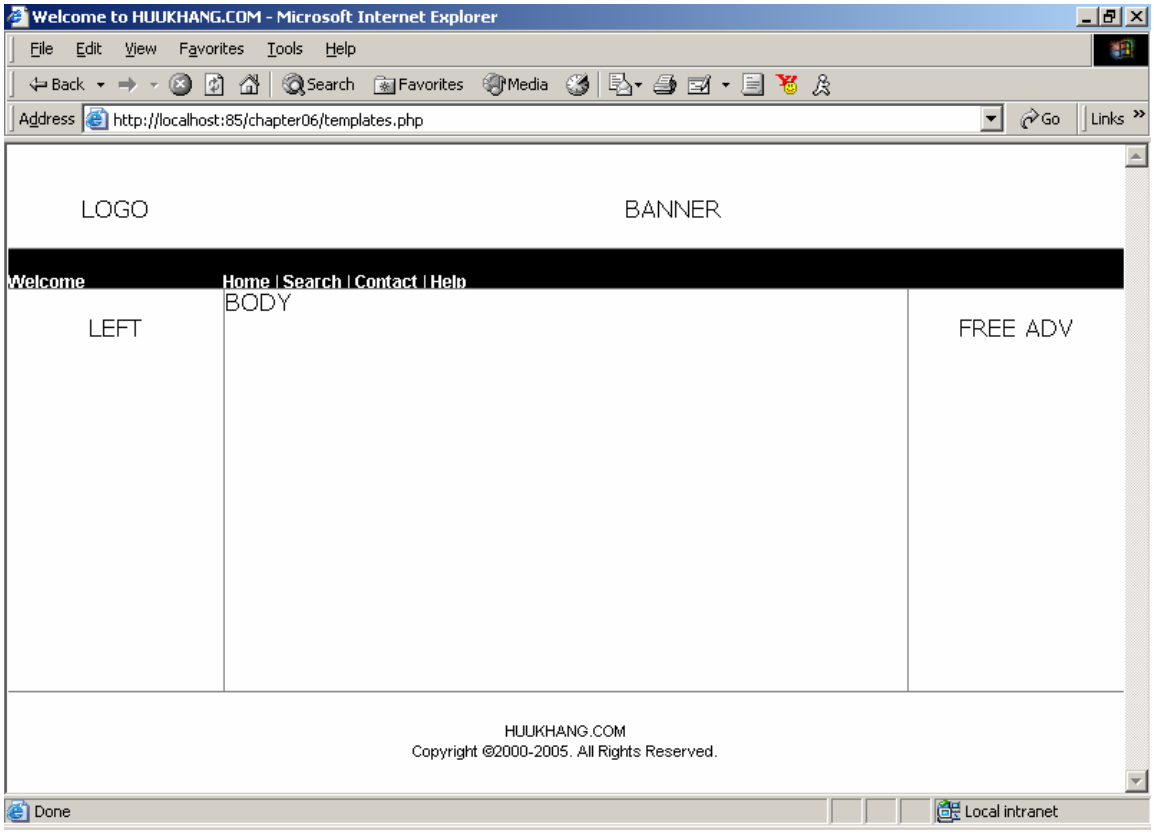
Trong đó trang *templates.PHP* khai báo chèn *top.htm, left.htm, right.htm* và *bottom.htm* như ví dụ 6-5.

Ví dụ 6-5: Khai báo chèn tập tin trong templates.php

```
<html>
<head>
<title>
     Welcome to HUUKHANG.COM
</title>
<LINK href="style.css" rel=stylesheet>
     <META http-equiv=Content-Type
     content="text/html; charset=utf-8">
</head>
<body bottomMargin=0 leftMargin=0
     topMargin=0 rightMargin=0>
     <TABLE width="778" border=0 cellSpacing=0
      cellPadding=0 align=center>
     <TR HEIGHT="100">
       <TD  Align=center colspan=5>
          <?php include("top.htm")?>
       </TD>
      </TR>
     <!---Khai báo đường phân cách--->
     <TR HEIGHT="1">
       <TD colspan=5 bgcolor=gray></TD>
     </TR>
```

```
        <TR HEIGHT="280">
           <TD vAlign=top width="150">
           <?php include("left.htm")?>
           </TD>
           <!---Khai báo đường phân cách--->
           <TD bgcolor=gray width="1"></TD>
           <TD vAlign=top width="476">BODY</TD>
           <!---Khai báo đường phân cách--->
           <TD bgcolor=gray width="1"></TD>
           <TD vAlign=top width="150">
           <?php include ("right.htm")?>
           </TD>
        </TR>
        <!---Khai báo đường phân cách--->
        <TR HEIGHT="1">
           <TD colspan=5 bgcolor=gray></TD>
        </TR>
        <TR HEIGHT="50">
        <TD colspan=5 align=center>
           <?php include("bottom.htm")?>
        </TD
        </TR>
       </TABLE>
    </body>
    </html>
```

Khi triệu gọi trang *templates.php*, nội dung của 4 tang *left.htm*, *right.htm, top.htm, bottom.htm* chèn vào trang *templates.php* như hình 6-5.



Hình 6-5: Trang templates.php sau khi chèn

Trong đó, nội dung của trang *top.htm* định nghĩa tương tự như ví dụ 6-5-1.

Ví dụ 6-5-1: Nội dung trang top.htm

```
<TABLE width="100%" border=0 cellSpacing=0
    cellPadding=0 HEIGHT="100%" align=center>
       <TR >
         <TD width="150" Align=center>
       LOGO
     </TD>
     <TD  Align=center>
       BANNER
     </TD>
    </TR>
    <TR HEIGHT="1">
         <TD colspan=2 bgcolor=gray></TD>
    </TR>
    <TR HEIGHT="20%" bgcolor=black class=menu>
         <TD width="150" >
       Welcome
     </TD>
     <TD>
       Home | Search | Contact | Help
     </TD>
    </TR>
</TABLE>
```

Nội dung của tập tin *left.htm* được định nghĩa tương tự như ví dụ 6-5-2.

Ví dụ 6-5-2: Nội dung trang left.htm

```
<TABLE width="100%" border=0 cellSpacing=0
     cellPadding=0 HEIGHT="100%" align=center>
        <TR >
          <TD width="150" Align=center>
          LEFT
     </TD>
     </TR>
</TABLE>
```

Nếu có sử dụng trang right.htm thì nội dung của tập tin này được định nghĩa tương tự như ví dụ 6-5-3.

Ví dụ 6-5-3: Nội dung trang right.htm

```
<TABLE width="100%" border=0 cellSpacing=0
     cellPadding=0 HEIGHT="100%" align=center>
        <TR >
          <TD width="150" Align=center>
          FREE ADV
     </TD>
     </TR>
</TABLE>
```

Tương tự như vậy, trang *bottom.htm* có nội dung như ví dụ 6-5-4.

Ví dụ 6-5-4: Nội dung trang bottom.htm

```
<TABLE width="100%" border=0 cellSpacing=0
     cellPadding=0 HEIGHT="100%" align=center>
        <TR class=text>
          <TD Align=center>
          HUUKHANG.COM<br>
        Copyright ©2000-2005.
        All Rights Reserved.
     </TD>
     </TR>
</TABLE>
```

Chú ý rằng, trong mỗi trang khai báo chèn không có các thẻ đóng và mở *html*, *body* bởi khi chèn thì nội dung của tập tin được chèn sẽ được chèn vào tập tin bị chèn và trong tập tin bị chèn đã có hai thẻ này.

Kịch bản trình chủ PHP hỗ trợ các tập tin được chèn với các tên mở rộng như *htm, PHP, inc, lib, html*. Do thực chất của việc khai báo chèn là chèn đoạn mã trong tập tin chèn vào tập tin bọ chèn, trong trường hợp này trang chèn *htm* hay *PHP* đều giống nhau đó là lý do tại sao các trang chèn ở trên đều có tên mở rộng là *htm*.

Tuy nhiên, khi bạn gọi trang chèn này một mình ví dụ *tom.htm*, nếu bên trong có mã *PHP* thì mã đó không được thông dịch. Nếu những trang chèn này có nhu cầu gọi một mình thì bạn có thể chuyển chúng thành trang *PHP* thay vì *htm* như đã trình bày.

Sau khi có được trang *templates.php*, bạn có thể sử dụng trang này là mẫu cho các trang khác bằng cách *save as* thành các trang *PHP* khác khi lập trình. Khi khai báo chèn tập tin, bạn có thể sử dụng đường dẫn tương đối hoặc tuyệt đối của tập tin chèn so với ập tin bị chèn.

4. TẬP TIN DÙNG CHUNG

Ngoài cách chèn ở trên, nếu bạn có những hàm sử dụng chung cho các trang PHP khác thì bạn khai báo thành một trang PHP khác sau đó dùng cú pháp chèn tập tin để chèn chúng vào khi có nhu cầu.

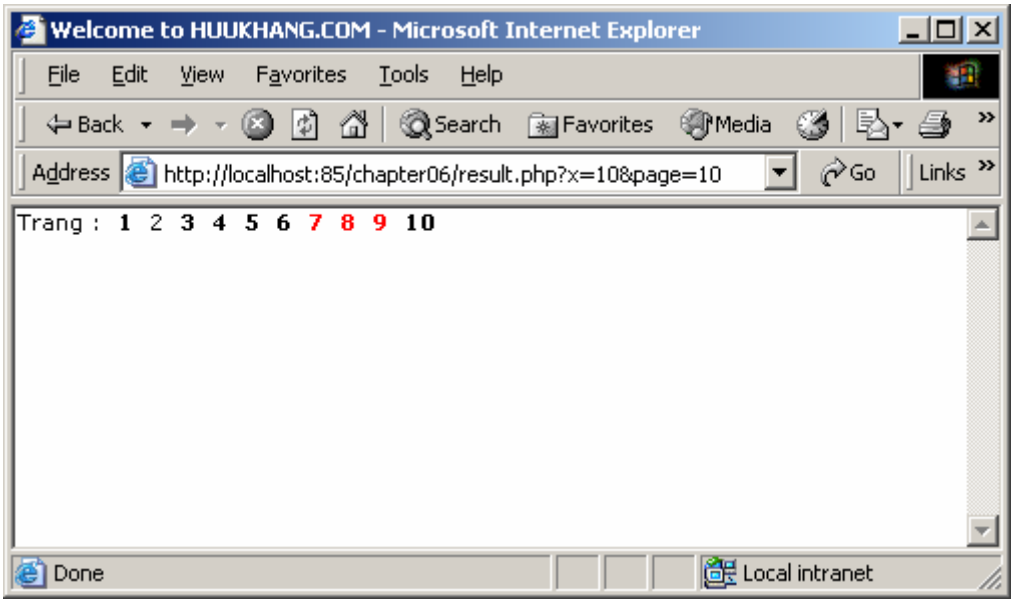
Ví dụ trong trường hợp này chúng ta muốn sử dụng chung hàm có tên getPaging nhận 5 tham số $totalRows (tổng số mẩu tin), $curPg (số trang hiện hành), $pg (số trang trình bày), $re (số mẩu tin trên 1 trang), $file (trang php cần gọi) trong tập tin paging.php.

```
<?php
function paging($totalRows,$curPg,$pg,$re,$file)
{
$paging="";
$mxR = $re;
$mxP = $pg;
if($totalRows%$mxR==0)
     $totalPages = (int)($totalRows/$mxR);
else
     $totalPages = (int)($totalRows/$mxR+1);
$curRow = ($curPg-1)*$mxR+1;
if($totalRows>$mxR)
{
$start=1;
$end=1;
$paging1 ="";
for($i=1;$i<=$totalPages;$i++)
{
if(($i>((int)(($curPg-1)/$mxP))* $mxP) && ($i<=((int)(($curPg-
1)/$mxP+1))* $mxP))
     {
       if($start==1) $start=$i;
       if($i==$curPg)
         $paging1 .= $i."  ";
       else
         {
         $paging1 .= "<a class=lslink href='$file";
         $paging1 .="&page=".$i."'>".$i;
         $paging1 .="</a>  ";
         }
       $end=$i;
     }
}
$paging.= "Trang :  " ;
if($curPg>$mxP)
     {
     $paging .="<a class=lslink href='$file";
     $paging .="&page=".($start-1);
     $paging .="'>Previous</a>  ";
     }
$paging.=$paging1;
if((((($curPg-1)/$mxP+1)*$mxP) < $totalPages)
     {
     $paging .= "<a class=lslink href='$file";
     $paging .="&page=".($end+1);
     $paging .="'>Next</a>  ";
     }
}
return $paging;
}
?>
```

Sau đó khai báo trang result.php, chèn tập tin paging.php và gọi hàm getPaging nhưu sau:

```
<html>
<head>
<title>
     Welcome to HUUKHANG.COM
</title>
<LINK href="style.css" rel=stylesheet>
     <META http-equiv=Content-Type
     content="text/html; charset=utf-8">
</head>
<body bottomMargin=0 leftMargin=0 topMargin=0 rightMargin=0>
<?php
     include("paging.php");
     echo paging(47,2,10,5,"result.php?x=10");
?>
</body>
</html>
```

Kết quả trả về như hình 6-6 sau



Hình 6-6: Hàm dùng chung

5. KẾT LUÂN

Trong bài này, chúng ta tìm hiểu cách khai báo hàm, trang php và khai báo chèn tập tin.

Môn học: PHP

Bài 7

Bài học này chúng ta sẽ làm quen cách xử lý chuỗi, mảng, kiểu DataTime trong PHP:

- ✓ *Xử lý chuỗi*
- ✓ *Làm việc với mảng dữ liệu*
- ✓ *Kiểu DateTime*

1. XỬ LÝ CHUỖI

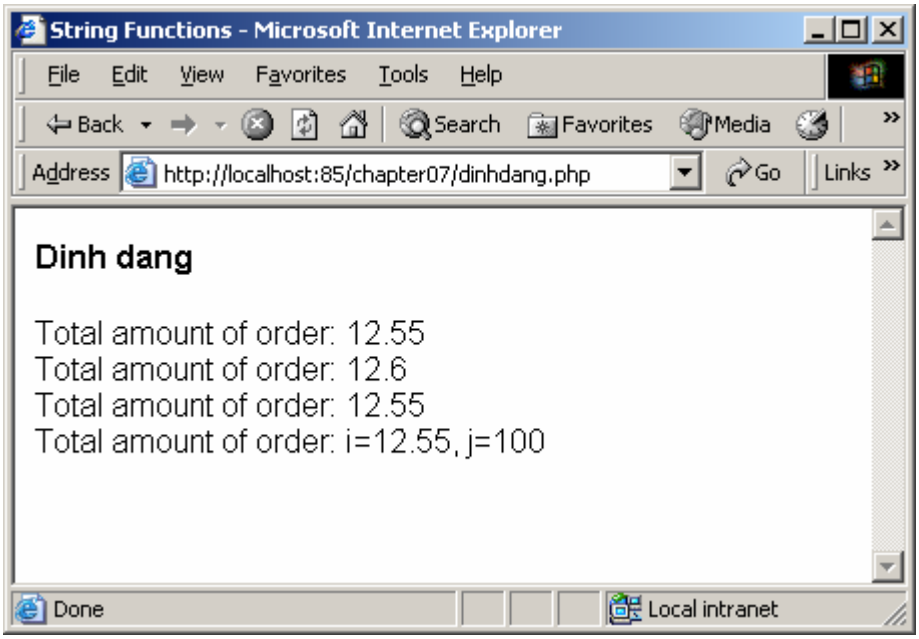
PHP là kịch bản được xem là tốt nhất cho xử lý chuỗi, bằng cách sử dụng các hàm xử lý chuỗi, bạn có thể thực hiện các ý định của mình khi tương tác cơ sở dữ liệu, tập tin hay dữ liệu khác.

1.1. Định dạng chuỗi

Khi xuất kết quả ra trình duyệt, bạn có thể sử dụng các định dạng chuỗi tương tự như ngôn ngữ lập trình C. Chẳng hạn, chúng ta in giá trị của biến $i trong trang dinhdang.php như ví dụ 7-1.

```
<html>
<head>
 <title>String Functions</title>
</head>
<body>
<h4>Dinh dang</h4>
<?php
$i=12.55;
$j=100;
echo "Total amount of order: $i<br>";
printf("Total amount of order: %.1f", $i);
echo "<br>";
printf("Total amount of order: %.2f", $i);
echo "<br>";
printf("Total amount of order: i=%.2f, j=%.0f", $i,$j);
?>
</body>
</html>
```

Kết quả xuất hiện như hình 7-1



Hình 7-1: Định dạng chuỗi in

Trong đó các định dạng được chia ra nhiêu loại tuỳ thuộc vào các ký tự bạn sử dụng.

```
% - Không yêu cầu tham số.
b - Trình bày dạng số integer và hiện thực dưới dạng binary.
c - Trình bày dạng số integer và hiện thực dưới dạng mã ASCII.
d - Trình bày dạng số integer và hiện thực dưới dạng decimal.
e - Trình bày dạng số logic và hiện thực dưới dạng 1.2e+2.
u - Trình bày dạng số integer và hiện thực dưới dạng decimal không dấu.
f - Trình bày dạng số float và hiện thực dưới dạng số chấm động.
o - Trình bày dạng số integer và hiện thực dưới dạng hệ số 10.
s - Trình bày dạng chuỗi.
x - Trình bày dạng số integer và hiện thực dưới dạng hệ số 16 với ký tự thường.
X - Trình bày dạng số integer và hiện thực dưới dạng hệ số 16 với ký tự hoa.
```

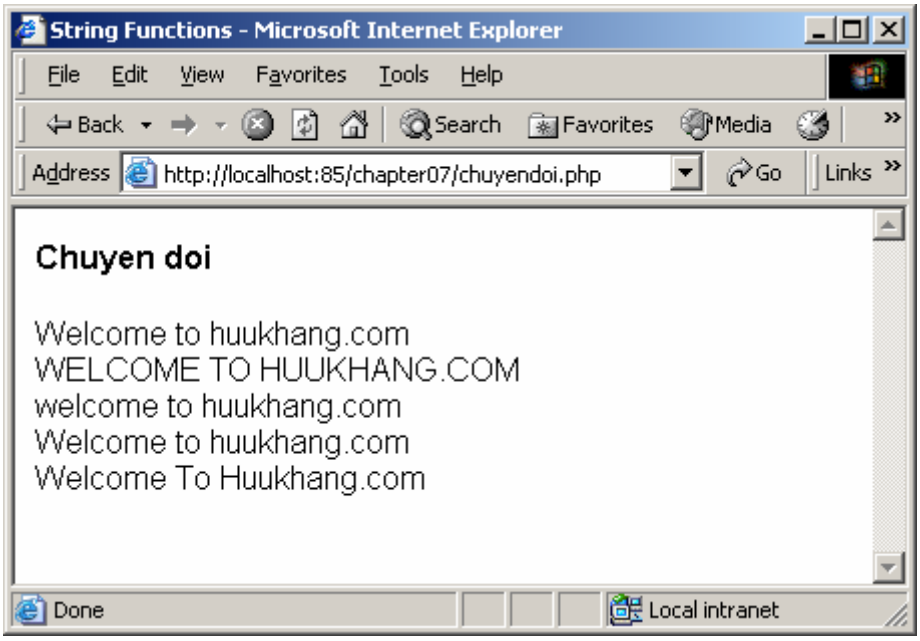
1.2. Hàm chuyển đổi chuỗi

Để chuyển đổi chuỗi ra ký tự hoa thường bạn sử dụng một trong 4 hàm như ví dụ 7-2 trong trang chuyendoi.php:

```
<html>
   <head>
    <title>String Functions</title>
   </head>
   <body>
   <h4>Chuyen doi</h4>
   <?php
      $str="Welcome to huukhang.com";
      echo $str;
      echo "<br>";
      echo strtoupper($str);
      echo "<br>";
      echo strtolower($str);
      echo "<br>";
      echo ucfirst($str);
      echo "<br>";
      echo ucwords($str);
      echo "<br>";
   ?>
```

```
</body>
</html>
```

Kết quả trình bày như hình 7-2.



Hình 7-2: Chuyển đổi chuỗi

1.3. Hàm tách hay kết hợp chuỗi

Để tách hay kết hợp chuỗi, bạn sử dụng một trong các hàm thường sử dụng như strtok, explode hay substr. Chẳng hạn, chúng ta sử dụng 4 hàm này trong ví dụ 7-4 trong trang tachchuoi.php.

```
<html>
<head>
 <title>String Functions</title>
</head>
<body>
<h4>Tach hop chuoi</h4>
<?php
   $string = "Xin chao ban da den voi huukhang.com";
   $str = $string;
   echo $string."<br>";
   $tok = strtok($string, " ");
   while ($tok)
   {
     echo "Word= $tok<br />";
     $tok = strtok(" \n\t");
   }
   echo $str."<br>";
   echo substr($str,24)."<br>";
   $a[]=array();
   $a=explode(" ",$str);
   while($i=each($a))
   {
```

```
        echo $i["value"]."<br>";
    }
?>
</body>
</html>
```

Kết quả trình bày như hình 7-4.

Hình 7-4: Hàm tách chuỗi

Trong trường hợp kết hợp giá trị của các phần tử của mảng thành chuỗi, bạn sử dụng hàm implode như ví dụ 7-5 trong trang kethop.php:

```
<html>
<head>
 <title>String Functions</title>
</head>
<body>
<h4>Ket hop chuoi</h4>
<?php
     $str = "Xin chao ban da den voi huukhang.com";
     $a[]=array();
     $a=explode(" ",$str);
     while($i=each($a))
     {
        echo $i["value"]."<br>";
     }
     $str=implode(" ",$a);
     echo $str;
```

```
?>
</body>
</html>
```

Kết quả trình bày như hình 7-5.

Hình 7-5: Hàm kết hợp chuỗi

1.4. Tìm kiếm và thay thế chuỗi

Để thay thế chuỗi, bạn sử dụng hàm str_replace, chẳng hạn trong trường hợp hợp bạn lấy giá trị từ thẻ nhập liệu, sau đó tìm kiếm nếu phát hiện dấu ‘ thì thay thế thành hai dấu nháy như trang replace.php.

```
<html>
<head>
 <title>String Functions</title>
</head>
<body>
<h4>That the chuoi</h4>
<?php
   $str="";
   if (isset($txtfullname))
      $str = $txtfullname;
   if($str != "");
      $str=str_replace("o","a",$str);
   echo $str."<br>";
?>
<form action=replace.php method=post>
fullname: <input name=txtfullname value="<?=$str?>"><br>
<input type=submit value=Submit>
</form>
```

```
</body>
</html>
```

Khi triệu gọi trang replace.php trên trình duyệt, bạn sẽ có kết quả như sau:

Hình 7-6: Hàm thay thế chuỗi

Ngoài ra, bạn có thể sử dụng các hàm như strpos (trả về vị trí chuỗi con trong chuỗi mẹ), ...

2. LÀM VIỆC VỚI MẢNG DỮ LIỆU

Như trong bài kiểu dữ liệu chúng ta đã làm quen với kiểu dữ liệu mảng, trong phần này chúng ta tiếp tục tìm hiểu các khai báo, truy cập và tương tác với tập tin từ mảng một chiều, hai chiều.

2.1. Mảng một chiều

Để khai báo mảng một chiều, bạn có thể sử dụng cú pháp như sau:

```
$arr=array();
$arrs=array(5);
```

Truy cập vào phần tử mảng, bạn có thể sử dụng chỉ mục của phần tử như sau:

```
$arr[0]=1;
$arrs[1]=12;
```

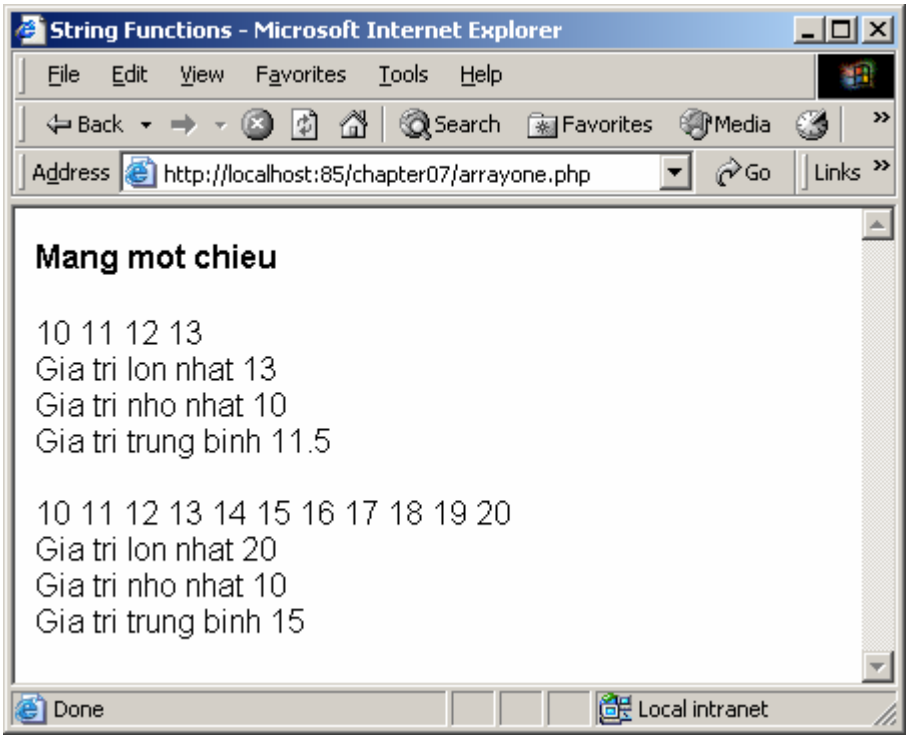
Lấy giá trị của phần tử mảng, bạn cũng thực hiện tương tự như trường hợp truy cập mảng phần tử.

```
echo $arr[0];
$x=$arrs[5];
```

Chẳng hạn, chúng ta khai báo mảng động và mảng có số phần tử cho trước, sau đó truy cập và lấy giá trị của chúng như ví dụ trong trang arrayone.php sau:

```
<html>
<head>
 <title>Array</title>
</head>
<body>
<h4>Mang mot chieu</h4>
<?php
 $i=0;
$myarr=array(1,2,3,4,5,6,7);
 $arr=array();
 $arrs=array(10);
 $arr[0]=10;$arr[1]=11;$arr[2]=12;$arr[3]=13;
 for($i=0;$i<sizeof($arr);$i++)
 {
   echo $arr[$i]." ";
 }
 echo "<br>";
 echo "Gia tri lon nhat ".max($arr)."<br>";
 echo "Gia tri nho nhat ".min($arr)."<br>" ;
 echo "Gia tri trung binh ".array_sum($arr) / sizeof($arr)."<br>" ;
 echo "<br>";
 for($i=0;$i<=10;$i++)
 {
   $arrs[$i]=10+$i;
 }
 for($i=0;$i<=10;$i++)
 {
   echo $arrs[$i]." ";
 }
 echo "<br>";
 echo "Gia tri lon nhat ".max($arrs)."<br>";
 echo "Gia tri nho nhat ".min($arrs)."<br>" ;
 echo "Gia tri trung binh ".array_sum($arrs) / sizeof($arrs)."<br>" ;
?>
</body>
</html>
```

Kết quả trình bày như hình 7-7 khi triệu gọi trang arrayone.php.



Hình 7-7: Khai báo và sử dụng mảng một chiều

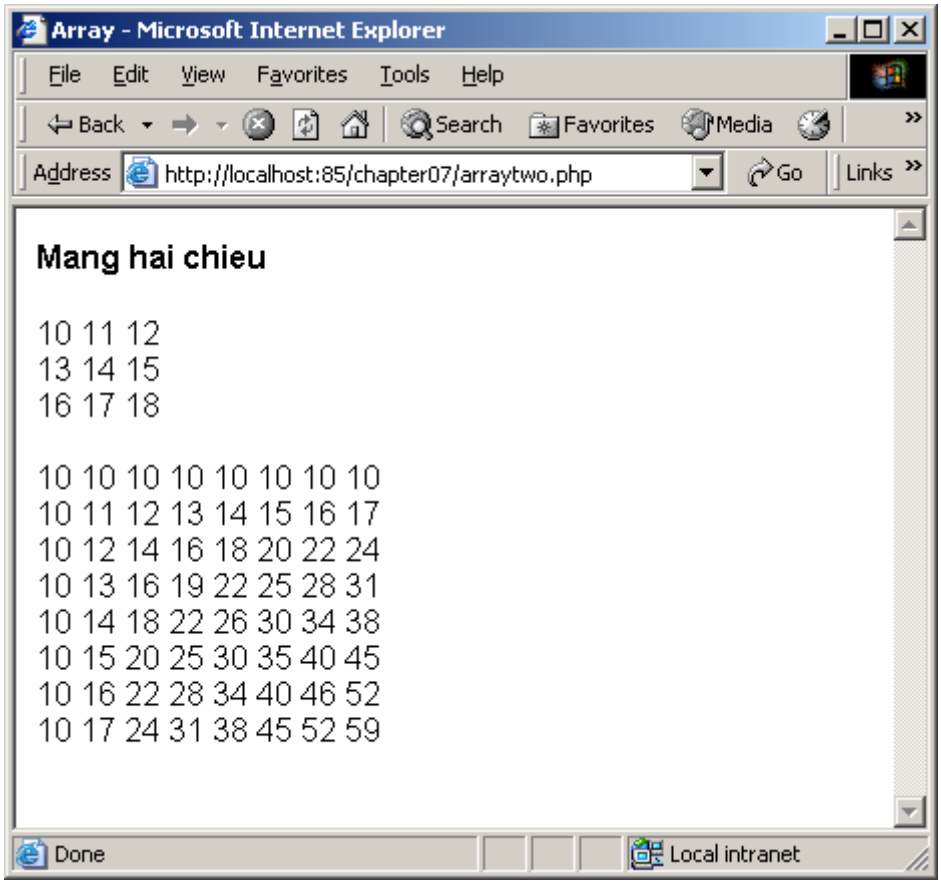
2.2. Mảng hai chiều

Tương tự như mảng một chiều, trong trường hợp làm việc mảng hai chiều bạn khai báo tương tự như trang arraytwo.php.

```
  <html>
<head>
 <title>Array</title>
</head>
<body>
<h4>Mang hai chieu</h4>
<?php
  $i=0;$j=0;
  $arr=array();
  $arr[0][0]=10;
  $arr[0][1]=11;
  $arr[0][2]=12;
  $arr[1][0]=13;
  $arr[1][1]=14;
  $arr[1][2]=15;
  $arr[2][0]=16;
  $arr[2][1]=17;
  $arr[2][2]=18;
  for($i=0;$i<sizeof($arr);$i++)
  {
  for($j=0;$j<sizeof($arr);$j++)
    {
      echo $arr[$i][$j]." ";
    }
  echo "<br>";
  }
  echo "<br>";

  $arrs=array(array(1,2,3,4,5,6,7),
  array(11,12,13,14,15,16,17));
```

```
    for($i=0;$i<=7;$i++)
    {
    for($j=0;$j<=7;$j++)
       {
          $arrs[$i][$j]=10+$i*$j;
       }
    }

    for($i=0;$i<=7;$i++)
    {
    for($j=0;$j<=7;$j++)
       {
          echo $arrs[$i][$j]." ";
       }
    echo "<br>";
    }
    echo "<br>";

?>
</body>
</html>
```

Khi triệu gọi trang này trên trình duyệt, kết quả trình bày như hình 7-8.



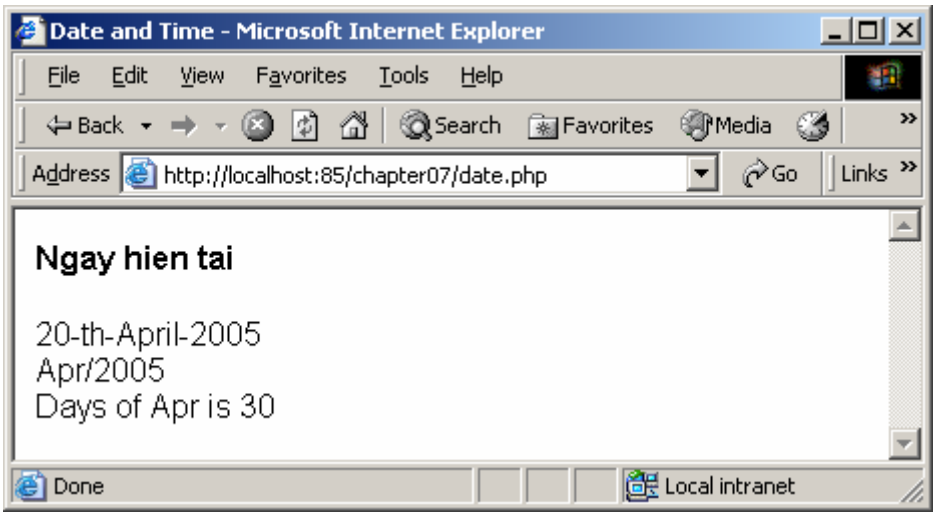
Hình 7-8: Mảng hai chiều

3. KIỂU DATETIME

Để làm việc với kiểu dữ liệu Date và Time, bạn sử dụng hàm của PHP có sẵn. Chẳng hạn, muốn trình bày chuỗi ngày tháng, bạn dùng hàm date với các tham số như ví dụ sau:

```
<html>
<head>
 <title>Date and Time</title>
</head>
<body>
<h4>Ngay hien tai</h4>
<?php
     echo date("j-S-F-Y");
     echo "<br>";
     echo date("M/Y");
     echo "<br>";
     echo "Days of ".date("M")." is ".date("t");
     echo "<br>";
?>

</body>
</html>
```

Kết quả trả về như hình 7-9.



Hình 7-9: Sử dụng hàm Date

Lưu ý rằng, than số trong hàm date được trình bày trong bảng sau

Code	Diễn giải
a	Buổi sáng/Chiều bằng hai ký tự thường *am/pm*.
A	Buổi sáng/Chiều bằng hai ký tự hoa *AM/PM*.
B	Định dạng thời gian *Swatch Internet*, bạn có thể tham khảo *http://swatch.com/internettime/internettime.php3*.
d	*Day* (01-31) trong tháng với hai số, nếu ngày 1-9 sẽ có kèm số 0.
D	*Day* (*Mon-Sun*) trong tuần với 3 ký tự.

F	Tháng (*January-December*) trong năm với tên tháng đầy đủ dạng text.
g	*Hour* (1-12) trong ngày 1 hoặc 2 số (không kèm 0 nếu giờ từ 1-9).
G	*Hour* (0-23) trong ngày 1 hoặc 2 số (không kèm 0 nếu giờ từ 0-9).
h	*Hour* (01-12) trong ngày 2 số (kèm 0 nếu giờ từ 01-09).
H	*Hour* (00-23) trong ngày 2 số (kèm 00 nếu giờ từ 00-09).
i	*Minutes* (01-59) đã trôi qua (kèm 00 nếu phút từ 00-59).
j	*Day* (1-31) 1 hoặc 2 số (không kèm 0 nếu ngày từ 1-9).
l	*Day* (*Monday-Sunday*) trong tuần dạng *text*.
L	Năm nhuần trả về 1, ngược lại hàm trả về 0.
m	*Month* (01-12) trong năm 2 số (kèm 00 nếu tháng từ 01-09).
M	*Month* (*Jan-Dec*) trong năm 3 ký tự.
n	*Month* (1-12) 1 hoặc 2 số (không kèm 0 nếu tháng từ 1-9).
s	*Seconds* (01-59) đã trôi qua (kèm 00 nếu giây từ 00-59).
S	Thêm hai ký tự *st*, *nd*, *rd* hay *th* theo sau ngày dạng hai ký tự số (ví dụ như 12^{th}).
t	Trả về tổng số ngày trong tháng (từ 28 -31).
T	Ký tự *Timezone* của server với 3 ký tự, chẳng hạn như *EST*.
U	Tổng số *Seconds* từ 1 January 1970 tới hôm nay ứng với *UNIX Time Stamp*.
w	Day (0-6) của tuần, 0 ứng với *Sunday* và 6 ứng với *Saturday*.
y	Năm định dạng 2 con số (03).
Y	Năm định dạng 4 con số (2003).
z	Ngày trong năm một hoặc 2 con số (0-365).
X	*Timezone* hiện tại tính bằng giây từ -43200 đến 43200.

---

4. <u>KẾT LUẬT</u>

Trong bài này, chúng ta tập trung tìm hiểu xử lý chuỗi, mảng và hàm ngày tháng. Trong bài tiếp, chúng ta tiếp tục tìm hiểu cơ sở dữ liệu mySQL.

Môn học: MySQL

Bài 8

Bài học này chúng ta sẽ làm quen cách thao tác trên cơ sở dữ liệu MySQL:

- ✓ *Giới thiệu cơ sở dữ liệu MySQL*
- ✓ *Cài đặt MySQL*
- ✓ *Cấu hình*
- ✓ *Kiểu dữ liệu*
- ✓ *Khai báo các phát biểu*

1. GIỚI THIỆU CƠ SỞ DỮ LIÊU MYSQL

MySQL là cơ sở dữ liệu được sử dụng cho các ứng dụng *Web* có quy mô vừa và nhỏ. Tuy không phải là một cơ sở dữ liệu lớn nhưng chúng cũng có trình giao diện trên *Windows* hay *Linux*, cho phép người dùng có thể thao tác các hành động liên quan đến cơ sở dữ liệu.

Cũng giống như các cơ sở dữ liệu, khi làm việc với cơ sở dữ liệu *MySQL*, bạn đăng ký kết nối, tạo cơ sở dữ liệu, quản lý người dùng, phần quyền sử dụng, thiết kế đối tượng *Table* của cơ sở dữ liệu và xử lý dữ liệu.

Tuy nhiên, trong bất kỳ ứng dụng cơ sở dữ liệu nào cũng vậy, nếu bản thân chúng có hỗ trợ một trình giao diện đồ hoạ, bạn có thể sử dụng chúng tiện lợi hơn các sử dụng *Command line*. Bởi vì, cho dù bạn điều khiển *MySQL* dưới bất kỳ hình thức nào, mục đích cũng quản lý và thao tác cơ sở dữ liệu.

2. CÀI ĐẶT MYSQL

Để cài đặt MySQL trên nền Windows bạn theo các bước sau:

- Trước tiên bạn chép tập tin mysql-4.0.0a-alpha-win.zip vào đĩa cứng hoặc chọn chúng từ đĩaq CD và giải nén tập tin
- Chạy tập tin Setup.exe, chọn đĩa C hay D
- Sau khi cài đặt thành công, bạn kiểm tra trong Windows Services xuất hiện dịch vụ mySQL hay không?. Để sử dụng được MySQL thì trạng thái của dịch vụ này phải ở chế độ Started.

Lưu ý rằng, trong trường hợp MySQL không thể chạy được, do dịch vụ của MySQL chưa Started như , để có thể chạy được MySQL thì bạn cần một số thay đổi trong tập tin my.ini trong thư mục WINNT

```
----------------------------------------------------------------------------
#This File was made using the WinMySQLAdmin 1.3
#Tool
#9/11/2003 10:50:13 AM
#Uncomment or Add only the keys that you know how works.
#Read the MySQL Manual for instructions
[mysqld-nt]
basedir=C:/mysql
#bind-address=127.0.0.1
datadir=C:/mysql/data
#language=C:/mysql/share/your language directory
#slow query log#=
#tmpdir#=
#port=3306
#set-variable=key_buffer=16M
```

```
[WinMySQLadmin]
Server=C:/mysql/bin/mysqld-nt.exe
user=root
password=
QueryInterval=10
```

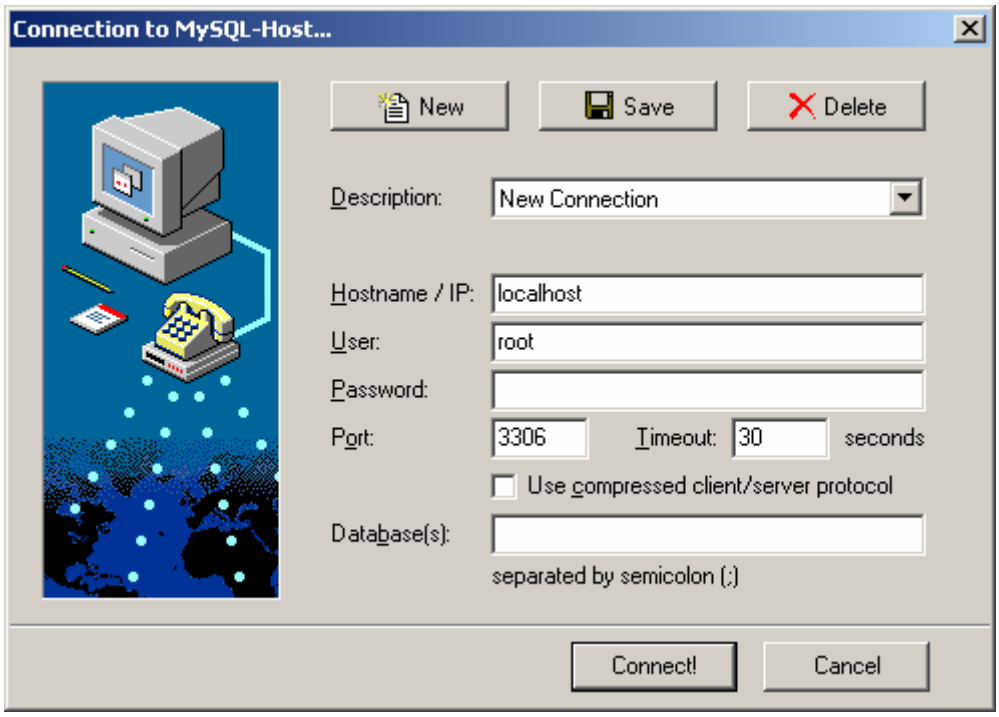
3. TAO CƠ SỞ DỮ LIỆU VÀ NGƯỜI DÙNG

Trong trường hợp bạn sử dụng giao diện đồ hoạ thì dùng ích quản trị cơ sở dữ liệu MySQL, bạn có thể chạy tập tin *mysqlfront.exe* trong thư mục *MySQL Control*, bằng cách chạy tập tin cửa sổ xuất hiện như hình 8-1. Nếu lần đầu tiên tạo kết nối cơ sở dữ liệu, bạn cần phải tạo một *Connection*, cung cấp tên *Server* hay *IP* của máy chứa *MySQL*.

Tuy nhiên, trong trường hợp máy chứa cơ sở dữ liệu *MySQL* là máy đang sử dụng, bạn có thể sử dụng *localhost*. Ngài ra, cũng giống như các cơ sở dữ liệu khác, *Username* mặc định của cơ sở dữ liệu *MySQl* là *root* và *Password* là rỗng.

Nếu bạn đã có cơ sở dữ liệu đang tồn tại, bạn có thể gõ tên cơ sở dữ liệu trong phần *Databases* (nếu muốn mở nhiều *database*, bạn có thể dùng dấu ; để phân cách).

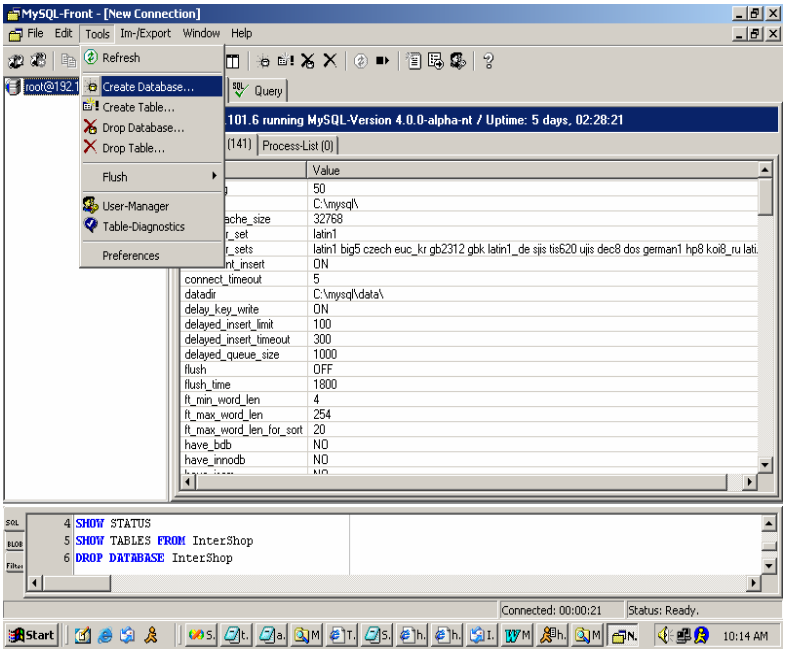
Trong trường hợp lần đầu tiên, bạn không cần cung cấp tên cơ sở dữ liệu, bạn có thể tạo chúng sau khi kết nối.



Hình 8-1: Kết nối cơ sở dữ liệu bằng MySQLFront Tool

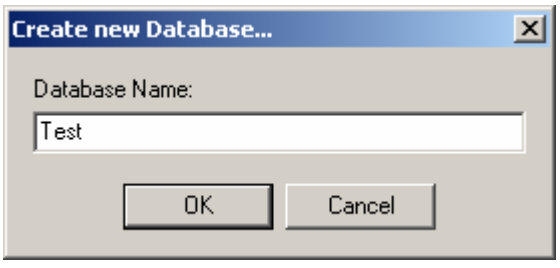
Sau kết nối cơ sở dữ liệu thành công, trình điều khiển cơ sở dữ liệu *MySQL* có giao diện như hình 8-2, cộng việc đầu tiên bạn phải thực hiện là tạo cơ sở dữ liệu.

Bắt đầu từ *menu* có tên *Tools* | *Create Database* hay chọn tên *root@localhost* | *R-Click* | *Create Database*, cửa sổ xuất hiện như hình 8-3.



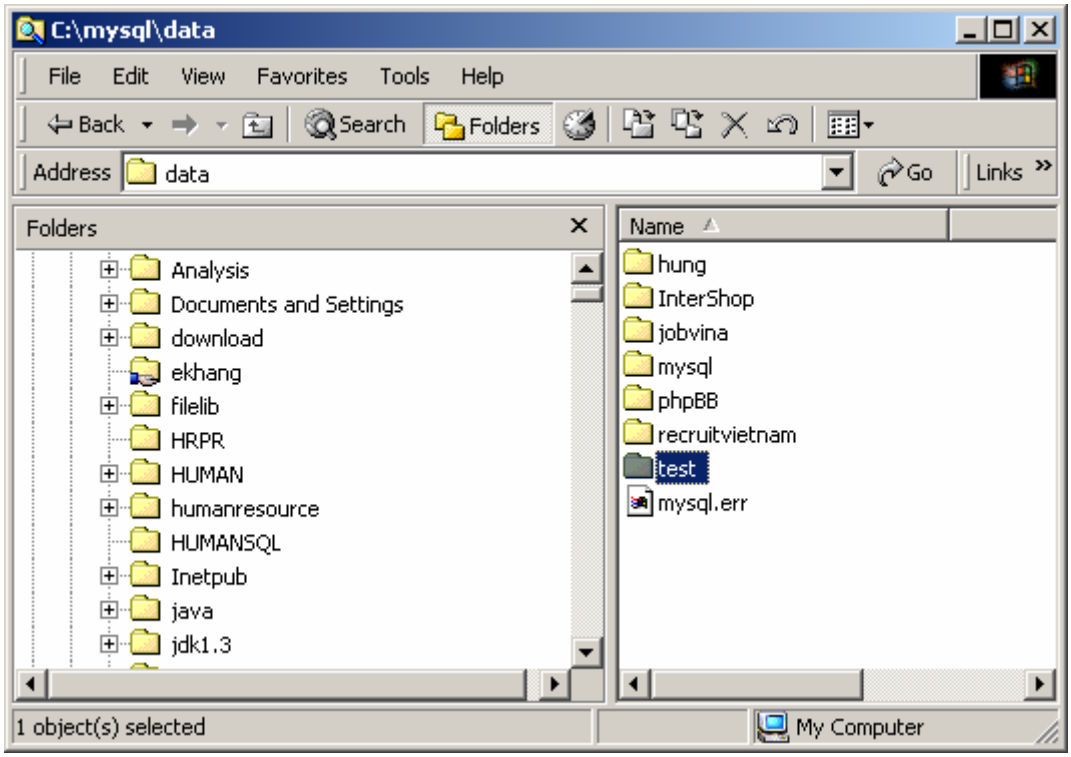
Hình 8-2: Giao diện điều khiển cơ sở dữ liệu MySQL

Cung cấp tên cơ sở dữ liệu, trong trường hợp này bạn có thể nhập *Test*, bấm nút *OK*, cơ sở dữ liệu xuất hiện trong cửa sổ điều khiển.



Hình 8-3: Tạo cơ sở dữ liệu có tên Test

Trong cả hai trường hợp tạo cơ sở dữ liệu bằng *MySQL* thành công như trên, bạn có thể tìm thấy tên cơ sở dữ liệu đó trong thư mục *mysql/data* như hình 8-4 sau:



Hình 8-4: Thư mục tin cơ sở dữ liệu Test

3.1. Quản lý người dùng

Làm thế nào để đăng nhập vào cơ sở dữ liệu *MySQL*, bạn có thể sử dụng hai cách như trình bày ở trên. Tuy nhiên, sau khi tạo ra các *username* khác, bạn có thể sử dụng chúng để đăng nhập.

Để đăng nhập vào *MySQL* bằng *Command line*, bạn chỉ cần gõ *>mysql – hostname –u username – p* từ dấu nhắc hay đăng nhập bằng cách sử dụng trình giao diện đồ hoạ. Từ khoá *-h* hỉ ra rằng tên (*computer name*), *IP*, hay *localhost* của máy có sử dụng cơ sở dữ liệu *MySQL*, *-u* chỉ ra rằng bạn sử dụng *username*, *username* là tên *username*, *-p* được chỉ định khi *username* này có *password*. Trong trường hợp *password* là rỗng, bạn có thể không cung cấp tham số *–p*.

Để tạo *User* trong cơ sở dữ liệu *MySQL*, bạn có thể sử dụng hai cách trên. Nếu bạn thực hiện việc tạo một *Username* bằng *Command line*, bạn có thể gõ từ dấu nhắc như phát biểu sau:

```
GRANT
            Select, Insert, Update,
            Delete, Index, Alter,
            Create, Drop, References
            ON *.* TO 'myis'@'%'
            IDENTIFIED BY '12345678'
```

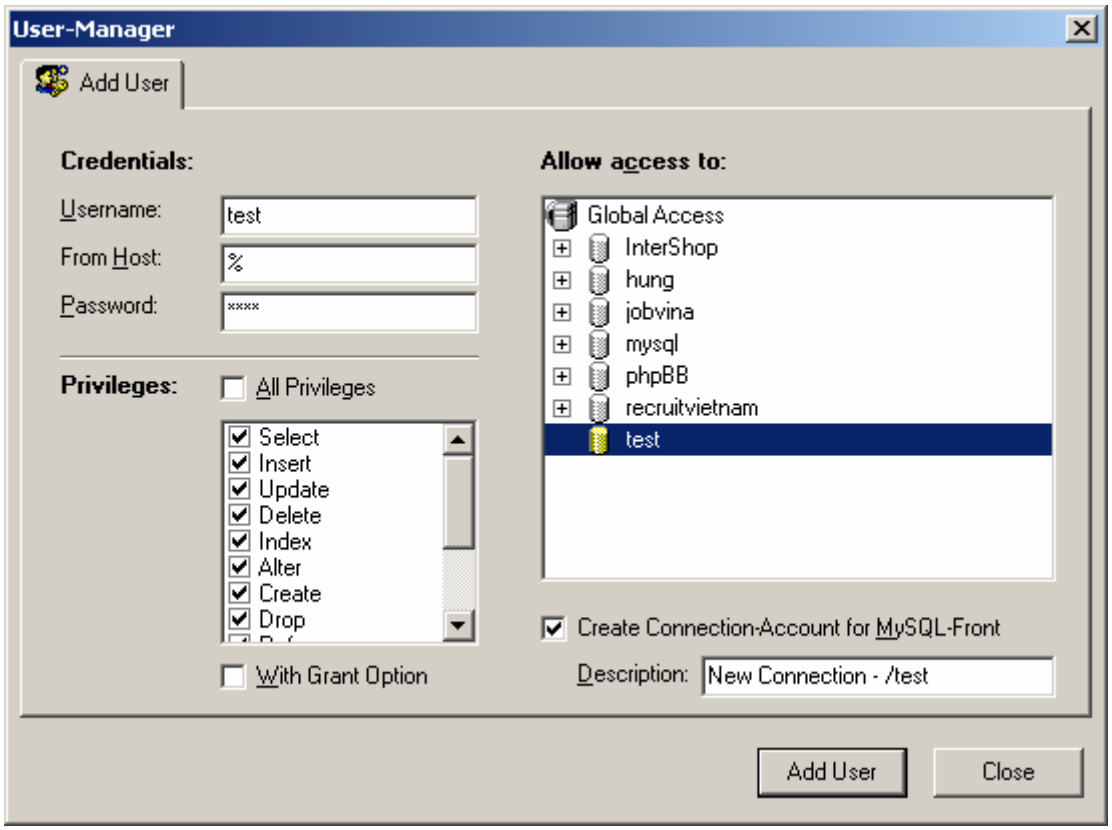
Trong phát biểu trên, vừa tạo ra *User* có tên *myis*, với *hostname* là cơ sở dữ liệu hiện hành, *password* là *1234* và được các đặt quyền *Select, Insert, Update, Delete, Index, Alter, Create*, *Drop* trên cơ sở dữ liệu hiện hành.

Trong trường hợp bạn tạo ra một *Username* không cung cấp các đặt quyền trên cơ sở dữ liệu, bạn có thể thực hiện như phát biểu tạo *username: test, password*: *1234* sau:

```
GRANT
            usage
            ON *.* TO 'test'@'%'
```

```
IDENTIFIED BY '1234'
```

Nếu bạn sử dụng giao diện đồ hoạ, bạn có thể tạo *username* và gán quyền như trên bằng cách sử dụng *menu* có tên *Tools | User-Manager*, cửa sổ xuất hiện như hình 8-5.



Hình 8-5: Tạo Username

3.2. Cấp quyền cho người dùng

Các đặt quyền *Select, Insert, Update, Delete, Index, Alter, Create*, *Drop* trên cơ sở dữ liệu, bạn có thể tham khảo chi tiết trong bảng 8-1.

Bảng 8-1: Các đặt quyền trên cơ sở dữ liệu

Loại	áp dụng	Diễn giải
select	tables, columns	Cho phép *user* truy vấn mẩu tin từ *Table*.
insert	tables, columns	Cho phép *user* thêm mới mẩu tin vào *Table*.
update	tables, columns	Cho phép *user* thay đổi giá trị của mẩu tin tồn tại trong *Table*.
delete	tables	Cho phép user mẩu tin tồn tại trong *Table*.
index	tables	Cho phép *user* thêm mới hay xoá chỉ mục của *Table*.
alter	tables	Cho phép *user* thay đổi cấu trúc của đối tượng *Table*

		hay Database tồn tại, như thêm cột vào trong *Table* tồn tại, thay đổi kiểu dữ liệu của cột dữ liệu, ..
create	databases tables	Cho phép *user* tạo mới đối tượng *Table* hay Database.
drop	databases tables	Cho phép user xoá đối tượng *Table* hay *Database*.

Xuất phát từ các quyền có ảnh hưởng đến cấu trúc cơ sở dữ liệu, các đối tượng của cơ sở dữ liệu và dữ liệu, bạn có thể xem xét kỹ càng trước khi cấp quyền cho *user* àm việc trên cơ sở dữ liệu.

Ngoài các quyền trên, trong *MySQL* còn có một số quyền không gán mặc định như trong bảng 8-2, bạn có thể xem xét các đặt quyền quản trị để cấp cho người dùng.

Bảng 8-2: Các đặt quyền quản trị trên cơ sở dữ liệu

Loại	Diễn giải
reload	Cho phép người quản trị nạp lại các *Table*, quyền, *host, logs* và *Table*.
shutdown	Cho phép người quản trị chấm dứt hoạt động *MySQL Server*.
process	Cho phép người quản trị xem quá trình thực hiện của trình chủ và có thể chấm dứt một số quá trình đang thực thi.
file	Cho phép dữ liệu ghi vào *Table* từ tập tin.

Lưu ý: Những *username* bình thường không nên cấp quyền như trong bảng 8-2 cho họ, trong trường hợp bạn muốn cấp tất cả các quyền trong bảng 8-1 và Bảng 8-2 cho *username* khi tạo ra họ, bạn *Table* sử dụng từ khoá *All* thay vì *All Privileges* trong phát biểu tạo *user* như sau:

```
GRANT
            ALL
            ON *.* TO 'ekhang'@'%'
            IDENTIFIED BY '12345678'
```

Tương tự như vậy, trong trường hợp bạn không cung cấp bất kỳ đặt quyền nào trên cơ sở dữ liệu hiện hành, bạn có thể khai báo phát biểu cấp uyền như sau:

```
GRANT
            usage
            ON *.* TO 'ekhang'@'%'
            IDENTIFIED BY '12345678'
```

3.3. Xoá quyền của user

Để xoá các quyền của *user* từ cơ sở dữ liệu hiện hành, bạn có thể sử dụng phát biểu *SQL* có tên *Revoke*, phát biểu *Revoke* ngược lại với phát biểu *Grant*.

Nếu bạn xoá một số quyền của *user*, bạn có thể sử dụng khai báo như phát biểu sau:

```
Revoke privileges [(columns)]
ON item
From username
```

Trong trường hợp xoá tất cả các quyền của *user*, bạn có thể sử dụng phát biểu như sau:

```
Revoke All
ON item
From username
```

Nếu *user* đó được cấp quyền với tuỳ chọn *Grant Option*, để xoá các quyền đó của *user*, bạn có thể khai báo như sau:

```
Revoke Grant Option
ON item
From username
```

Để tham khảo chi tiết quá trình cấp và xoá quyền của một *user*, bạn có thể tham khảo một số phát biểu như sau:

Gán quyền *Administrator* cho *user* có tên *fred* trên mọi cơ sở dữ liệu trong *MySQL*, *password* của anh ta là *mnb123*, bạn có thể khai báo như sau:

```
Grant all
On *
To fred indetifyed by 'mnb123'
With Grant Option;
```

Nếu bạn không muốn *user* có tên *fred* trong hệ thống, bạn có thể xoá anh ta bằng cách khai báo phát biểu sau:

```
Revoke all
On *
From fred;
```

Tạo một *user* có tên *ekhang* với *password* là *12345678*, được làm việc trên cơ sở dữ liệu *Test*, không cấp quyền cho *user* này, bạn có thể khai báo như sau:

```
Grant usage
On Test.*
To ekhang identified by '12345678';
```

Tương tự như vậy, trong trường hợp bạn muốn cấp một số quyền cho *user* có tên *ekhang* trên cơ sở dữ liệu *Test*, bạn có thể khai báo như sau:

```
Grant select, insert, delete, update, index, drop
On Test.*
To ekhang;
```

Nếu bạn muốn xoá bớt một số quyền của *user* có tên *ekhang* trên cơ sở dữ liệu *Test*, bạn có thể khai báo như sau:

```
Revoke update, delete, drop
On Test.*
From ekhang;
```

Nhưng trong trường hợp xoá tất cả các quyền của *user* có tên *ekhang* trên cơ sở dữ liệu *Test*, bạn có thể khai báo:

```
Revoke All
On Test.*
From ekhang;
```

4. KIỂU DỮ LIÊU CỦA CƠ SỞ DỮ LIÊU MYSQL

Trước khi thiết kế cơ sở dữ liệu trên *MySQL*, bạn cần phải tham khảo một số kiểu dữ liệu thường dùng, chúng bao gồm các nhóm như: *numeric, date and time* và *string*.

Đều cần lưu ý trong khi thiết kế cơ sở dữ liệu, bạn cần phải xem xét kiểu dữ liệu cho môt cột trong *Table* sao cho phù hợp với dữ liệu của thế giới thực.

Điều này có nghĩa là khi chọn dữ liệu cho cột trong *Table*, bạn phải xem xét đến loại dữ liệu cần lưu trữ thuộc nhóm kiểu dữ liệu nào, chiều dài cũng như các ràng buộc khác, nhằm khai báo cho phù hợp.

4.1. Loại dữ liệu numeric

Kiểu dữ liệu *numeric* bao gồm kiểu số nguyên trình bày trong bảng 8-3 và kiểu số chấm động, trong trường hợp dữ liệu kiểu dấu chấm động bạn cần phải chỉ rõ bao nhiều số sau đấu phần lẻ như trong bảng 8-4.

Bảng 8-3: Kiểu dữ liệu số nguyên

Loại	Range	Bytes	Diễn giải
tinyint	-127->128 hay 0..255	1	Số nguyên rất nhỏ.
smallint	-32768 ->32767 hay 0..65535	2	Số nguyên nhỏ.
mediumint	-8388608 -> 838860 hay 0..16777215	3	Số nguyên vừa.
int	-2^{31}->2^{31}-1 hay 0..2^{32}-1	4	Số nguyên.
bigint	-2^{63}->2^{63}-1 hay 0..2^{64}-1	8	Số nguyên lớn.

Bảng 8-4: Kiểu dữ liệu số chấm động

Loại	Range	Bytes	Diễn giải
float	phụ thuộc Số thập Phân		Số thập phân dạng *Single* hay *Double*.
Float(M,D)	±1.175494351E-38 ±3.40282346638	4	Số thập phân dạng *Single*.
Double(M,D)	±1.7976931348623157308 ±2.2250738585072014E-308	8	Số thập phân dạng *Double*.

```
Float(M[,D])
                                Số chấm động lưu
                                dưới dạng char.
-----------------------------------------------
```

4.2. Loại dữ liệu Datet and Time

Kiểu dữ liệu *Date and Time* cho phép bạn nhập liệu dưới dạng chuỗi hay dạng số như trong bảng 8-5.

Bảng 8-5: Kiểu dữ liệu số nguyên

```
-----------------------------------------------
Loại      Range       Diễn giải
-----------------------------------------------
Date      1000-01-01  Date trình bày dưới dạng
                      yyyy-mm-dd.

Time      -838:59:59  Time trình bày dưới
          838:59:59   dạng hh:mm:ss.

DateTime  1000-01-01  Date và Time trình bày dưới
          00:00:00    dạng yyyy-mm-dd hh:mm:ss.
          9999-12-31
          23:59:59

TimeStamp[(M)]
          1970-01-01  TimeStamp trình bày dưới
          00:00:00    dạng yyyy-mm-dd hh:mm:ss.
Year[(2|4)]
          1970-2069   Year trình bày dưới
          1901-2155   dạng 2 số hay 4 số.
-----------------------------------------------
```

Đối với kiểu dữ liệu *TimeStamp*, bạn có thể định dạng nhiều cách như trình bày trong bảng 8-6.

Bảng 8-6: Trình bày đại diện của TimeStamp

```
-----------------------------------------------
Loại               Hiển thị
-----------------------------------------------
TimeStamp          YYYYMMDDHHMMSS
TimeStamp(14)      YYYYMMDDHHMMSS
TimeStamp(12)      YYMMDDHHMMSS
TimeStamp(10)      YYMMDDHHMM
TimeStamp(8)       YYYYMMDD
TimeStamp(6)       YYMMDD
TimeStamp(4)       YYMM
TimeStamp(2)       YY
-----------------------------------------------
```

4.3. Loại dữ liệu String

Kiểu dữ liệu *String* chia làm ba loại, loại thứ nhất như char (chiều dài cố định) và *varchar* (chiều dài biến thiên). *Char* cho phép bạn nhập liệu dưới dạng chuỗi với chiếu dài lớn nhất bằng chiều dài bạn đã định nghĩa, nhưng khi truy cập dữ liệu trên *Field* có khai báo dạng này, bạn cần phải xử lý khoảng trắng. Điều này có nghĩa là nếu khai báo chiều dài là 10, nhưng bạn chỉ nhập hcuỗi 4 ký tự, *MySQL* lưu trữ trong bộ nhớ chiều dài 10.

Ngược lại với kiểu dữ liệu *Char* là *Varchar*, chiều dài lớn hất người dùng có thể nhập vào bằng chiều dài bạn đã định nghĩa cho *Field* này, bộ nhớ chỉ lưu trữ chiều dài đúng với chiều dài của chuỗi bạn đã nhập.

Như vậy, có nghĩa là nếu bạn khai báo kiểu *varchar* 10 ký tự, nhưng bạn hcỉ nhập 5 ký tự, *MySQL* chỉ lưu trữ chiều dài 5 ký tự, ngoài ra, khi bạn truy cập đến *Field* có kiểu dữ liệu này, bạn không cần phải giải quyết khoảng trắng.

Loại thứ hai là *Text* hay *Blob*, *Text* cho phép lưu chuỗi rất lớn, *Blob* cho phép lưu đối tượng nhị phân. Loại thứ 3 là *Enum* và *Set*. Bạn có thể tham khảo cả ba loại trên trong bảng 8-7.

Bảng 8-7: Kiểu dữ liệu String

Loại	Range	Diễn giải
char	1-255 characters	Chiều dài của chuỗi lớn nhất 255 ký tự.
varchar	1-255 characters	Chiều dài của chuỗi lớn nhất 255 ký tự (*characters*).
tinyblob	2^8-1	Khai báo cho *Field* chứa kiểu đối tượng nhị phân cở 255 *characters*.
tinytext	2^8-1	Khai báo cho *Field* chứa kiểu chuỗi cở 255 *characters*.
blob	2^{16}-1	Khai báo cho *Field* chứa kiểu *blob* cở 65,535 *characters*..
text	2^{16}-1	Khai báo cho *Field* chứa kiểu chuỗi dạng văn bản cở 65,535 *characters*.
Mediumblob	2^{24}-1	Khai báo cho *Field* chứa kiểu blob vừa khoảng 16,777,215 *characters*.
Mediumtext	2^{24}-1	Khai báo cho *Field* chứa kiểu chuỗi dạng văn bản vừa khoảng 16,777,215 *characters*.
Longblob	2^{32}-1	Khai báo cho *Field* chứa kiểu *blob* lớn khoảng 4,294,967,295 *characters*.
Longtext	2^{32}-1	Khai báo cho *Field* chứa kiểu chuỗi dạng văn bản lớn khoảng 4,294,967,295 *characters*.

5. PHÁT BIỂU SQL

MySQL là một hệ thống quản lý cơ sở dữ liệu quan hệ (*RDBMS*) hay còn được gọi là *Relational Database Management System*. *RDBMS* là một trong những mô hình cơ sở dữ liệu quan hệ thông dụng hiện nay.

5.1. Nhóm phát biểu SQL

Như đã trình bày trong chương 3, hầu hết sản phẩm cơ sở dữ liệu quan hệ hiện nay đều dựa trên chuẩn của *SQL* và *ANSI-SQL*, chẳng hạn như *SQL Server*, *Oracle*, *PostgreSQL* và *MySQL*. Điều này có nghĩa là tất cả những cơ sở dữ liệu quan hệ đều phải có những tiêu chuẩn theo cú pháp *SQL* và *MySQL* cũng không phải là ngoại lệ.

Ngôn ngữ *SQL* chia làm 4 loại sau:

- *DDL (Data Definition Language)*: Ngôn ngữ định nghĩa dữ liệu, dùng để tạo cơ sở dữ liệu, định nghĩa các đối tượng cơ sở dữ liệu như *Table*, *Query*, *Views* hay các đối tượng khác.
- *DML (Data Manipulation Language)*: Ngôn ngữ thao tác dữ liệu, dùng để thao tác dữ liệu, chẳng hạn như các phát biểu: *Select, Inert, Delete, Update, ...*
- *DCL*: *(Data Control Language)*: Ngôn ngữ sử dụng truy cập đối tượng cơ sở dữ liệu, dùng để thay đổi cấu trúc, tạo người dùng, gán quyền chẳng hạn như: *Alter, Grant, Revoke, ...*
- *TCL*: *(Transaction Control Language)*: Ngôn sử dụng để khai báo chuyển tác chẳng hạn như: *Begin Tran, Rollback, Commit, ...*

5.2. Phát biểu SQL thao tác dữ liệu

Phát biểu *SQL* bao gồm các loại như sau:

- *SELECT (Truy vấn mẩu tin).*
- *INSERT (Thêm mẩu tin).*
- *UPDATE (Cập nhật dữ liệu).*
- *DELETE (Xoá* mẩu tin).

5.2.1. Khái niệm cơ bản về Select

Phát biểu *Select* dùng để truy vấn dữ liệu từ một hay nhiều bảng khác nhau, kết quả trả về là một tập mẩu tin thoã các điều kiện cho trước nếu có, cú pháp của phát biểu *SQL* dạng *SELECT*:

```
SELECT <danh sách các cột>
[FROM <danh sách bảng>]
[WHERE <các điều kiện ràng buộc>]
[GROUP BY <tên cột / biểu thức trong SELECT> ]
[HAVING <điều kiện bắt buộc của GROUP BY>]
[ORDER BY <danh sách cột>]
[LIMIT FromNumber | ToNumber]
```

Danh sách các cột: Khai báo các tên cột, biểu thức kết hợp giữa các cột của *Table* bạn cần truy lục. Trong trường hợp có hai cột cùng tên của hai *Table* trong phát biểu, bạn cần phải chỉ định tên *Table* đi trước. Chẳng hạn, như ví dụ 8-1.

Ví dụ 8-1: Phát biểu SELECT

```
Select ItemID,ItemName
From tblItems
Where Cost>100;

Select tblOrders.OrderID,OrderDate,ItemID,Qtty
From tblOrders,tblOrderDetails
Where tblOrders.OrderID = _ tblOrderDetail.OrderID;
```

5.2.2. Phát biểu SELECT với mệnh đề FROM

Phát biểu *SQL* dạng *SELECT* là một trong những phát biểu yêu cầu *MySQL* truy lục dữ liệu trên cơ sở dữ liệu chỉ định. *SELECT* dùng để đọc thông tin từ cơ sở dữ liệu theo những trường quy định, hay những biểu thức cho trường đó.

Mệnh đề *FROM* chỉ ra tên một bảng hay những bảng có quan hệ cần truy vấn thông tin. Thường chúng ta sử dụng công cụ *MySQL-Front | Query* để thực thi phát biểu *SQL*.

Sau khi thực thi phát biểu *SQL*, kết quả trả về số mẩu tin và tổng số mẩu tin được lấy ra từ bảng.

Dấu * cho phép lọc mẩu tin với tất cả các trường trong bảng, nếu muốn chỉ rõ những trường nào cần lọc bạn cần nêu tên cụ thể những trường đó.

Để tiện tham khảo trong giáo trình này chúng tôi sử dụng một phần cơ sở dữ liệu có sẵn của *MySQL*, đồng thời bổ sung thêm cơ sở dữ liệu dành cho ứng dụng bán hàng qua mạng.

Cơ sở dữ liệu bán hàng qua mạng có tên là *Test*, và bao gồm nhiều bảng. Bằng phát biểu *SELECT* chúng ta có thể biết số bảng hay đối tượng khác đang có trong cơ sở dữ liệu *Test*

Ví dụ 8-2: Thực thi phát biểu SQL SELECT hệ thống

```
show tables
from Test
/* Hiển thị tất cả tên bảng của cơ sở dữ liệu hiện hành */
```

Kết quả trả về danh sách bảng như sau:

```
TABLES_IN_TEST
---------------------------------------
tblCountries
tblProvinces
tblAuthors

tblPayment
tblItemsion
tblCustomers
tblSoftware
```

Ghi chú:

Bạn có thể sử dụng phát biểu SQL trên để hiển thị những đối tượng trong cơ sở dữ liệu, bằng cách thay thế các tham số và điều kiện.

Cú pháp đơn giản

```
Select *
From tablename
/* Lọc tất cả số liệu của tất cả các cột (field) của tablename*/

Select field1,field2
From tablename
/* Lọc tất cả số liệu của 2 field: field1, field2 của tablename*/

Select *
From tablename
Limit 0,10
/* Lọc top 10 mẩu tin  đầu tiên của tất cả các field của tablename*/

Select field1, field2
From tablename
Limit 0,10
/* Lọc top 10 mẩu tin  đầu tiên của 2  fields field1, field2 của
```

tablename/*

Ví dụ 8-3: phát biểu phát biểu SQL dạng Select

```
Select *
From tblCountries
/* Liệt kê tất cả các quốc gia trong bảng tblCountries hoặc bạn có thể liệt kê tên như phát biểu sau */

Select CountryName
From tblCountries
```

Kết quả trả về như sau:

```
CountryCode   CountryName
--------- ---------- ------------ -------------
VNA           Vietnam
SNG           Singapore
USS           United Stated
UKD           United Kingdom
GER           Germany
CAM           Cambodia
THA           Thai Land
MAL           Malaysia
INC           Indonesia
CHN           China
```

5.2.3. Phát biểu SQL dạng SELECT với mệnh đề Where

Khi bạn dùng mệnh đề *WHERE* để tạo nên tiêu chuẩn cần lọc mẩu tin theo tiêu chuẩn được định nghĩa, thông thường *WHERE* dùng cột (trường) để so sánh với giá trị, cột khác, hay biểu thức chứa cột (trường) bất kỳ có trong bảng. Phát biểu *SQL* dạng *Select* với mệnh đề *Where* cú pháp có dạng như sau:

```
Select *
from tablename
where conditions

Select field1, field2, field3
from tablename
where conditions
```

Với *conditions* trong cả hai phát biểu trên được định nghĩa điều kiện truy vấn như khai báo sau:

```
Select *
From tablename
where field1>10

select *
from tblCountries
where CountryCode in('VNA','CHN')
```

Các phép toán so sánh trong *conditions* bao gồm:

- ♦ > : lớn hơn where Amount > 100000;
- ♦ < : nhỏ hơn where Amount < 100000;
- ♦ >= : lớn hơn hoặc bằng where Amount >= 100000;
- ♦ >= : nhỏ hơn hoặc bằng where Amount <= 100000;
- ♦ = : bằng where CustID='12';

- != : Khác where CustID!='12';
- <> : Khác where CustID<>'12';

Các phép toán *logic* có thể sử dụng trong *conditions*

- *and* : Phép toán "*and*"

```
SELECT *
FROM tblOrders
Where Amount!>100000
And CustID='12';
```

- *Or* : Phép toán "*or*"

```
SELECT *
FROM tblOrderDetails
Where Amount!>100000
Or CustID='12';
```

- *Not* : Phép toán phủ định (*not*)

```
SELECT *
FROM tblOrders
where OrderDate is not null;
```

- *Not in* : Phép toán phủ định (*not in*)

```
SELECT *
FROM tblOrders
where OrderID not in ('12','15');
```

- *Between*: Kết quả thuộc trong miền giá trị

```
SELECT *
FROM tblOrders
Where Amount  between 10
And 500;
```

- *Like* : Phép toán so sánh gần giống, sử dụng dấu % để thể hiện thay thế bằng ký tự đại diện

```
SELECT *
FROM tblCustomers
where CustName like '%A';
```

- *Not Like* : Phép toán phủ định so sánh gần giống, sử dụng dấu % để thể hiện thay thế bằng ký tự đại diện

```
SELECT *
FROM tblCustomers
where CustName not like '%A';
```

- *IN* : Phép toán so sánh trong một tập hợp

```
SELECT *
FROM tblOrders
Where OrderID in ('100','200','300');
```

Ví dụ 8-5: Ví dụ về SQL dạng SELECT và Where

```
/* > : lớn hơn */

     Select *
     From tblOrders
     Where Amount > 100000;

/* < : nhỏ hơn */

     Select *
     From tblOrders
     Where Amount < 100000;

/* >=: lớn hơn hoặc bằng */

     Select *
     From tblOrders
     Where Amount >= 100000;

/* >=: nhỏ hơn hoặc bằng */

     Select *
     From tblOrders
     Where Amount <= 100000;

/* = : bằng */

     Select *
     From tblOrders
     Where CustID='12';

/* != :Khác */

     Select *
     From tblOrders
     Where CustID !='12';

/* <>: Khác */

     Select *
     From tblOrders
     Where CustID <>'12';

/* !> : Không lớn hơn */

     Select *
     From tblOrders
     Where Amount !> 100000;

/* !< : Không nhỏ hơn */

     Select *
     From tblOrders
     Where Amount !< 100000;

-- Các phép toán logic
/* and    : Phép toán và */

     Select *
     From tblOrders
```

```
Where Amount !>100000
And CustID='12';
```

/ Or : Phép toán hoặc */*

```
Select *
From tblOrders
Where Amount !>100000
Or CustID='12';
```

/ Not : Phép toán phủ định */*

```
Select *
From tblOrders
Where OrderDate is NOT NULL;
```

/ Between: giá trị nằm trong miền */*

```
Select *
From tblOrders
Where Amount
Between 10 and 500;
```

/ Like : Phép toán so sánh gần giống, sử dụng dấu %*
*để thể hiện thay thế bất kỳ ký tự */*

```
Select *
From tblOrders
Where Descriion like '%A'
Or CustID ='152';
```

/ Not Like : Phép toán phủ định so sánh gần giống,*
*sử dụng dấu % để thể hiện thay thế bất kỳ ký tự */*

```
Select *
From tblOrders
Where Descriion not like '%A'
Or CustID ='152';
```

/ IN: Phép toán so sánh trong một tập hợp */*

```
Select *
From tblOrders
Where OrderID in ('134','244','433');
```

/ Not IN : Phép toán phủ định so sánh trong một tập hợp */*

```
Select *
From tblOrders
Where OrderID not in ('134','244','433');
```

5.2.4. Mệnh đề Order by

Thông thường, trong khi truy vấn mẩu tin từ bảng dữ liệu, kết quả hiển thị cần sắp xếp theo chiều tăng hay giảm dựa trên ký tự *ALPHABET*. Nhưng bạn cũng có thể sắp xếp theo một tiêu chuẩn bất kỳ, chẳng hạn như biểu thức.

Khi sắp xếp dữ liệu trình bày trong kết quả, cần phải chọn trường hay biểu thức theo trật tự tăng dần hoặc giảm dần.

Cú pháp cho mệnh đề *ORDER BY* cùng với trạng thái tăng hay giảm, ứng với *ASC* sắp xếp tăng dần, *DESC* giảm dần.

Cú pháp có dạng như sau:

```
Order by columnname DESC
Order by columnname1 + columnname2 DESC
Order by columnname ASC
Order by columnname1 ASC, columnname2 DESC
```

Ví dụ 8-6: SELECT với mệnh đề Order by DESC

```
/*-- Giảm dần theo thời gian */
Select OrderID , OrderDate, CustID, Amount
From tblOrders
Where Amount >1000
Order by OrderDate DESC
```

Kết quả trả về như sau:

```
OrderID  OrderDate    CustID   Amount
---------- ---------- ---------- --------------------
17        2001-09-20    12     178.243
18        2001-09-20    12     2.78534
16        2001-09-19    12     398.798
15        2001-09-18    12     5.758.876
14        2001-09-17    12     5.539.647
12        2001-09-16    12     1.330
13        2001-09-16    12     1.585.563
31        2001-09-16    13      459.525
11        2001-09-15    11     1.401.803
28        2001-09-15    13      1.45200
```

Ví dụ 8-7: SQL dạng SELECT với mệnh đề Order by và ASC

```
/*-- Tăng dần theo thời gian */
Select OrderID , OrderDate, CustID, Amount
From tblOrders
Where Amount >1000
 Order by OrderDate ASC
```

Kết quả trả về như sau

```
OrderID OrderDate      CustID Amount
---------- ---------- ---------- --------
01        2001-09-05    10     2.903.576
02        2001-09-05    10     48.168.567
03        2001-09-05    10     5.107.032
04        2001-09-08    10     2.355.537
05        2001-09-08    16     1.817.487
06        2001-09-10    16     26.000
19        2001-09-10    12     575.667
```

```
29        2001-09-10    13      466.500
07        2001-09-11    16      186.782
23        2001-09-11    12      459.162
```

Nếu muốn sắp xếp theo nhiều cột (trường), chỉ cần sử dụng dấu phẩy (,) để phân cách các cột.

Ví dụ 8-7: SELECT với mệnh đề Order by với 2 cột dữ liệu

```
Select OrderID , OrderDate, CustID, Amount
From tblOrders
Where Amount >1000
Order by OrderID,CustID DESC
```

Kết quả trả về như sau:

```
OrderID OrderDate    CustID   Amount
---------- -------------------------- --------
31  2001-09-16       13       459.525
30  2001-09-15       13       153.120
29  2001-09-10       13       466.500
28  2001-09-15       13       145.200
27  2001-09-14       13       603.033
26  2001-09-13       13       230.000
25  2001-09-11       13       244.904
24  2001-09-12       13       1.367.228
23  2001-09-11       12       459.162
19  2001-09-10       12       575.667
```

Nếu muốn sắp xếp theo nhiều trường kết hợp, chỉ cần dùng thứ tự từng cột cách nhau bằng dấu +.

Ví dụ 8-8: SELECT với mệnh đề Order by hợp 2 cột

```
/*-- Giảm dần theo số OrderID và CustID */
Select OrderID , OrderDate, CustID, Amount
From tblOrders
Where Amount >1000
Order by OrderID + CustID DESC
```

Kết quả trả về như sau:

```
OrderID OrderDate    CustID   Amount
---------- -------------------------- --------
31  2001-09-16       13       459.525
30  2001-09-15       13       153.120
29  2001-09-10       13       466.500
28  2001-09-15       13       145.200
27  2001-09-14       13       603.033
26  2001-09-13       13       230.000
25  2001-09-11       13       244.904
24  2001-09-12       13       1.367.228
23  2001-09-11       12       459.162
19  2001-09-10       12       575.667
```

Nếu trong phát biểu *SQL* dạng *SELECT* có nhiều bảng kết hợp lại với nhau, bạn có thể dùng thêm tên bảng ứng với cột của bảng đó. Phần này sẽ được diễn giải cụ thể hơn trong phần kế tiếp (*JOIN* -Phép hợp).

5.2.5. SQL dạng SELECT với mệnh đề GROUP BY

Khi truy vấn mẩu tin trên một hay nhiều bảng dữ liệu, thông thường có những nghiệp vụ thuộc trường nào đó có cùng giá trị, ví dụ khi hiển thị hợp đồng phát sinh trong tháng, kết quả sẽ có nhiều hợp đồng của khách hàng lặp đi lặp lại như ví dụ 8-9.

Ví dụ 8-9: SQL dạng SELECT với mệnh đề Order by

```
Select CustID, Amount
from tblOrders
```

Với phát biểu trên kết quả trả về như sau:

```
CustID     Amount
---------- -------------------------
10         2.903.576
10         48.168.567
10         5.107.032
10         2.3555347
16         181.074.847
16         26.000
16         1.867.682
16       3.600.000
16       195.713.899
16       961.804.228
16       140.180.347
12       138
12       158.555.638
12       5.539.647
12       575.887.767
12       39.879.489
12       17.824.938
12       278.503.048
12       5.756.667
12       459.162
13       136.727.628
13       244.904
13       230.000
13       603.033
13       1.452.000
13       4.665.100
13       1.531.200
13       459.525
```

Trong báo cáo chúng ta lại cần phải biết mỗi khách hàng có bao nhiêu lần trả tiền, tổng số tiền của mỗi khách hàng đã trả là bao nhiêu?

Để làm điều này, chúng ta sử dụng mệnh đề *GROUP BY* trong phát biểu *SQL* dạng *SELECT* cùng với một số hàm trong *MySQL*, bạn tham khảo ví dụ 8-10 được trình bày chi tiết từ ví dụ 4-8 nhưng nhóm mẩu tin bằng mệnh đề *Group By*.

Ví dụ 8-10: SQL dạng SELECT với mệnh đề Group By

```
Select CustID, count (CustID),
Sum(Amount)
From tblOrders
Group by CustID
Order by CustID
```

Kết quả trả về như sau:

```
CustID
----------   ----    -------------------
16            7      2.956.562.368
12            9      3.843.022.604
13            8      145.913.378
10            4      72.382.804
```

5.3. Các hàm thông dụng trong MySQL

5.3.1. Các hàm trong phát biểu GROUB BY

- Hàm *AVG*: Hàm trả về giá trị bình quân của cột hay trường trong câu truy vấn, ví dụ như phát biểu sau:

```
Select AVG(Amount)
From tblOrders
```

- Hàm *MIN*: Hàm trả về giá trị nhỏ nhất của cột hay trường trong câu truy vấn, ví dụ như phát biểu sau:

```
Select Min(Amount)
From tblOrders
```

- Hàm *MAX*: Hàm trả về giá trị lớn nhất của cột hay trường trong câu truy vấn, ví dụ như các phát biểu sau:

```
Select Max(Amount)
From tblOrders
```

- Hàm *Count*: Hàm trả về số lượng mẩu tin trong câu truy vấn trên bảng, ví dụ như các phát biểu sau:

```
Select count(*)
From tblOrders

Select count(CustID)
From tblOrders

Select count(*)
From tblOrderDetails
```

- Hàm *Sum*: Hàm trả về tổng các giá trị của trường, cột trong câu truy vấn, ví dụ như các phát biểu sau:

```
Select sum(Amount)
From tblOrders
```

Chẳng hạn, bạn có thể tham khảo diễn giải toàn bộ các hàm dùng trong mệnh đề *GROUP BY*.

Ví dụ 8-11: SQL dạng SELECT với Group By và các hàm

```
Select CustID,
Count (CustID),Sum(Amount),
Max(Amount),
Min(Amount),
Avg(Amount)
From tblOrders
Group by CustID
```

```
Order by CustID
```

Kết quả trả về như sau:

```
CustID
---------- ----------- --------------------------
16      7  2956562368 1.95713899  26000   422366052
12      9  3843022604 39879489    459162  427002511
13      8  145913378  1.36727628  230000  18239172.25
10      4  72382804   48168567    2903576 18095701
```

5.3.2. Các hàm xử lý chuỗi

- Hàm *ASCII*: Hàm trả về giá trị mã *ASCII* của ký tự bên trái của chuỗi, ví dụ như khai báo:

  ```
  Select ASCII('TOI')
  ```

  Kết quả trả về như sau:

  ```
  84
  ```

- Hàm *Char*: Hàm này chuyển đổi kiểu mã *ASCII* từ số nguyên sang dạng chuỗi:

  ```
  Select char(35)
  ```

  Kết quả trả về như sau:

  ```
  #
  ```

- Hàm *UPPER*: Hàm này chuyển đổi chuỗi sang kiểu chữ hoạ:

  ```
  Select UPPER('Khang')
  ```

  Kết quả trả về như sau:

  ```
  KHANG
  ```

- Hàm *LOWER*: Hàm này chuyển đổi chuỗi sang kiểu chữ thường:

  ```
  Select LOWER('Khang')
  ```

  Kết quả trả về như sau:

  ```
  khang
  ```

- Hàm *Len*: Hàm này trả về chiều dài của chuỗi:

  ```
  Select len('I Love You')
  ```

  Kết quả trả về như sau:

  ```
  10
  ```

- Thủ tục *LTRIM*: Thủ tục loại bỏ khoảng trắng bên trái của chuỗi:

  ```
  Select ltrim(' Khang')
  ```

  Kết quả trả về như sau:

  ```
  'khang'
  ```

- Thủ tục *RTRIM*: Thủ tục loại bỏ khoảng trắng bên phải của chuỗi:

```
Select ltrim('Khang ')
```

Kết quả trả về như sau:

```
'khang'
```

- Hàm *Left*: Hàm trả về chuỗi bên trái tính từ đầu cho đến vị trí thứ *n*:

```
Select left('Khang',3)
```

Kết quả trả về như sau:

```
'Kha'
```

- Hàm *Right*: Hàm trả về chuỗi bên phải tính từ cuối cho đến vị trí thứ *n*:

```
Select Right('KHang',4)
```

Kết quả trả về như sau:

```
'Hang'
```

- Hàm *Instr*: Hàm trả về vị trí chuỗi bắt đầu của chuỗi con trong chuỗi xét:

```
Select INSTR ('Khang','Pham Huu Khang')
```

Kết quả trả về như sau:

```
11
```

11 là tương đương vị trí thứ 11 của chữ *Khang* trong chuỗi *"Pham Huu Khang"*

5.3.3. Các hàm về xử lý thời gian

- Hàm *CurDate*(): Hàm trả về ngày, tháng và năm hiện hành của hệ thống:

```
Select curdate() as 'Today is'
```

Kết quả trả về như sau

```
Today is
--------------------------
2001-11-21
```

- Hàm *CurTime*(): Hàm trả về giờ, phút và giây hiện hành của hệ thống:

```
Select curtime() as 'Time is'
```

Kết quả trả về như sau

```
Time is
--------------------------
09:12:05
```

- Hàm *Period_Diff*: Hàm trả về số ngày trong khoảng thời gian giữa 2 ngày:

```
Select
Period_diff (OrderDate, getdate())
```

```
as 'So ngay giua ngay thu tien den hom nay:'
from tblOrders
```

Kết quả trả về như sau

```
So ngay giua ngay thu tien den hom nay:
---------------------------------
74
72
```

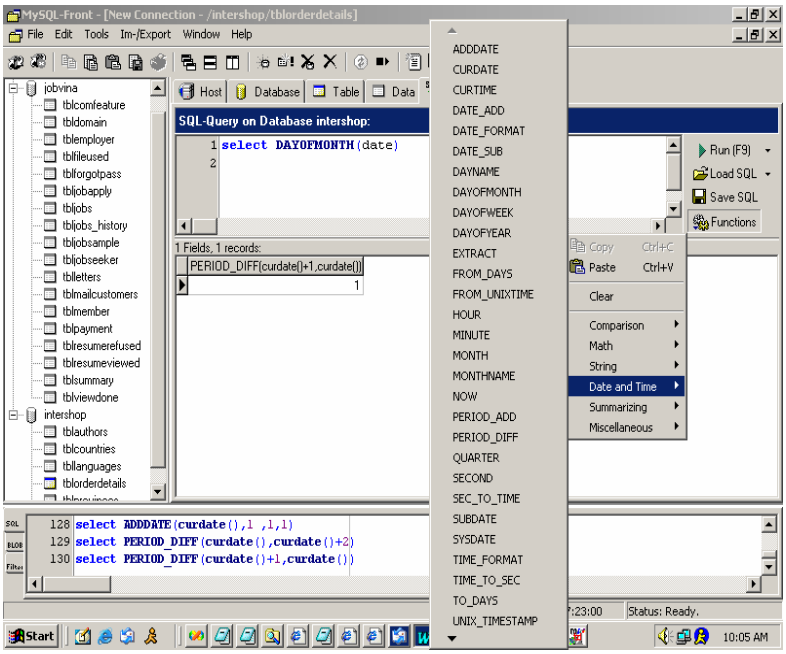
- Hàm *dayofmonth*: Hàm *dayofmonth* trả về ngày thứ mấy trong tháng:

```
Select dayofmonth(curdate())
as 'hom nay ngay
```

Kết quả trả về như sau:

```
21
```

Ngoài các hàm trình bày như trên, bạn có thể tìm thấy nhiều hàm xử lý về thời gian trong phần *Funtions* xuất hiện bên phải màn hình của trình điều khiển như hình 8-6.



Hình 8-6: Sử dụng chức năng Funcitons

5.3.4. Các hàm về toán học

- Hàm *sqrt*: Hàm trả về là căn bật hai của một biểu thức:

```
Select sqrt (4)
```

Kết quả trả về là

```
2
```

- Hàm *Round*: Hàm trả về là số làm tròn của một biểu thức:

```
Select round (748.58,-1)
```

Kết quả trả về là

```
7500
```

Để tham khảo thêm một số hàm khác bạn có thể tham khảo trong phần *Functions* như hình 8-9.

5.4. Phát biểu SQL dạng Select với AS

Khi cần thiết phải thay đổi tên trường nào đó trong câu truy vấn, bạn chỉ cần dùng phát biểu *AS*. *AS* cho phép ánh xạ tên cũ, hay giá trị chưa có tên thành tên mới (*header*).

Ví dụ, khi sử dụng *GROUP BY* ở trong phần trên, những cột tạo ra từ các phép toán *count*, *sum*, *max*, *min*, ... cho ra kết quả không có *header*, nghĩa là không có tên cột để tham chiếu trong khi gọi đến chúng. Chúng ta phải cần phát biểu *AS* cho những trường hợp này.

Ví dụ 4-11: SQL dạng SELECT với AS và các hàm

```
Select CustID,
Count (CustID) as No,
Sum(Amount) as TIENHD,
Max(Amount) as HDLONNHAT,
Min(Amount) as HDNHONHAT,
Avg(Amount) as TRUNGBINH
From tblOrders
Group by CustID
Order by CustID
```

Kết quả hiển thị như sau:

```
CustID No TIENHD    HDLONNHAT   HDNHONHAT TRUNGBINH
---------- ----------- ------------------------
16      7  2956562368 1.95713899  26000    422366052
12      9  3843022604 39879489    459162   427002511
13      8  145913378  1.36727628  230000   18239172.25
10      4  72382804   48168567    2903576  18095701
```

5.5. Phát biểu SQL dạng Select với Limit N , M

Phát biểu *SQL* dạng *SELECT* cho phép truy lục chỉ một số mẩu tin tính từ vị trí thứ *n* đến vị trí thứ *m* trong *Table* (theo một tiêu chuẩn hay sắp xếp nào đó). Để làm điều này, trong phát biểu *SQL* dạng *SELECT* bạn dùng chỉ định từ khoá *LIMIT* với số lượng mẩu tin cần lấy từ vị trí thứ *n* đến *m*.

Chẳng hạn, trong trường hợp bạn khai báo *Select * from tblOrders limit* 0,10. Kết quả sẽ trả về 10 mẩu tin đầu tiên trong bảng *tblOrders*.

Bạn cũng có thể sử dụng kết hợp *LIMIT* với các mệnh đề như *WHERE*, *ORDER BY* nhằm tạo ra kết quả như ý muốn.

Do yêu cầu khác nhau thông qua phát biểu *SQL* dạng *SELECT* có sử dụng *LIMIT,* nghĩa là kết quả trả về số lượng 10 mẩu tin đầu tiên với tất cả các cột trong bảng *tblOrders*

Ví dụ 8-12: Phát biểu SQL dạng SELECT với Limit N,M

```
Select *
From tblOrders
Limit 0,10
```

Kết quả trả về như sau:

```
OrderID OrderDate  CustID   Amount
--------- ------------------------- -------
01  2001-09-05     10       2903576
02  2001-09-05     10       48168567
03  2001-09-05     10       5107032
04  2001-09-08     10       2.3555347
05  2001-09-08     16       1.81074847
06  2001-09-10     16       26000
07  2001-09-11     16       1867682
08  2001-09-12     16       3600000
09  2001-09-13     16       1.95713899
10  2001-09-14     16       9.61804228
```

Nếu muốn lọc ra 10 hợp đồng có số tiền nhiều nhất, bạn chỉ cần sử dụng sắp xếp theo cột *TotalAmount* hay *Amount* trong bảng *tblOrders*.

Ví dụ 8-13: Phát biểu SQL dạng SELECT với Limit N,M

```
Select OrderID,OrderDate,CustID,Amount
From tblOrders
Order by Amount Desc
Limit 0,10
```

Kết quả trả về như sau:

```
OrderID OrderDate    CustID      Amount
---------- -------------------------- -------
06        2001-09-10    16         26000
26        2001-09-13    13         230000
25        2001-09-11    13         244904
23        2001-09-11    12         459162
31        2001-09-16    13         459525
27        2001-09-14    13         603033
28        2001-09-15    13         1452000
30         2001-09-15    13        1531200
07        2001-09-11    16         1867682
01        2001-09-05    10         2903576
```

Nếu muốn lọc ra 10 sản phẩm có số lượng bán nhiều nhất, bạn chỉ cần sử dụng sắp xếp theo cột số lượng *Qtty*.

Ví dụ 8-14: Phát biểu SQL dạng Select với Limit N,M

```
Select ItemID,Qtty,Price,Amount
from tblOrderDetails
Where Amount>10
order by Qtty
Limit 0,10
```

Kết quả trả về như sau:

```
ItemID               Qtty Price    Amount
-----------------------------------------------
1                    900    12000    12960000
```

```
2                    1000      12000    14400000
3                    5000      12000    72000000
3                    6000      12000    86400000
4                    8000      12000    15200000
4                    8000      12000    15200000
4                    8000      10000    15200000
5                    9000      12000    29600000
5                    9000      12000   129600000
5                    9000      12000   129600000
```

5.6. Phát biểu SQL dạng SELECT với DISTINCT

Nếu có một hay nhiều bảng kết nối với nhau, sẽ xảy ra trùng lặp nhiều mẩu tin. Nhưng trong trường hợp này bạn chỉ cần lấy ra một mẩu tin trong tập mẩu tin trùng lặp, bạn sử dụng phát biểu *SQL* dạng *SELECT* với chỉ định *DISTINCT*.

Ví dụ 8-14: Phát biểu SQL dạng SELECT

```
Select ItemID,Qtty,Price,Amount
from tblOrderDetails
order by Qtty
```

Kết quả trả về như sau:

```
ItemID               Qtty   Price    Amount
-----------------------------------------------
1                     900    12000    12960000
2                    1000    12000    14400000
3                    5000    12000    72000000
3                    6000    12000    86400000
4                    8000    12000    115200000
4                    8000    12000    115200000
4                    8000    10000    115200000
5                    9000    12000    129600000
5                    9000    12000    129600000
5                    9000    12000    129600000

...
...
```

Ví dụ 8-15: Phát biểu SQL dạng SELECT với DISTINCT

```
Select Distinct ItemID,Qtty,Price,Amount
From tblOrderDetails
Order by Qtty
```

Kết quả loại bỏ những mẩu tin trùng lắp như sau:

```
ItemID               Qtty   Price     Amount
-----------------------------------------------
1                     900    12000    12960000
2                    1000    12000    14400000
3                    6000    12000    86400000
4                    8000    12000    115200000
5                    9000    12000    129600000
...
...
...
```

5.7. Nhập dữ liệu bằng phát biểu SQL dạng Insert

Khi cần thêm mẩu tin vào bảng trong cơ sở dữ liệu *MySQL*, bạn có nhiều cách để thực hiện công việc này. Trong *Visual Basic 6.0, VB.NET, C Sharp* hay *Java* có những phương thức để thêm mẩu tin vào bảng trong cơ sở dữ liệu. Tuy nhiên, để sử dụng các phát biểu *SQL* mang tính chuyên nghiệp trong *MySQL*, bạn cần sử dụng phát biểu *INSERT*.

Bạn có thể sử dụng phát biểu *Insert* ngay trên ứng dụng kết nối với *MySQL*. Trong trường hợp bạn sử dụng cơ sở dữ liệu *SQL Server* hay *Oracle*, bạn có thể tạo ra một *Stored Procedure* với mục đích *INSERT* dữ liệu vào bảng chỉ định trước.

Khi thêm dữ liệu, cần chú ý kiểu dữ liệu giống hoặc tương ứng kiểu dữ liệu đã khai báo của cột đó, nếu không phù hợp thì lỗi sẽ phát sinh.

Ngoài ra bạn cần quan tâm đến quyền của *User* đang truy cập cơ sở dữ liệu. *User* phải được cấp quyền *Insert* dữ liệu vào từng bảng cụ thể (quyền này do nhà quản trị cơ sở dữ liệu phân quyền cho *User* đó).

Trong phát biểu *INSERT INTO* chúng tôi thực hiện trên bảng *tblOrderDetails* và bảng *tblOrderDetailsHist*, hai bảng này có cấu trúc như sau:

```
/* Bảng tblOrderDetails*/
CREATE TABLE tblorderdetails (
     ItemID int(3) unsigned DEFAULT '0' ,
     OrderID int(3) unsigned DEFAULT '0' ,
     No tinyint(3) unsigned DEFAULT '0' ,
     Qtty int(3) unsigned DEFAULT '0' ,
     Price int(3) unsigned DEFAULT '0' ,
     Discount int(3) unsigned DEFAULT '0' ,
     Amount bigint(3) unsigned DEFAULT '0'
);
```

```
/* Bảng tblOrderDetailsHist, dùng để chứa các thông tin
hợp đồng chi tiết khi hợp đồng của khách hàng này kết thúc,
chương trình tự động xoá trong tblOrderDetails và lưu trữ lại
trong bảng tblOrderDetailsHist.*/

CREATE TABLE tblorderdetailshist (
     ItemID int(3) unsigned DEFAULT '0' ,
     OrderID int(3) unsigned DEFAULT '0' ,
     No tinyint(3) unsigned DEFAULT '0' ,
     Qtty int(3) unsigned DEFAULT '0' ,
     Price int(3) unsigned DEFAULT '0' ,
     Discount int(3) unsigned DEFAULT '0' ,
     Amount bigint(3) unsigned DEFAULT '0'
);
```

Khi *Insert* dữ liệu vào bảng, có 3 trường hợp xảy ra: *insert* dữ liệu vào bảng từ các giá trị cụ thể, *insert* vào bảng lấy giá trị từ một hay nhiều bảng khác, và cuối cùng là kết hợp cả hai trường hợp trên.

5.7.1. Insert vào bảng lấy giá trị cụ thể:

```
INSERT INTO <Tablename>[<columnname list>]
Values (data_value)
```

Ví dụ 8-16: INSERT dữ liệu vào bảng từ giá trị cụ thể

```
/* Thêm mẩu tin với một số cột */

INSERT INTO
TBLCUSTOMERS
     (CustName,Username,Password,
     Address,Tel,FaxNo,Email,Contact,
     CountryCode,ProvinceCode)
Values ('Khach San CENTURY', 'century',
     '1111','5 Le Loi','8676767','8767676',
     'century@yahoo.com','Hoang Anh',
     'VNA','HCM')


/* Thêm mẩu tin với một số cột */

INSERT INTO
TBLORDERS(OrderID,OrderDate,
CustID,Description,Amount)
Values ('11',curdate(),'1',
   'Dat hang qua mang', 20000)
```

5.7.2. Insert vào bảng lấy giá trị từ bảng khác:

```
INSERT INTO <Tablename1>[<columnname list>]
Select [columnname list]
From <Tablename2>
Where <Conditions>
```

Ví dụ 8-17: INSERT vào bảng từ giá trị của bảng khác

```
/* Thêm mẩu tin với các cột cụ thể */

/* Chuyển tất cả những hợp đồng chi tiết từ bảng

 tblOrderDetails vào bảng tblOrderDetailsHist */


INSERT INTO
TBLORDERDETAILSHIST(
ItemID,
OrderID,
No,
Qtty,
Price,
Discount,
Amount)

SELECT
ItemID,
OrderID,
No,
Qtty,
Price,
Discount,
Amount
From tblOrderDetails
ORDER BY OrderID ASC


/* Có thể viết lại thêm mẩu tin với tất cả các cột như sau

 Chuyển tất cả những hợp đồng chi tiết từ bảng tblOrderDetails vào bảng tblOrderDetailsHist với điều
kiện số cột tương ứng trong bảng tblOrderDetails bằng với số cột trong bảng tblOrderDetailsHist, bạn có
thể viết lại như sau */
```

```
INSERT INTO TBLORDERDETAILSHIST
SELECT * from
tblOrderDetails
ORDER BY OrderID ASC
```

5.7.3. Insert vào bảng lấy giá trị cụ thể, bảng khác:

```
INSERT INTO <Tablename1>[<columnname list>]
Select [columnname list], valueslist
From <Tablename2>
Where <conditions>
ORDER BY <column name> ASC/DESC
```

Ví dụ 8-18: INSERT vào bảng từ giá trị cụ thể, bảng khác

/ Thêm mẩu tin với các cột cụ thể */*

/ Chuyển tất cả những hợp đồng chi tiết từ bảng tblOrderDetails vào bảng tblOrderDetailsHist. Giả sử rằng, ngoài những cột giống như tblOrderDetails, bảng tblOrderDetailsHist còn có thêm cột Tranferdate. */*

```
INSERT INTO
TBLORDERSHIST(
OrderID,
OrderDate,
ReceiveFolio,
CustID,
Descriion,
Amount,
Historydate)

SELECT
OrderID,
OrderDate,
ReceiveFolio,
CustID,
Descriion,
Amount,
getdate() as Historydate
From tblOrders
where Month(OrderDate)=12
Order by OrderDate,CustID
```

/ Có thể viết lại thêm mẩu tin với tất cả các cột như sau */*

/ Chuyển tất cả những phiếu thu trong tháng 12 từ bảng tblOrders vào bảng tblOrdersHist với điều kiện số cột tương ứng trong bảng tblOrders bằng với số cột trong bảng tblOrdersHist, bạn có thể viết lại như sau */*

```
INSERT INTO
TBLORDERDETAILSHIST(
ItemID,
OrderID,
No,
Qtty,
Price,
Discount,
Amount,TranferDate)

SELECT
ItemID,
```

```
OrderID,
No,
Qtty,
Price,
Discount,
Amount,CurDate()
From tblOrderDetails
ORDER BY OrderID ASC
```

5.8. Phát biểu SQL dạng UPDATE

Phát biểu *SQL* dạng *UPDATE* dùng cập nhật lại dữ liệu đã tồn tại trong bảng. Khi *UPDATE* dùng cập nhật dữ liệu cho một mẩu tin chỉ định nào đó thường *UPDATE* sử dụng chung với mệnh đề *WHERE*.

Nếu cần cập nhật tất cả các mẩu tin trong bảng bạn có thể bỏ mệnh đề *WHERE*. Phát biểu này có cấu trúc như sau:

```
/* nếu cập nhất giá trị cụ thể */
Update <table name>
Set <column>=<value>,[<column>=<value>]
[where <restrictive conditions>]

/* nếu cập nhất giá trị là kết quả trả về từ phát biểu
select trên một hay nhiều bảng khác */

Update <table name>
Set <column>=<select .. from tablename where ...>
[where <restrictive conditions>]
```

UPDATE có thể ảnh hưởng đến nhiều bảng, nhưng cập nhất giá trị chỉ có hiệu lực trên bảng đó, bạn có thể tham khảo phần này trong chương kế tiếp *JOIN TABLE*.

Cập nhật giá trị cụ thể vào một hay nhiều cột minh hoạ trong ví dụ 8-18 sau:

Ví dụ 8-18: UPDATE trên các cột dữ liệu từ giá trị cụ thể

```
/* cập nhật cột với giá trị cụ thể */

Update tblCustomers
Set CustName='Cong ty TNHH Coca cola Vietnam'
Where CustID='12'

/* cập nhật một cột với giá trị cột khác trong bảng
tblOrderDetails*/

Update tblOrders
Set Amount= Amount*.01,
TotalAmount=Amount*0.1
Where Month(OrderDate)=12

/* cập nhật một cột với giá trị từ bảng khác*/
/* cập nhật cột Price với giá trị từ cột Cost của bảng tblItems, khai báo sau chỉ đúng trong MySQL 4.1 trở
về sau*/


Update tblOrderDetails
```

```
Set Price=
(select distinct Cost]
from tblItems
where ItemID=tblOrderDetails.ItemID)
Where Price<1000
```

/ cập nhật một cột với giá trị cụ thể với điều kiện từ bảng khác, , khai báo sau chỉ đúng trong MySQL 4.1 trở về sau */*

```
Update tblOrderDetails
Set Price= Price*10,
Amount= Qtty*(Price+1)
Where ItemID in
(select distinct ItemID
from tblOrderDetails
where Price>1000)
```

5.9. Phát biểu SQL dạng DELETE

Với phát biểu *SQL* dạng *DELETE* thì đơn giản hơn. Khi thực hiện lệnh xoá mẩu tin trong bảng chúng ta chỉ cần quan tâm đến tên bảng, và mệnh đề *WHERE* để xoá với những mẩu tin đã chọn lọc nếu có. Cú pháp của *Delete*:

```
Delete from <table name>
Where <condition>
```

Với mệnh đề *WHERE* giống như bất kỳ mệnh đề *WHERE* nào trong phát biểu *SELECT* hay *UPDATE* và *INSERT* của bất kỳ ứng dụng cơ sở dữ liệu nào có sử dụng *SQL*.

Conditions có thể là phép toán giữa các cột và giá trị, nhưng cũng có thể giá trị là kết quả trả về từ một phát biểu *SELECT* khác.

Ghi chú: Không có khái niệm xóa giá trị trong một cột, vì xóa giá trị một cột đồng nghĩa với cập nhật cột đó bằng giá trị rỗng.

Ví dụ 8-19: Xóa mẩu tin với phát biểu SQL dạng DELETE

/ Xoá mẩu tin từ bảng với điều kiện */*

```
Delete from tblCustomers
Where CustName is null
```

Trong trường hợp có ràng buộc về quan hệ của dữ liệu, thì xóa mẩu tin phải tuân thủ theo quy tắc: Xoá mẩu tin con trước rồi mới xoá mẩu tin cha.

Chẳng hạn, trong trường hợp ta có 2 bảng: hợp đồng bán hàng (*tblOrders*) và hợp đồng bán hàng chi tiết (*tblOrderDetails*).

Để xoá một hợp đồng bạn cần xóa mẩu tin trong bảng *tblOrders* trước rồi mới đến các mẩu tin trong bảng *tblOrderDetails*.

Ví dụ 8-20: Xoá mẩu tin với Delete

```
/* Xoá mẩu tin từ bảng con */
Delete from tblOrderDetails
where OrderID=123

/* Xoá mẩu tin từ bảng cha */
Delete from tblOrders
where OrderID=123
```

Bạn có thể thực hiện một phát biểu *SQL* dạng *DELETE* với điều kiện trong mệnh đề *WHERE* lấy giá trị trả về từ phát biểu *SELECT* từ bảng khác, khai báo như vậy chỉ có hiệu lực trong cơ sở dữ liệu *MySQL* phiên bản 8.1 trở về sau hay trong cơ sở dữ liệu *SQL Server* và *Oracle*.

Ví dụ 8-21: Xoá mẩu tin theo quy tắc có ràng buộc quan hệ

```
/* Xoá mẩu tin từ bảng với điều kiện lấy giá trị từ bảng khác */
Delete from tblOrderDetails
where ItemID in
(select ItemID
from tblItems
where ItemName like 'IT%')
```

6. PHÁT BIỂU SQL DẠNG JOIN

Ngoài các phát biểu *SQL* với 4 dạng trên, trong phần kế tiếp, chúng tôi trình bày một số phát biểu *SQL* dạng *Select* để kết nối dữ liệu giữa các bảng có quan hệ với nhau, những phát biểu sẽ trình bày trong chương 5 như:

- *Khái niệm JOIN*
- *Phát biểu INNER JOIN*
- *Phát biểu LEFTJOIN*
- *Phát biểu RIGHT JOIN*

6.1. Khái niệm về quan hệ

Để phát triển ứng dụng *Web* bằng bất kỳ loại cơ sở dữ liệu nào, giai đoạn phân tích thiết kế hệ thống cực kỳ quan trọng. Nếu kết quả phân tích không tối ưu thì ứng dụng đó không thể đạt được giá trị kỹ thuật cũng như giá trị thương mại. Thiết kế cơ sở dữ liệu không tối ưu, chúng có thể dẫn đến việc chương trình chạy chậm và không bền vững.

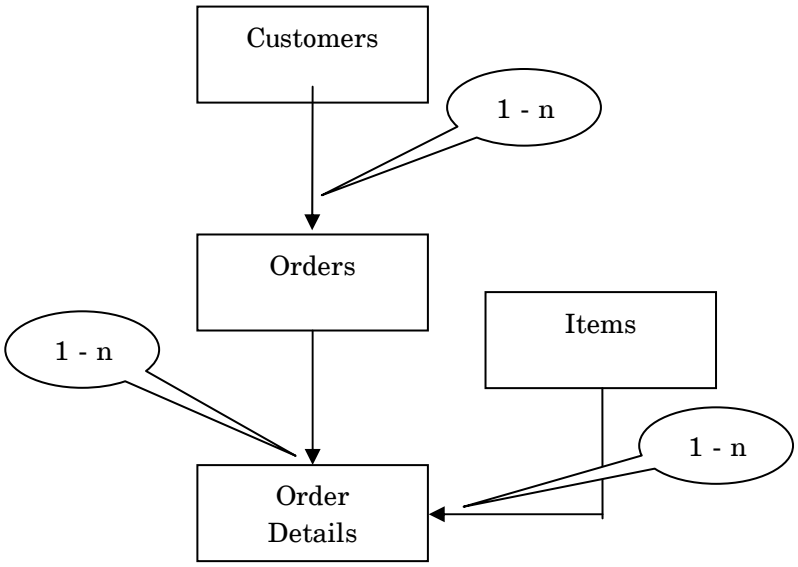
Một khi ứng dụng chạy chậm đi do cơ sở dữ liệu không tối ưu thì rất có thể bạn phải thiết kế và xây dựng lại từ đầu toàn bộ cấu trúc của chương trình và cơ sở dữ liệu.

Xuất phát từ lý do này, khi xây dựng một ứng dụng thông tin quản lý, chúng ta cần phải qua những bước phân tích thiết kế hệ thống kỹ lưỡng để có được mô hình quan hệ và *ERD* trước khi đến các mô hình chức năng chi tiết.

Tuy nhiên, trong lý thuyết một số kiến thức cơ bản bắt buộc bạn phải thực hiện theo mô hình hệ thống ứng với những quan hệ toàn vẹn, nhưng trong thực tế, do tính đặc thù của ứng dụng, thường bạn phải thiết kế lại mô hình theo nhu cầu cân đối giữa độ phức tạp và tính tối ưu.

Trong ứng dụng bán hàng qua mạng *Test* đã trình bày trong chương 3, khi quan tâm đến một hợp đồng trên mạng, ngoài những thông tin liên lạc về khách hàng, bạn cần phải lưu trữ dữ liệu khác như chiết hàng mua, phương thức trả tiền, phương thức giao hàng,... Vấn đề được thảo luận ở đây, mỗi hợp đồng có nhi6u mặt hàng chi tiết.

Trong trường hợp này, chúng ta có 6 thực thể liên quan như sau, thực thể danh mục *Customers* (thông tin liên lạc của khách hàng), *Orders* (hợp đồng mua hàng), *OrderDetails* (chi tiết hàng mua), *Items* (danh mục sản phẩm).



Sơ đồ 8-1: Mô hình quan hệ

Giả sử rằng khi nhập số liệu vào cơ sở dữ liệu, ứng với hợp đồng có mã 101, của khách hàng có tên Nguyễn Văn A, ... có hai sản phẩm chi tiết: 11 (Nước ngọt) và 32 (xà phòng *Lux*).

Trong trường hợp này bạn đang có một mẩu tin hợp đồng trong bảng *tblCustomers,* một mẩu tin hợp đồng trong bảng *tblOrders* và hai mẩu tin trong bảng *tblOrderDetails*.

Nếu muốn biết thông tin hợp đồng của khách hàng A, rõ ràng bạn cần dùng phát biểu *SELECT* với mệnh đề kết hợp từ 3 bảng trên. Kết quả trả về 2 mẩu tin là sự kết hợp thông tin từ hai bảng *tblCustomers*, *tblOrders* và *tblOrderDetails*.

Khi thực thi phát biểu *SQL* dạng *SELECT* ứng với cơ sở dữ liệu như trên bạn phải duyệt qua hai mẩu tin.

Tất nhiên, khi viết ứng dụng thì điều này chấp nhận được, và có thể coi là tối ưu. Giả sử rằng, ứng dụng này được phát triển trên *WEB* cần lưu tâm đến vấn đề tối ưu tốc độ truy vấn thì sao?

Người thiết kế cơ sở dữ liệu trong trường hợp này phải thay đổi lại cấu trúc để tăng tốc độ truy cập qua mạng khi xử lý trên cơ sở dữ liệu của người dùng.

6.2. Khái niệm về mệnh đề JOIN

Trong hầu hết phát biểu *SELECT*, phần lớn kết quả mà bạn mong muốn lấy về đều có liên quan đến một hoặc nhiều bảng khác nhau. Trong trường hợp như vậy, khi truy vấn dữ liệu bạn cần sử dụng mệnh đề *JOIN* để kết hợp dữ liệu trên hai hay nhiều bảng lại với nhau.

Khi sử dụng *JOIN*, bạn cần quan tâm đến trường (cột) nào trong bảng thứ nhất có quan hệ với trường (cột) nào trong bảng thứ hai. Nếu mô hình quan hệ của bạn không tối ưu hay không đúng, quản trình sử dụng *JOIN* sẽ cho kết quả trả về không như ý muốn.

Trở lại ứng dụng bán hàng qua mạng trong giáo trình này, khi xuất một hợp đồng bán hàng cho khách hàng, theo thiết kế trong cơ sở dữ liệu chúng ta có rất nhiều bảng liên quan đến nhau.

Chẳng hạn, nếu quan tâm bán hàng thì bán cho ai. Suy ra, liên quan đến thông tin khách hàng, bán sản phẩm gì cho họ thì liên quan đến mã sản phẩm, nếu khách hàng trả tiền thì liên quan đến phiếu thu, nếu khách hàng có công nợ thì liên quan đến nợ kỳ trước...

Trong phân này, chúng tôi tiếp tục thiết kế một số bảng dữ liệu cùng với kiểu dữ liệu tương ứng và quan hệ giữa các bảng được mô tả như sau:

```
tblCustomers (danh sách khách hàng)
   [CustID] int auto_increment Primary key,
   [CustName] [varchar] (50) NULL ,
   [Address] [varchar] (100) NULL,
   [Tel] [varchar] (20) NULL,
   [FaxNo] [varchar] (20) NULL,
   [Email] [varchar] (50) NULL,
   [Contact] [varchar] (50) NULL
   [Country] [varchar] (3) NULL,
   [Province] [varchar] (3) NULL
```

tblOrders (Hợp đồng bán hàng)

```
   [OrderID] [int] Not null
        auto_increment Primary Key,
   [OrderDate] [date] NULL ,
   [CustID] int ,
   [Description] [varchar] (200) NULL ,
   [ShipCost] [float] NULL ,
   [TranID] [tinyint] NULL ,
   [PaymentID] [tinyint] NULL ,
   [Amount] [float] NULL ,
   [TotalAmount] [float] NULL ,
```

tblOrderDetails (Hợp đồng bán hàng chi tiết)

```
   [SubID] [int] auto_increment  NOT NULL ,
   [OrderID] int ,
   [ItemID] int,
   [No] int,
   [Qtty] [int] NULL ,
   [Price] int NULL ,
   [Discount] [Float] NULL ,
   [Amount] [Float] NULL
```

tblItems (Danh sách sản phẩm)

```
   [ItemID] int auto_increment Primary key,
   [ItemName] [varchar] (200) NULL ,
   [Unit] [nvarchar] (20) NULL ,
   [Cost] [Float] NULL ,
   [Active] [tinyint] NOT NULL ,
   [Category] int
```

Bạn có thể tìm thấy các bảng dữ liệu còn lại trong dữ liệu *Test* trong đĩa đính kèm theo sách.

6.3. Mệnh đề INNER JOIN

Phát biểu *SQL* dạng *SELECT* có sử dụng mệnh đề *INNER JOIN* thường dùng để kết hợp hai hay nhiều bảng dữ liệu lại với nhau, cú pháp của *SELECT* có sử dụng mệnh đề *INNER JOIN*:

```
SELECT [SELECT LIST]
FROM <FIRST_TABLENAME>
INNER JOIN <SECOND_TABLENAME>
ON <JOIN CONDITION>
WHERE <CRITERIANS>
ORDER BY <COLUMN LIST>
[ASC / DESC]
```

Nếu bạn cần lấy ra một số cột trong các bảng có kết nối lại với nhau bằng mệnh đề *INNER JOIN* thì cú pháp này viết lại như sau:

```
SELECT [FIELD1,FIELD2, ...]
FROM <FIRST_TABLENAME>
INNER JOIN <SECOND_TABLENAME>
```

```
ON <JOIN CONDITION>
WHERE <CRITERIANS>
ORDER BY <COLUMN LIST>
[ASC / DESC]
```

Ví dụ 8-23: INNER JOIN với một số cột chỉ định

```
/* in ra danh sách khách hàng mua hàng trong tháng 10 */

Select CustName,OrderID,
OrderDate,Amount,
TotalAmount
from tblCustomers
inner join tblOrders
on tblCustomers.CustID = tblOrders.CustID
where month (OrderDate) = 10
order by CustName
```

Kết quả trả về như sau:

```
CustName       OrderID OrderDate .. TotalAmount
-------------------------------------------
CENTURY Hotel   13  2001-10-17   388800000
CENTURY Hotel   14  2001-10-18   518400000
CENTURY Hotel   16  2001-10-17   388800000
CENTURY Hotel   17  2001-10-18   14400000
CENTURY Hotel   18  2001-10-18   12960000
CENTURY Hotel   110 2001-10-18   216000000
Plaza Hotel     12  2001-10-17   403200000
Plaza Hotel     19  2001-10-17   86400000
Plaza Hotel     11   2001-10-17  576000000
Plaza Hotel     15   2001-10-17  288000000
```

Nếu bạn cần lấy ra tất cả các cột trong các bảng có kết nối lại với nhau bằng mệnh đề *INNER JOIN,* cú pháp trên có thể viết lại như sau:

```
SELECT first_tablename.*,
second_tablename.*
[,next table name]
FROM <first_tablename>
INNER JOIN <second_tablename>
ON <join conditions>
[INNER JOIN <next_tablename>
ON <join conditions>]
WHERE <conditions>
ORDER BY <column list>
[ASC / DESC]
```

Ví dụ 8-24: INNER JOIN với tất các trường liên quan

```
/* in ra danh sách khách hàng mua hàng trong tháng 10 */

Select CustID,CustName,OrderID,
OrderDate,TotalAmount
from tblCustomers
inner join tblOrders
On TblCustomers.CustID=tblOrders.CustID
where month (OrderDate) = 10
order by CustName DESC
```

Kết quả trả về như sau:

```
CustID   CustName    ..OrderID ..TotalAmount
------ ---------------------------------------
13     Plaza Hotel   ..11   ..   576000000
13     Plaza Hotel   ..15   ..   288000000
12     Plaza Hotel   ..12   . .  403200000
12     Plaza Hotel   ..19    ..  86400000
16     CENTURY Hotel ..13    ..  388800000
16     CENTURY Hotel ..14    ..  518400000
16     CENTURY Hotel ..16    ..  388800000
16     CENTURY Hotel ..17    ..  14400000
16     CENTURY Hotel ..18    ..  12960000
16     CENTURY Hotel ..110   ..  216000000
```

Nếu trong những bảng cần kết nối có tên trường (cột) giống nhau thì khi thực thi phát biểu *SQL* dạng *SELECT* phải chỉ rõ cột thuộc bảng nào. Trong trường hợp cả hai cùng lấy dữ liệu ra thì bạn cần chuyển ánh xạ tên khác cho cột thông qua mệnh đề *AS*, ví dụ như:

```
SELECT first_tablename.CustID as CUSTID,
  second_tablename.CustID as CUSTID
FROM <first_tablename>
INNER JOIN <second_tablename>
ON <join condition>
WHERE <criterians>
ORDER BY <column list>
[ASC / DESC]
```

Nếu trong những bảng cần kết nối đó có tên trường (cột) giống nhau và không được chỉ rõ như trường hợp trên khi khai báo trong cơ sở dữ liệu *SQL Server*, khi thực thi phát biểu *SQL* dạng *SELECT* bạn sẽ bị lỗi, chẳng hạn như:

```
SELECT first_tablename.*, second_tablename.*
FROM <first_tablename>
INNER JOIN <second_tablename>
ON <join condition>
WHERE <criterians>
ORDER BY <column list>
[ASC / DESC]
```

Server: Msg 209, Lecel 16, State Line 1
Ambiguous column name 'CustID'

Tuy nhiên, với phát biểu trên bạn có thể thực thi trong cơ sở dữ liệu *MySQL*. Ngoài ra, phát biểu *SQL* dạng *SELECT* sử dụng *INNER JOIN* bạn có thể ánh xạ (*alias*) tên của bảng thành tên ngắn gọn để dễ tham chiếu về sau.

Thực ra phát biểu *ALIAS* có ý nghĩa giống như *AS* với tên cột trong bảng thành tên cột khác trong phát biểu *SELECT*.

```
Select p.*,s.*
from tablename1
inner join tablename2
On tablename1.field1 = tablename2.field2
```

Ví dụ 8-25: INNER JOIN với ánh xạ tên bảng

/ in ra danh sách khách hàng mua hàng trong tháng 10 */*

```
Select c.CustName,
  s.OrderID,s.OrderDate,
  s.TotalAmount
from tblCustomer c
inner join tblOrders s
  On c.CustID=s.CustID
where month (s.OrderDate) = 10
order by  c.CustName DESC
```

Kết quả trả về như sau:

```
CustName        OrderID OrderDate .. TotalAmount
-----------------------------------------------CENTURY Hotel    13  2001-
10-17 ..    388800000
CENTURY Hotel    14  2001-10-18 ..    518400000
CENTURY Hotel    16  2001-10-17 ..    388800000
CENTURY Hotel    17  2001-10-18 ..    14400000
CENTURY Hotel    18  2001-10-18 ..    12960000
CENTURY Hotel    11   2001-10-18 ..   216000000
Plaza Hotel      12  2001-10-17 ..    403200000
Plaza Hotel      19  2001-10-17 ..    86400000
Plaza Hotel      11   2001-10-17 ..   576000000
Plaza Hotel      15   2001-10-17 ..   288000000
```

Tất nhiên, bạn cũng có thể viết phát biểu trên ứng với từng cột muốn lấy ra bằng cách khai báo tên cột.

6.4. Mệnh đề Left Join

Trường hợp bạn mong muốn kết quả lấy ra trong hai bảng kết hợp nhau theo điều kiện: Những mẩu tin bảng bên trái tồn tại ứng với những mẩu tin ở bảng bên phải không tồn tại bạn hãy dùng mệnh đề *LEFT JOIN* trong phát biểu *SQL* dạng *SELECT, c*ú pháp có dạng:

```
select <Column list>
from lefttablename
LEFT  JOIN righttablename
on lefttabkename.field1=righttablename.field2
Where <conditions>
Order by <column name>
ASC/DESC
```

Chẳng hạn, bạn chọn ra tất cả các sản phẩm (với các cột) có hay không có doanh số bán trong tháng hiện tại. Một số sản phẩm không bán trong tháng sẽ có cột *Amount* có cột *Amount* giá trị *NULL*.

Ví dụ 8-26: SELECT dùng LEFT JOIN

```
/* in ra danh sách sản phẩm bán trong tháng 10 */

select ItemID,ItemName,Amount
from tblItems
left  join tblOrderDetails
on tblItems.ItemID=tblOrderDetails.ItemID
order by Amount
```

Kết quả trả về như sau:

```
ItemID          ItemName             Amount
-----------------------------------------------
12              ASW-60VP               NULL
```

```
13              ASW-60VT                NULL
14              ASW-660T 120V TW 29340  NULL
14              ASW-685V 120V TW 29440  NULL
15              ASW60VP 220V 34571      NULL
16              ASW-45Z1T1              2960000
17              ASW-45Y1T 127V          14400000
18              ASW-45Y1T 220V          72000000
19              ASW-45Y1T 220V          86400000
20              ASW-45Z1T               15200000
...
```

6.5. Mệnh đề Right Join

Ngược lại với phát biểu *SQL* dạng *SELECT* sử dụng mệnh đề *LEFT JOIN* là phát biểu *SQL* dạng *SELECT* sử dụng mệnh đề *RIGHT JOIN* sẽ xuất dữ liệu của bảng bên phải cho dù dữ liệu của bảng bên trái không tồn tại, cú pháp có dạng:

```
Select <Column list>
From lefttablename
RIGHT JOIN righttablename
On lefttabkename.field1=righttablename.field2
Where <conditions>
Order by <column name>
ASC/DESC
```

Trong ví dụ sau, bạn có thể chọn ra tất cả các sản phẩm có hay không có doanh số bán trong tháng hiện tại. Các sản phẩm không tồn tại doanh số bán sẽ không hiện ra.

Ví dụ 8-27: SELECT dùng RIGHT JOIN

```
/* in ra danh sách sản phẩm bán trong tháng ngày 17 */
/* trong phát biểu SELECT này có sử dụng mệnh đề
WHERE sử dụng phát biểu SELECT khác, kết quả của SELECT trong mệnh đề WHERE trả về một mảng
OrderID */

Select ItemName,Qtty,
Price,Amount
From tblItems
Right join tblOrderDetails
On tblItems.ItemID=tblOrderDetails.ItemID
Where OrderID in (12,14,23,15)

Order by ItemID
```

Kết quả trả về như sau:

```
ItemName             Qtty Price  Amount
---------------------------------------------
ASW-45Y1T 127V SDIA29350  11000  12000  58400000
ASW-45Y1T 127V SDIA29350  10000  12000  44000000
ASW-45Y1T 127V SDIA 29350  10000  12000 14400000
ASW-45Y1T 127V SDIA 29350  10000  12000 44000000
ASW-45Y1T 127V SDIA 29350  11000  12000 58400000
ASW-45Y1T 127V SDIA 29350  10000  12000 44000000
ASW-45Y1T 127V SDIA 29350  11000  12000 58400000
ASW-45Y1T 220V ARG 29391   6000   12000 86400000
ASW-45Z1T                  9000   12000 29600000
ASW-45Z1T                  9000   12000 29600000
...
```

6.6. Phép toán hợp (union)

Union không giống như những mệnh đề *JOIN* đã giới thiệu trên đây. *Union* là phép toán dùng để nối hai hay nhiều câu truy vấn dạng *Select* lại với nhau.

Đối với *JOIN*, bạn có thể kết nối dữ liệu được thực hiện theo chiều ngang. Đối với *Union* bạn kết nối dữ liệu được thực hiện theo chiều dọc.

Để chọn ra những khách hàng thường xuyên trong *tblCustomers*, kết quả trả về là danh sách các khách hàng thường xuyên.

Ví dụ 8-28: Khách hàng thường xuyên trong tblCustomers

```
Select CustID,CustName
from tblCustomers
```

Kết quả trả về như sau:

```
CustID      CustName
----------- -----------------
13          New World Hotel
12          Kinh Do Hotel
16          CENTURY Hotel
10          PLAZA Hotel
```

Để chọn ra những khách hàng vãng lai trong *tblTempCustomers*, kết quả trả về là danh sách các khách hàng vãng lai.

Ví dụ 8-29: Khách hàng vãng lai trong tblTempCustomers

```
Select CustID,CustName
from tblTempCustomers
```

Kết quả trả về như sau:

```
CustID      CustName
----------- ---------------------------------
23          Cong ty nuoc giai khat '12'COLA
24          Cong ty nuoc giai khat PEPSI
25          Cong ty nuoc giai khat REDBULK
26          Cong ty nuoc giai khat TRIBICO
```

Nếu dùng phép toán *UNION* để kết nối hai bảng trên, kết quả trả về là danh sách cả hai loại khách hàng trong cùng một *recordset*.

Ví dụ 8-30: SELECT sử dụng phép hợp UNION

```
Select CustID,CustName
From tblCustomers

UNION

Select CustID,CustName
From tblTempCustomers
```

Kết quả trả về như sau:

```
CustID      CustName
```

```
---------- ----------------------------------
23         Cong ty nuoc giai khat '12'COLA
24         Cong ty nuoc giai khat PEPSI
25         Cong ty nuoc giai khat REDBULK
26         Cong ty nuoc giai khat TRIBICO
12         Kinh Do Hotel
10         PLAZA Hotel
16         CENTURY Hotel
13         New World Hotel
```

Ghi chú: Khi sử dụng phép toán *Union* trong phát biểu *SQL* dạng *Select*, bạn cần lưu ý các quy định sau:

- Tất cả những truy vấn trong *UNION* phải cùng số cột hay trường. Nếu truy vấn thứ nhất có hai cột thì truy vấn thứ hai được sử dụng *UNION* cũng phải có hai cột tương tự.
- Khi sử dụng *UNION*, những cột nào có tên cột hay bí danh (alias) mới thì kết quả trả về sẽ có tựa đề (*header*) của từng cột và tên là tên cột của truy vấn thứ nhất.
- Kiểu dữ liệu trong các cột của truy vấn 2 tương thích với kiểu dữ liệu các cột tương ứng trong truy vấn thứ nhất.
- Trong *UNION* bạn có thể kết hợp nhiều câu truy vấn lại với nhau.
- Kết quả hiện ra theo thứ tự của truy vấn từ dưới lên trên.

6.7. SQL dạng thay đổi và định nghĩa cơ sở dữ liệu

6.7.1. Phát biểu SQL dạng CREATE

Phát biểu *SQL* dạng *CREATE* dùng để tạo cơ sở dữ liệu và những đối tượng của cơ sở dữ liệu trong *MySQL*, *SQL Server*, *Oracle*, ..., chúng cú pháp như sau:

```
CREATE Database <Database NAME>

CREATE <OBJECT TYPE>
<OBJECT NAME>
```

- *OBJECT TYPE*: Loại đối tượng của cơ sở dữ liệu ví dụ như *Procedure*, *Table*, *View*,...
- *OBJECT NAME:* Tên của đối tượng trong cơ sở dữ liệu *SQL* như *sp_IC*, *tblEmployer*, ...

6.7.2. Tạo cơ sở dữ liệu - Create database

Khi xây dựng cơ sở dữ liệu, bạn bắt đầu từ mô hình cơ sở dữ liệu *ERD*, hay từ một giai đoạn nào đó trong quy trình phân tích thiết kế hệ thống. Để tạo cơ sở dữ liệu trên *MySQL* hay *SQL Server* bạn sử dụng cú pháp sau:

```
CREATE DATABASE <Database name>
```

Cú pháp đầy đủ của phát biểu tạo cơ sở dữ liệu như sau, nếu bạn sử dụng cơ sở dữ liệu *SQL Server*:

```
CREATE DATABASE <database_name>
[ ON [PRIMARY]  (
     [Name= <'Logical file name'>,]     FileName=<'File Name'>
     [, SIZE=<Size in Megabyte or KiloByte> ]
     [, MAXSIZE=<Size in Megabyte or KiloByte> ][, FILEGROWTH = <No of
     Kylobyte|Percentage>]
  )]
```

```
[ LOG ON
   (
      [Name= <'Logical file name'>,]     FileName=<'File Name'>
      [, SIZE=<Size in Megabyte or KiloByte> ]
      [, MAXSIZE=<Size in Megabyte or KiloByte> ][, FILEGROWTH = <No of
     Kylobyte|Percentage>]
   )]
[COLLATE <Collation Name>]
[For Load | For Attach]
```

6.7.3. Diễn giải CREATE Database trong SQL Server

- *ON*: Dùng để định nghĩa nơi chứa cơ sở dữ liệu và không gian chứa tập tin *log*.
- *NAME*: Dùng định nghĩa tên của cơ sở dữ liệu. Tên này dùng tham chiếu khi gọi đến cơ sở dữ liệu, tên được dùng cho quá trình *backup, export, Import, Shrink* cơ sở dữ liệu đó.
- *FILENAME*: Tên tập tin cơ sở dữ liệu lưu trong đĩa cứng, thông thường khi cài *SQL Server* lên ổ đĩa nào thì giá trị mặc định cho phép lưu tập tin đến thư mục đó. Tuy nhiên, nếu muốn bạn cũng có thể thay đổi vị trí các *file* này.

Khi tạo cơ sở dữ liệu, bạn đã định nghĩa vị trí đặt tập tin ở thư mục nào thì không thể di chuyển một cách thủ công (như dùng *Explorer* của *Windows*), vì làm điều đó thật nguy hiểm nhất là khi dữ liệu trong cơ sở dữ liệu đang có giá trị kinh tế.

- *SIZE*: Dung lượng của cơ sở dữ liệu khi khởi tạo chúng. Thông thường giá trị mặc định là *1 MB*.
- Dung lượng phải là số nguyên, có thể tăng thêm bằng cách sử dụng thủ tục *Shrink* trong *SQL Server*.
- *MAXSIZE*: Dung lượng lớn nhất, khi dung lượng cơ sở dữ liệu tăng lên đến mức *MaxSize* thì dừng lại.

Nếu khi dung lượng bằng *MaxSize*, các chuyển tác có thể bị huỷ bỏ hay trả về lỗi không thể thực hiện được, và có thể làm cho cơ sở dữ liệu của bạn bị treo.

Để tránh điều này xảy ra, thì người quản trị cơ sở dữ liệu phải thường xuyên theo giỏi quá trình tăng dung lượng cơ sở dữ liệu theo thời gian, để có biện pháp tránh mọi rủi ro có thể xảy ra.

- *FILEGROWTH*: Dung lượng khởi tạo cùng dung lượng tối đa cho phép tăng trong quá trình thêm dữ liệu vào cơ sở dữ liệu. Nhằm tự động hóa, chúng ta phải thiết lập quá trình tăng tự động theo chỉ số *KB* cho trước hay tỷ lệ phần trăm theo dung lượng đang có.
- *LOG ON*: *Log* on cho phép bạn quản lý những chuyển tác xảy ra trong quá trình sử dụng cơ sở dữ liệu của *SQL Server*.

Xây dựng cơ sở dữ liệu Test

Như đã trình bày ở trên, sau đây ví dụ tạo cơ sở dữ liệu *Test* có cú pháp như sau:

Ví dụ 8-31: Tạo cơ sở dữ liệu Test trong SQL Server

```
USE master

GO

CREATE DATABASE Test
   ON
   ( NAME = Test,
   FILENAME = 'c:\mssql7\data\Testdat.mdf',
```

```
  SIZE = 10,
  MAXSIZE = 50,
  FILEGROWTH = 5 )

  LOG ON
  ( NAME = 'Testlog',
  FILENAME = 'c:\mssql7\data\Testlog.ldf',
  SIZE = 5MB,
  MAXSIZE = 25MB,
  FILEGROWTH = 5MB )
  GO
```

Để đơn giản hoá các đối tượng *Table* trong cơ sở dữ liệu *Test*, chúng tôi chỉ trình bày một vài phát biểu *SQL* dạng *Create Table*, các *Table* khác bạn có thể tìm thấy trong cơ sở dữ liệu đính kèm.

Ví dụ 8-32: Tạo một số bảng trong Test

```
/* Tạo bảng danh sách khách hàng thường xuyên */

CREATE TABLE tblcustomers (
 CustID int(3) unsigned NOT NULL auto_increment,
 Username varchar(20) NOT NULL DEFAULT '' ,
 Password varchar(10) NOT NULL DEFAULT '' ,
 CustName varchar(50) ,
 Address varchar(100) ,
 Tel varchar(20) ,
 FaxNo varchar(10) ,
 Email varchar(50) ,
 Contact varchar(50) ,
 CountryCode char(3) ,
 ProvinceCode char(3) ,
 PRIMARY KEY (CustID),
 INDEX CustID (CustID)
);

/* Tạo bảng hợp đồng mua hàng qua mạng */

CREATE TABLE tblorders (
 OrderID int(3) NOT NULL auto_increment,
 OrderDate date ,
 CustID int(11) ,
 Description varchar(100) DEFAULT '0' ,
 TranID tinyint(3) DEFAULT '0' ,
 PaymentID tinyint(3) DEFAULT '0' ,
 Amount float DEFAULT '0' ,
 ShipCost float DEFAULT '0' ,
 TotalAmount float DEFAULT '0' ,
 PRIMARY KEY (OrderID),
 INDEX OrderID (OrderID)
);

/* Tạo bảng hợp đồng chi tiết mua hàng qua mạng */

CREATE TABLE tblorderdetails (
 ItemID int(3) unsigned DEFAULT '0' ,
 OrderID int(3) unsigned DEFAULT '0' ,
 No tinyint(3) unsigned DEFAULT '0' ,
 Qtty int(3) unsigned DEFAULT '0' ,
 Price int(3) unsigned DEFAULT '0' ,
 Discount int(3) unsigned DEFAULT '0' ,
 Amount bigint(3) unsigned DEFAULT '0'
);
```

Một số quy định khi thiết kế Table

6.7.4. Tên cột - Column Name

Đặt tên cột cũng giống như đặt tên bảng, có rất nhiều quy tắc đặt tên (như đã trình bày ở trên phần *table*), nhưng khuyến khích bạn nên theo một số quy tắc cơ bản sau:

- Tên cột bắt đầu chữ hoa, còn lại bằng chữ thường.
- Tên ngắn gọn và đầy đủ ý nghĩa.
- Không nên đặt tên cột có khoảng trắng, sau này bạn sẽ gặp những phiền toái khi tham chiếu đến cột đó.
- Không đặt tên cột trùng với những từ khoá, từ dành riêng, và những ký tự đặc biệt như những phép toán hay toán tử khác.
- Chú ý, nên đặt tên cột cùng tên những cột có quan hệ với những bảng khác trong cùng cơ sở dữ liệu, giúp dễ hiểu và tránh bị nhầm lẫn.

Một số người thích thêm vào dấu gạch chân (_) để phân biệt ý nghĩa hay tên gọi của cột, điều này là tùy vào sở thích của bạn. Tuy nhiên chúng tôi không thích qui tắc này.

Nhưng đối với kinh nghiệm lập thiết kế xây dựng cơ sở dữ liệu thì bạn không nên dùng dấu gạch dưới _, và dĩ nhiên trong nhiều trường hợp khác bạn sẽ cảm thấy khó chịu khi thêm một dấu _ trong tên của đối tượng của cơ sở dữ liệu.

Mặc dù không có vấn đề gì cho cú pháp hay các phát biểu tham chiếu đến chúng, nhưng bạn sẽ thấy tại sao chúng ta không nên dùng dấu gạch chân (_) khi đặt tên đối tượng hay tên cơ sở dữ liệu trong *MySQL*.

- Nếu bạn đặt tên có dấu _ ,bạn phải tốn thời gian hay năng lượng cho hành động tạo ra dấu _
- Trong chừng mực hay giới hạn nào đó do hiệu ứng của *Font* chữ có thể phát sinh lỗi sẽ gây ra nhầm lẫn cho người lập trình.
- Nói tóm lại là bạn sẽ mất thêm thời gian lưu tâm đến chúng.

6.7.5. Kiểu dữ liệu - Data type

Như đã trình bày các lại dữ liệu trong phần trên, khi xây dựng cơ sở dữ liệu, tất cả những trường trong bảng cần phải có kiểu dữ liệu cụ thể. Vấn đề quan trọng là chọn kiểu dữ liệu nào cho phù hợp với dữ liệu mà người dùng sẽ nhập vào.

Để thiết kế dữ liệu phù hợp với thực tế, ngoài tính ứng dụng hợp với ngữ cảnh bạn cũng cần quan tâm đến kiểu dữ liệu tương thích và chiều dài của từng cột. Chẳng hạn như:

```
[CustID] [varchar] (10)
/* hay */
[CustID] int
```

6.7.6. Giá trị mặc định - Default

Thông thường khi tạo ra một cột trong bảng đôi khi chúng ta cần áp dụng giá trị mặc định, không chỉ cho trường hợp số liệu không nhập từ bên ngoài mà còn cho các cột tự động có giá trị tự sinh. Với những lý do như vậy, chúng ta cần có một số giá trị mặc định cho những cột cần thiết, ví dụ :

- Nếu cột đó là số chúng ta có giá trị mặc định là 0

- Nếu cột đó là ngày tháng chúng ta có giá trị mặc định là ngày nào đó (như 0000-00-00 là *CurDate*())
- Nếu cột đó có giá trị là 0 hoặc 1, bạn có thể khai báo giá trị mặc định là 0 hoặc 1
- Nếu cột đó là chuỗi chúng ta có giá trị mặc định như là 'A'

6.7.7. Số tự động auto_increment

auto_increment là khái niệm cực kỳ quan trọng trong *MySQL* (tương đương với *Identity* trong *SQL Server, Autonumber* trong *MS Access*). Khi bạn muốn một cột có giá trị tăng tự động như *AutoNumber*/*Identity*, bạn nên định nghĩa cột đó như *auto_increment*,.

Khi sử dụng *auto_increment* làm số tăng tự động thì kiểu dữ liệu là số nguyên hoặc số nguyên lớn.

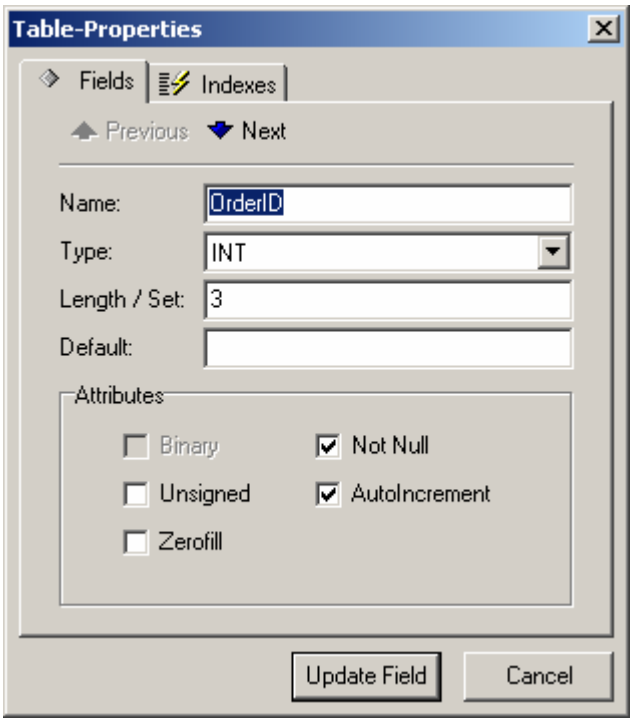
Trong trường hợp, bạn khai báo số tự động trong *SQL Server*, bạn cần phải khai báo thêm các thông số như *seed*. *Seed* là giá trị khởi đầu khi *SQL Server* tự động tăng giá trị, *Increament* là bước tăng, nó cho biết mỗi lần tăng cần bao nhiêu giá trị.

Vì dụ khi tạo *auto_increment* cho cột *ItemID [Int] auto_increment*, nghĩa là bắt đầu số 1 và mỗi lần tăng 1 số. Kết quả bạn sẽ có là 1,2,3,4, ...*n*.

Trong phát biểu *SQL* của *MySQL*, để tạo bảng có gá trị tăng tự động bạn chỉ cần khai báo tên cột, kiểu dữ liệu *Int (Integer)* và *auto_increment* như sau:

```
IDNO Int auto_increment NOT NULL
```

Trong giao diện đồ họa bạn chỉ cần *check* vào tuỳ chọn *AutoIncreament* như hình 8-10.



Hình 8-10: Chọn auto_increment

NULL / NOT NULL

Đây là trạng thái của một cột trong bảng cho phép chấp nhận giá trị *NULL* hay không? Nếu bạn chỉ ra ràng buộc giá trị *NOT NULL* thì bắt buộc phải có giá trị trong cột này mỗi khi mẩu tin được nhập vào.

Đối với một số kiểu dữ liệu không cho phép *NULL* bạn nên thiết lập giá trị mặc định cho cột đó, ví dụ như kiểu dữ liệu bit không cho phép *NULL*.

Trong phát biểu SQL tạo bảng, bạn chỉ cần khai báo *NULL* hay *NOT NULL* sau kiểu dữ liệu của cột đó. Trong giao diện đồ họa chỉ cần đánh dấu chọn vào tuỳ chọn *Not NULL* như hình 8-10.

6.8. Thay cấu trúc đối tượng bằng ALTER

Khi chúng ta cần thiết phải sửa đổi một phần cấu trúc của các đối tượng như *table (view*, hay *SP* trong *SQL Server*) vì mục đích nào đó, thì Bạn sử dụng phát biểu *ALTER* để thay đổi cấu trúc của đối tượng hiện có:

```
ALTER <Object type>
<Object Name>
```

Khi một bảng tồn tại trong cơ sở dữ liệu, do nhu cầu cần thiết phải thay đổi cấu trúc bảng, bạn sử dụng phát biểu *ALTER TABLE* cùng các tham số của chúng như cú pháp sau:

```
ALTER TABLE table alteration [,alteration]
```

Chẳng hạn, bạn có thể sử dụng phát biểu *ALTER TABLE* để thêm một cột tên Activate với kiểu dữ liệu TinyInt có giá trị mặc định là 1.

Ví dụ 8-33: Thêm một cột tên Activate vào bảng tblOrders

```
ALTER TABLE tblorders
ADD Activate TINYINT DEFAULT "1"
```

Khi thay đổi thiết lập giá trị mặc định cho cột bạn nên quan tâm đến giá trị mặc định đó có phù hợp cho những mẩu tin đang tồn tại hay không.

Muốn thay đổi giá trị mặc định của cột cho những mẩu tin đang tồn tại, bạn sử dụng đến mệnh đề phụ như trong ví dụ sau:

Ví dụ 8-34: Thiết lập giá trị mặc định trong bảng tblOrders

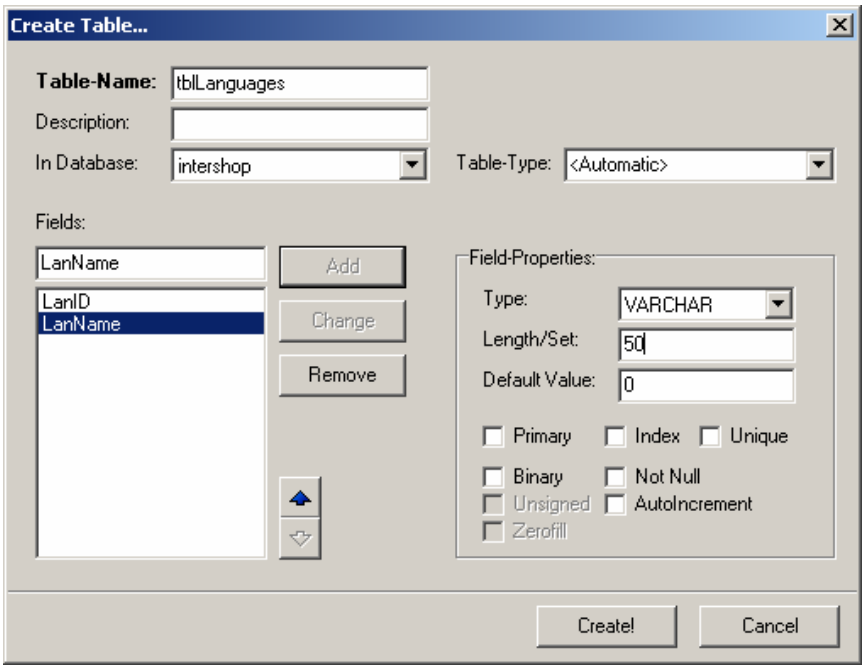
```
ALTER TABLE tblorders
CHANGE OrderDate
OrderDate DATETIME
DEFAULT "0000-00-00"
```

Thay đổi kiểu dữ liệu từ *Date* dang *DateTime*, bạn có thể khai báo như ví dụ 4-35 sau:

Ví dụ 8-35: Thay đổi kiểu dữ liệu

```
ALTER TABLE tblorders
CHANGE OrderDate
OrderDate DATE
DEFAULT "0000-00-00 00:00:00"
```

Mặc khác, bạn cũng có thể tạo hay thay đổi bảng trong màn hình *MySQL-Front*. Chỉ cần chọn ngăn *Database* | *R-Click* | *Create New Table*, cửa sổ xuất hiện như hình 8-11.



Hình 8-11: Giao diện tạo bảng bằng MySQL-Front

6.9. Phát biểu SQL dạng DROP

Drop là phát biểu thực hiện phép xoá. *DROP* dùng để xoá đối tượng của cơ sở dữ liệu như bảng, cơ sở dữ liệu, ...Cú pháp của phát biểu *DROP:*

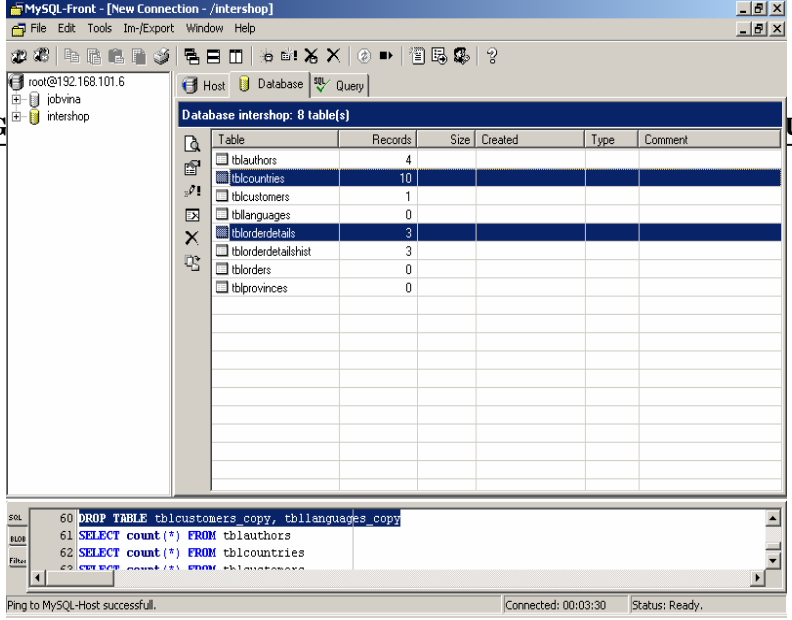
```
DROP <Object type> <Object name> [, .... n]
```

Bạn có thể xoá cơ sở dữ liệu, bằng cách khai báo như sau:

```
Drop Database Test
```

```
/* Phát biểu DROP TABLE chỉ rõ bảng nào cần xoá,
nếu xoá nhiều bảng thì bạn cần dùng dấu phẩy (,)  */
DROP TABLE tblCustomers, tblSuppliers
```

Ngoài ra, bạn cũng có thể dùng *MySQl-Front* để xoá bảng hay các đối tượng *Table* trong cơ sở dữ liệu chỉ định. Nếu chọn nhiều bảng cùng một lúc bạn sử dụng phím *Control* hay *Shift* như sau:



Hình 8-12: Chọn đối tượng để xoá bảng trong MySQL-Front

7. TAO KICH BẢN SQL- SQL SCRIPTS

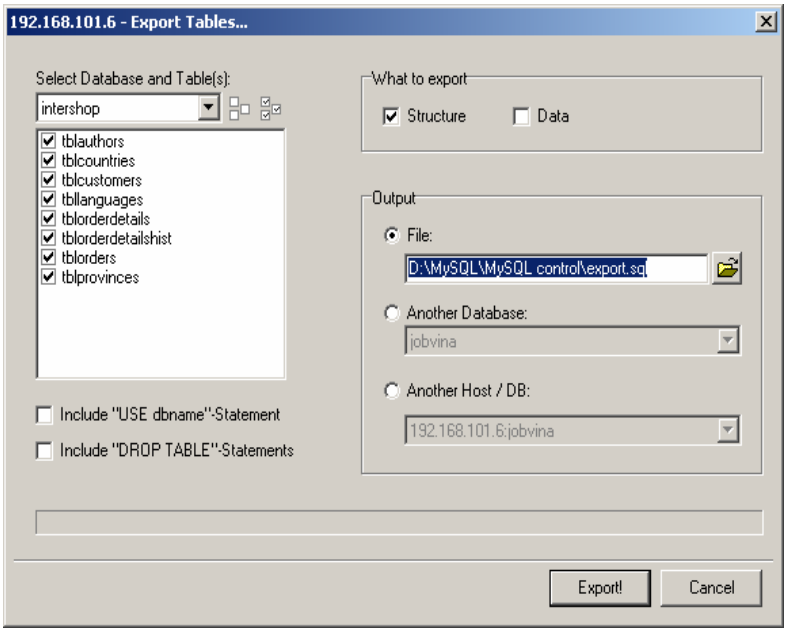
Thông thường khi xây dựng cơ sở dữ liệu để phát triển ứng dụng, đôi khi bạn cần chuyển cơ sở dữ liệu từ máy này sang máy khác, hay từ khu vực này hay đến khu vực khác.

Có rất nhiều cách để làm điều này, ở đây chúng tôi giới thiệu đến các bạn một công cụ tái tạo lại cơ sở dữ liệu mới từ kịch bản của cơ sở dữ liệu gốc.

Kịch bản *SQL* (*SQL Script*) là tổng hợp tất cả các phát biểu *SQL* dùng để tạo ra cơ sở dữ liệu trong quá trình xây dựng chúng, chúng lưu trữ dưới dạng văn bản có tên mở rộng *.sql* (*cautruc.sql*).

Công cụ này tạo kịch bản cho tất cả các đối tượng của cơ sở dữ liệu với những thuộc tính căn bản. Tuy nhiên, nếu bạn chọn vào tuỳ chọn *Data*, *SQL Script* bao gồm các phát biểu *SQL* dạng *Insert* cùng với dữ liệu trong bảng.

Trước tiên bạn có thể nhận thấy cửa sổ công cụ này trong MySQL-Front, bằng cách chọn tên cơ sở dữ liệu *Test*, sau đó chọn *Tools* / *Im-Export* / *Export Table*, cửa sổ xuất hiện như hình 8-13 sau:



Hình 8-13: Tạo kịch bản trong MySQl-Front

KẾT CHƯƠNG

Trong chương này, chúng tôi đã giới thiệu với bạn hầu hết các phát biểu *SQL* thuộc loại định nghĩa cơ sở dữ liệu, thao tác dữ liệu như *Select*, *Insert*, *Delete* và *Update*.

Phát biểu *SQL* dạng *Select* với các mệnh đề như *JOIN* cùng phép toán giữa hai hay nhiều bảng trong phát biểu *SQL* dạng *SELECT*.

Ngoài ra, chúng tôi cũng trình bày hai loại phát biểu *SQL* dạng định nghĩa và thay đổ cơ sở dữ liệu tạo như *CREATE* và *ALTER, DROP*.

Môn học: PHP

BÀI 9: PHP VÀ DATABASE

Để kết nối cơ sở dữ liệu mySQL trong PHP, chúng ta có nhiều cách ứng với nhiều phương thức kết nối cơ sở dữ liệu, trong phần này chúng ta tập trung tìm hiểu cách kết nối cơ sở dữ liệu mySQL từ PHP bằng chính gói của nó.

Những vấn đề chính sẽ được đề cập trong bài học này

- ✓ Khai báo kết nối cơ sở dữ liệu
- ✓ Thêm mẩu tin
- ✓ Cập nhật mẩu tin.
- ✓ Xoá mẩu tin
- ✓ Truy vấn dữ liệu

1. KẾT NỐI CƠ SỞ DỮ LIỆU

Để kết nối cơ sở dữ liệu mySQL bạn sử dụng khai báp như sau:

```
<?php
   $link = mysql_connect ("localhost", "root", "")
   or die ("Could not connect to MySQL Database");
   mysql_select_db("TestDB", $link);
?>
```

Trong đó khai báo sau là kết nối cơ sở dữ liệu mySQL với tên server/ip cùng với username vá password:

```
mysql_connect ("localhost", "root", "")
```

Và mysql_select_db("TestDB", $link); để chọn tên cơ sở dữ liệu sau khi mở kết nối cơ sở dữ liệu, nếu biến $link có giá trị là false thì kết nối cơ sở dữ liệu không thành công.

Sau khi mở kết nối cơ sở dữ liệu mà không sử dụng tìh bạn có thể đóng kết nối cơ sở dữ liệu với cú pháp như sau:

```
mysql_close($link);
```

Chẳng hạn, bạn khai báo trang connection.php để kết nối cơ sở dữ liệu và đóng kết nối ngay sau khi mở thành công.

```
<HTML>
   <HEAD>
   <TITLE>::Welcome to PHP and mySQL</TITLE>
   </HEAD>
   <BODY>
   Mo va dong ket noi CSDL MySQL
   <?php
      $link = mysql_connect ("localhost", "root", "")
      or die ("Could not connect to MySQL Database");
      mysql_select_db("InterShop", $link);
      mysql_close($link);
```

```
?>
</BODY>
</HTML>
```

2. THÊM MẨU TIN

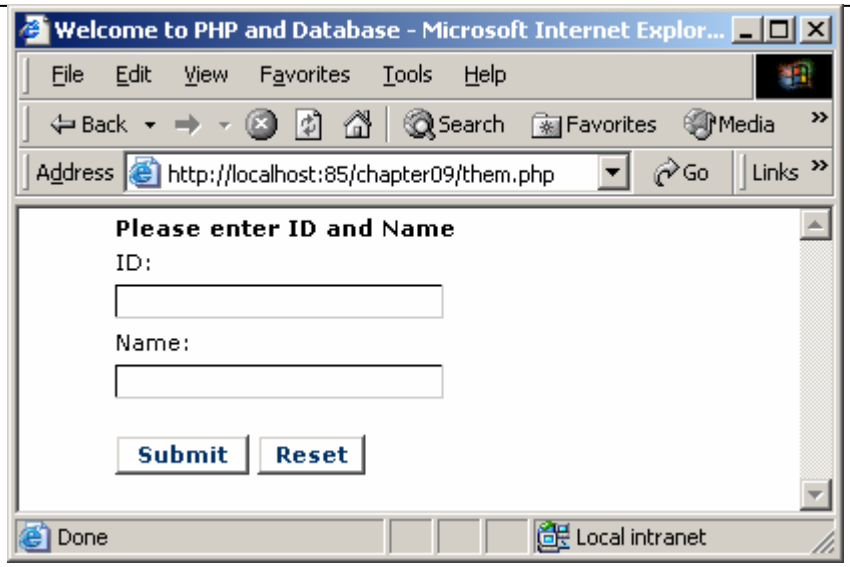
Để thêm mẩu tin, bạn sử dụng hàm mysql_query(chuỗi Insert). Chẳng hạn, chúng ta khai báo trang insert.php để thêm mẩu tin vào bảng tblships có hai cột dữ liệu là ShipID và ShipName như ví dụ trong trang insert.php.

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP and mySQL</TITLE>
</HEAD>
<BODY>
<h3>Them mau tin</h3>
<?php

     require("dbcon.php");
     $sql="insert into tblships values('A01','Testing')";
     $result = mysql_query($sql,$link);
     $affectrow=0;
     if($result)
      $affectrow=mysql_affected_rows();
     mysql_close($link);
?>
So mau tin them vao<?= $affectrow?>
</BODY>
</HTML>
```

Trong đó, bạn sử dụng hàm mysql_query với hai tham số là $sql và $link. Kết quả trả về là số mẩu tin thực thi. Ngoài ra, bạn có thể sử dụng đoạn kết nối cơ sở dữ liệu trong tập tin dbcon.php như ví dụ sau:

```
<?php
   $link = mysql_connect ("localhost", "root", "")
   or die ("Could not connect to MySQL Database");
   mysql_select_db("Test", $link);
?>
```

Trong trường hợp cho phép người sử dụng thêm mẩu tin thì bạn thiết kế form yêu cầu người sử dụng nhập hai giá trị sau đó submit đến trang kế tiếp để thực thi việc thêm gt sau đó submit đến trang kế tiếp để thực thi việc thêm giá trị vừa nhập vào cơ sở dữ liệu như hình 9-1.



Hình 9-1: Thêm mẩu tin

Để làm điều này, trước tiên bạn khai báo trang them.php, trong đó khai báo đoạn javascript để kiểm tra dữ liệu nhập như sau:

```
<SCRIPT language=JavaScript>
     function checkInput()
       {

        if (document.frmPHP.txtID.value=="")
        {
          alert("Invalid ID, Please enter ID");
          document.frmPHP.txtID.focus();
          return false;
        }

        if (document.frmPHP.txtName.value=="")
        {
          alert("Please enter Name");
          document.frmPHP.txtName.focus();
          return false;
        }
        return true;
       }
</script>
```

Kế đến khai báo thể form và hai thẻ input lại text yêu cầu người sử dụng nnập ID và Name như sau:

```
<form name="frmPHP" method="post"
     action="doinsert.php"
     onsubmit="return checkInput();">
  <tr>
   <td align="left" class="content-sm"><b>
    Please enter ID and Name
    </b></td>
  </tr>
  <tr>
   <td align="left" >ID:</td>
  </tr>
  <tr>
   <td align="left">
    <input type="text" name="txtID"
     size="25" maxlength="3" class="textbox">
   </td>
```

```
  </tr>
  <tr>
   <td align="left" >Name:</td>
  </tr>
  <tr>
   <td align="left" >
    <input type="text" name="txtName"
     size="25" maxlength="50" class="textbox">
   </td>
  </tr>
  <tr>
   <td align="left"  valign="top"> <br>
    <input type="submit"
     value="Submit" class="button">
    <input type="reset" value="Reset" class="button">
   </td>
  </tr>
 </form>
```

Lưu ý rằng, bạn khai báo số ký tự lớn nhất cho phép nhập bằng với kích thước đã khai báo trong cơ sở dữ liệu ứng với thuộc tính maxlength.

Khi người sử dụng nhập hai giá trị và nhấn nút submit, trang kế tiếp đực triệu gọi. Trang này lấy giá trị nhập bằng cách sử dụng biến form hay $HTTP_POST_VARS. Đối với trường hợp này chúng ta sử dụng biến form như trang doinsert.php.

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP and mySQL</TITLE>
</HEAD>
<BODY>
<h3>Them mau tin</h3>
<?php
     $affectrow=0;
     require("dbcon.php");
     $sql="insert into tblships(ShipID,ShipName) ";
     $sql .=" values('".$txtID."','".$txtName."')";
     $result = mysql_query($sql,$link);
     if($result)
      $affectrow=mysql_affected_rows();
     mysql_close($link);
?>
So mau tin them vao<?= $affectrow?>
</BODY>
</HTML>
```

3. CẬP NHẬT MẨU TIN

Đối với trường hợp cập nhật mẩu tin, bạn cũng sử dụng hàm mysql_query với phát biểu Update thay ví Insert như trên, ví dụ chúng ta khai báo trang update.php để cập nhật mẩu tin trong bảng tblShips với tên là UpdateTesting khi mã có giá trị là A01.

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP and mySQL</TITLE>
</HEAD>
<BODY>
<h3>Cap nhat mau tin</h3>
<?php

     require("dbcon.php");
     $sql="Update tblships set ShipName='UpdateTesting' ";
     $sql.=" where ShipID='A01'";
     $result = mysql_query($sql,$link);
     $affectrow=0;
     if($result)
```

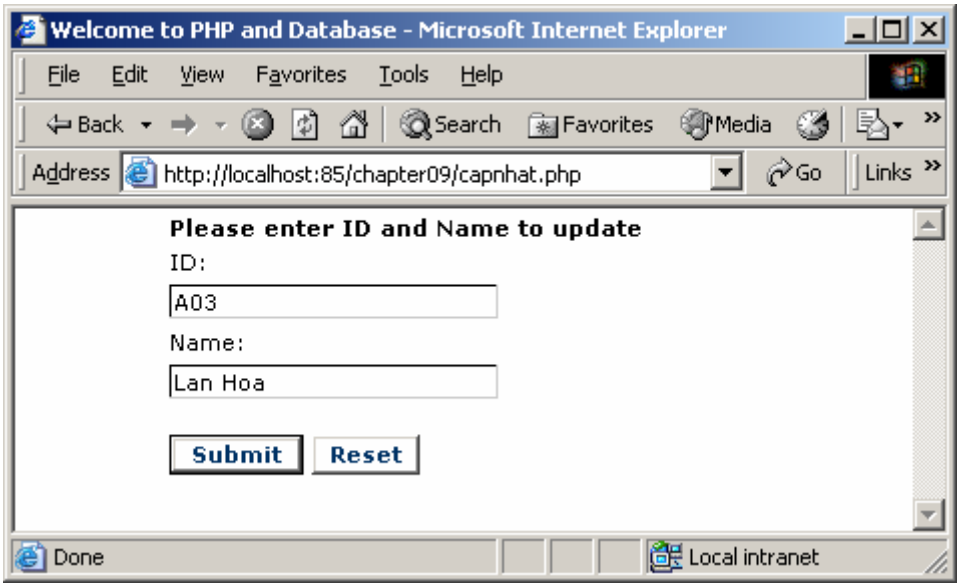
```
        $affectrow=mysql_affected_rows();
      mysql_close($link);
?>
So mau tin cap nhat <?= $affectrow?>
</BODY>
</HTML>
```

Lưu ý rằng, để biết số mẩu tin đã thực thi bởi phát biểu SQL bạn sử dụng hàm mysql_affected_rows.

```
if($result)
            $affectrow=mysql_affected_rows();
```

Tương tự như trên, bạn có thể thiết kế form cho phép người sử dụng cập nhật dữ liệu bằng cách thiết kế form yêu cầu người sử dụng nhập mã và tên cập nhận.

Trước tiên thiết kế from cho phép nhập dữ liệu để cập nhật như ví dụ trang capnhat.php, sau khi học phần truy vấn xong, thaqy vì nhập mã bạn cho phép người sử dụng chọn trong danh sách đã có như hình 9-2.



Hình 9-2: Cập nhật dữ liệu

Sau khi người sử dụng nhấn nút submit, trang doupdate.php sẽ triệu gọi, kết quả trả về 1 hay 0 mẩu tin.

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP and mySQL</TITLE>
</HEAD>
<BODY>
<h3>Cap nhat mau tin</h3>
<?php
      $affectrow=0;
      require("dbcon.php");
      $sql="update tblships set ShipName='";
      $sql .=$txtName."' where ShipID='".$txtID."'";
      $result = mysql_query($sql,$link);
      if($result)
        $affectrow=mysql_affected_rows();
      mysql_close($link);
?>
```

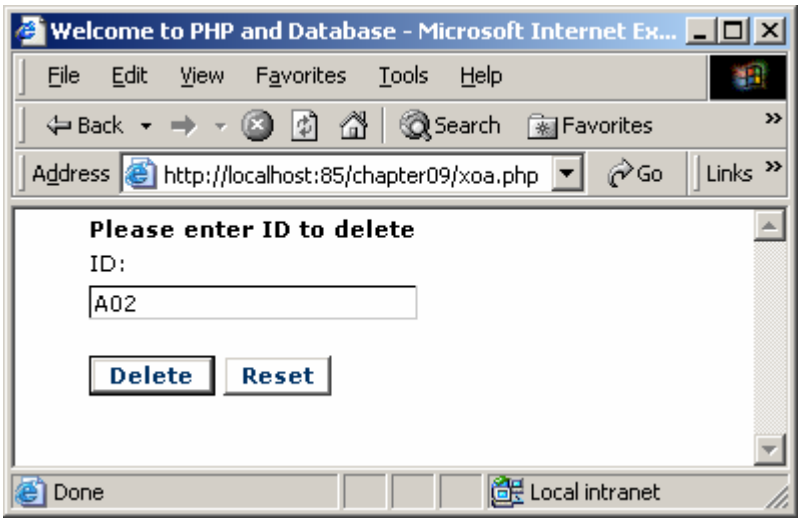
```
So mau tin cap nhat <?= $affectrow?>
</BODY>
</HTML>
```

4. XOÁ MẨU TIN

Tương tự như vậy khi xoá mẩu tin, bạn chỉ thay đổi phát biểu SQL dạng Delete như ví dụ trong tập tin delete.php.

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP and mySQL</TITLE>
</HEAD>
<BODY>
<h3>Xoa mau tin</h3>
<?php

     require("dbcon.php");
     $sql="Delete From tblships where ShipID='A01'";
     $result = mysql_query($sql,$link);
     $affectrow=0;
     if($result)
      $affectrow=mysql_affected_rows();
     mysql_close($link);
?>
So mau tin da xoa <?= $affectrow?>
</BODY>
</HTML>
```

Đối với trường hợp xoá thì đơn giản hơn, bạn cỉh cần biết được mã cần xoá, chính vì vậy trong trường hợp này chúng ta chỉ cần thiết kế trang cho phép nhập mã như hình 9-3.



Hình 9-3: Xoá 1 mẩu tin

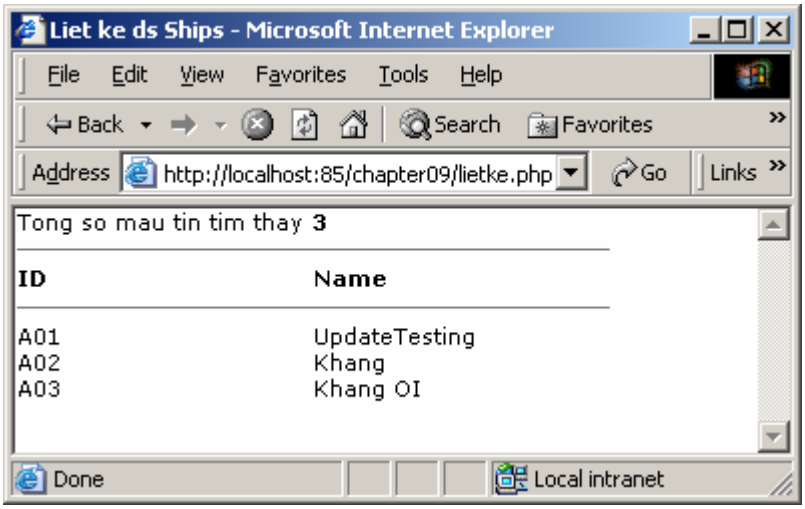
Sau khi nhập mã cần xoá, nếu người sử dụng nhấn nút Delete lập tức trang dodelete.php sẽ triệu gọi và xoá mẩu tin tương ứng.

```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP and mySQL</TITLE>
</HEAD>
```

```
<BODY>
<h3>Xoa mau tin</h3>
<?php
      $affectrow=0;
      require("dbcon.php");
      $sql="delete from tblships ";
      $sql .=" where ShipID='".$txtID."'";
      $result = mysql_query($sql,$link);
      if($result)
       $affectrow=mysql_affected_rows();
      mysql_close($link);
?>
So mau tin xoa <?= $affectrow?>
</BODY>
</HTML>
```

5. TRUY VẤN DỮ LIỆU

Để truy vấn dữ liệu bạn sử dụng hàm mysql_num_rows để biết được số mẩu tin trả về và hàm mysql_fetch_array để đọc từng mẩu tin và mảng sau đó trình bày giá trị từ mảng này. Chẳng hạn, chúng ta tạo một tập tin lietke.php dùng để liệt kê danh sách mẩu tin trong bảng tblShips như hình 9-4.



Hình 9-4: Liệt kê mẩu tin

Để làm điều này, bạn khai báo đoạn chương trình đọc bảng dữ liệu tương tự như ví dụ sau:

```
<?php
      require("dbcon.php");
      $totalRows = 0;
      $stSQL ="select * from tblShips";
      $result = mysql_query($stSQL, $link);
      $totalRows=mysql_num_rows($result);
?>
```

Sau đó, dùng hàm mysql_fetch_array để đọc từng mẩu tin và in ra như sau:

```
<?php
if($totalRows>0)
      {
       $i=0;
       while ($row = mysql_fetch_array ($result))
       {
      $i+=1;
```

```
        ?>
      <tr valign="top">
       <td>
        <?=$row["ShipID"]?> </td>
       <td ><?=$row["ShipName"]?></td>
      </tr>
```

Trong trường hợp số mẩu tin trả về là 0 thì in ra câu thông báo không tìm thấy như sau:

```
<?php
     }
}else{
     ?>
        <tr valign="top">
         <td > </td>
         <td > <b><font face="Arial" color="#FF0000">
          Oop! Ship not found!</font></b></td>
        </tr>
        <?php
     }
?>
```

6. KẾT LUÂN

Trong bài này, chúng ta tập trung tìm hiểu cách kết nối cơ sở dữ liệu, thêm, xoá cập nhật và liệt kê mẩu tin. Trong bài kế tiếp chúng ta tìm hiểu nhiều các trình bày dữ liệu, xoá mẩu tin theo dạng mảng.

Môn học: PHP

BÀI 10: XOÁ, CẬP NHẬT DỮ LIỆU DẠNG MẢNG

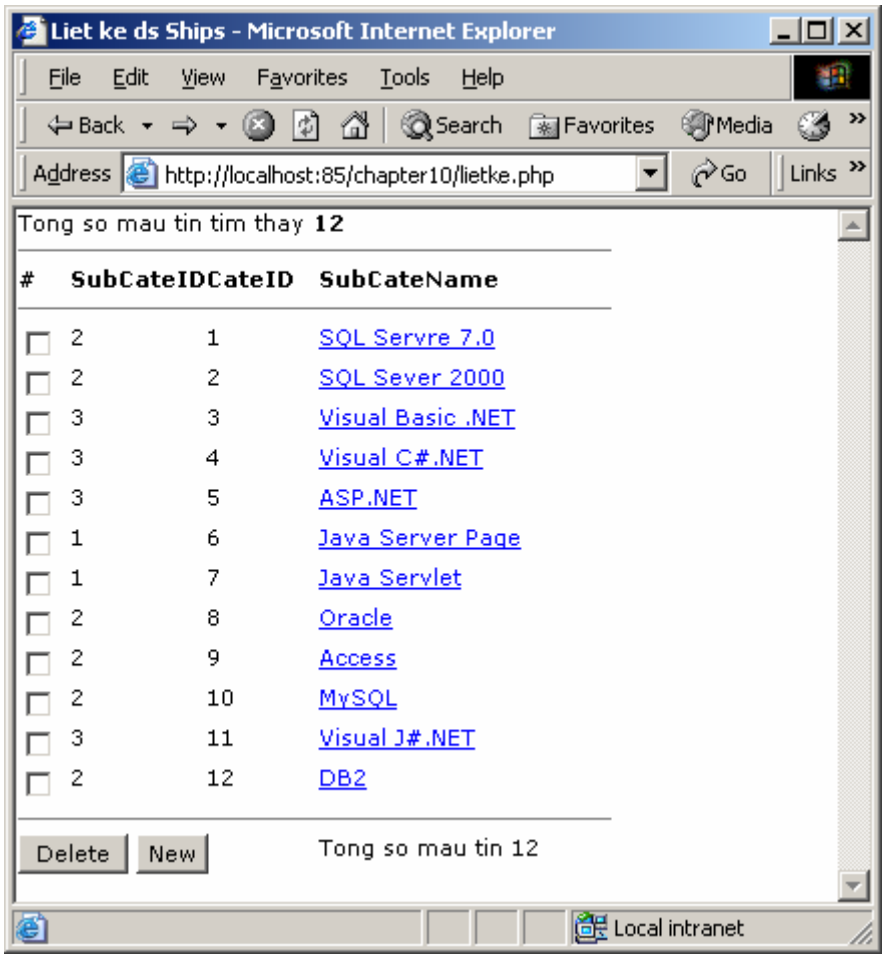
Trong bài trước chúng ta đã làm quen với cách xoá mẩu tin trong cơ sở dữ liệu mySQL. Đối với trường hợp xoá một lúc nhiều mẩu tin, chúng ta phải xây dựng trang PHP có sử dụng thẻ input dạng checkbox.

Những vấn đề chính sẽ được đề cập trong bài học này

- ✓ Liệt kê dữ liệu dạng danh sách
- ✓ Xoá nhiều mẩu tin
- ✓ Cập nhật nhiều mẩu tin

1. LIỆT KÊ DỮ LIỆU

Để xoá nhiều mẩu tin cùng một lúc, trước tiên bạn khai báo trang PHP để liệt kê danh sách mẩu tin trong mảng dữ liệu chẳng hạn, mỗi mẩu tin xuất hiện một checkbox tương ứng. Checkbox này có giá trị là mã nhận dạng của mẩu tin đó. Trong trường hợp này chúng ta dùng cột khoá của mã chuyển hàng (SubCateID) trong bảng tblSubCategories định nghĩa trong trang lietke.php như hình 10-1.



Hình 10-1: Liệt kê danh sách lại sản phẩm

Để cho phép lấy được nhiều giá trị chọn của sản phẩm như hình tren, bạn khai báo các checkbox này cùng tên (giả sử tên là chkid) và giá trị trị là SubCateID của mỗi sản phẩm như ví dụ 10-1 trong trang lietke.php.

```
<?php
  if($totalRows>0)
    {
     $i=0;
      while ($row = mysql_fetch_array ($result))
      {
    $i+=1;
     ?>
  <tr valign="top">
  <td><input type=checkbox name=chkid
    value="<?=$row["SubCateID"]?>"> </td>
  <td><?=$row["CateID"]?> </td>
        <td><?=$row["SubCateID"]?> </td>
        <td ><a href="capnhat.php?id=<?=$row["SubCateID"]?>">
    <?=$row["SubCateName"]?></a></td>

       </tr>
       <?php
    }
  ?>
  <tr valign="top">
        <td colspan="4" align="middle">
            <hr noshade size="1">
     </td>
  </tr>
  <tr valign="top">
     <td colspan=3><input type=submit value="Delete">
    <input type=hidden name=from_ value="subcategories">
    <input type=hidden name=type value="0">
    <input type=hidden name=chon value="">
    <input type=button value="New"
    onclick="window.open('them.php',target='_main')"></td>
        <td >Tong so mau tin <?=$i?></td>
       </tr>
  <?php
  }else{
  ?>
       <tr valign="top">
        <td > </td><td > </td><td > </td>
        <td > <b><font face="Arial" color="#FF0000">
         Oop! Ship not found!</font></b></td>
       </tr>
       <?php
   }
    ?>
```

Trong đó, hai khai báo sau:

```
<input type=hidden name=from_ value="subcategories">
<input type=hidden name=type value="0">
<input type=hidden name=chon value="">
```

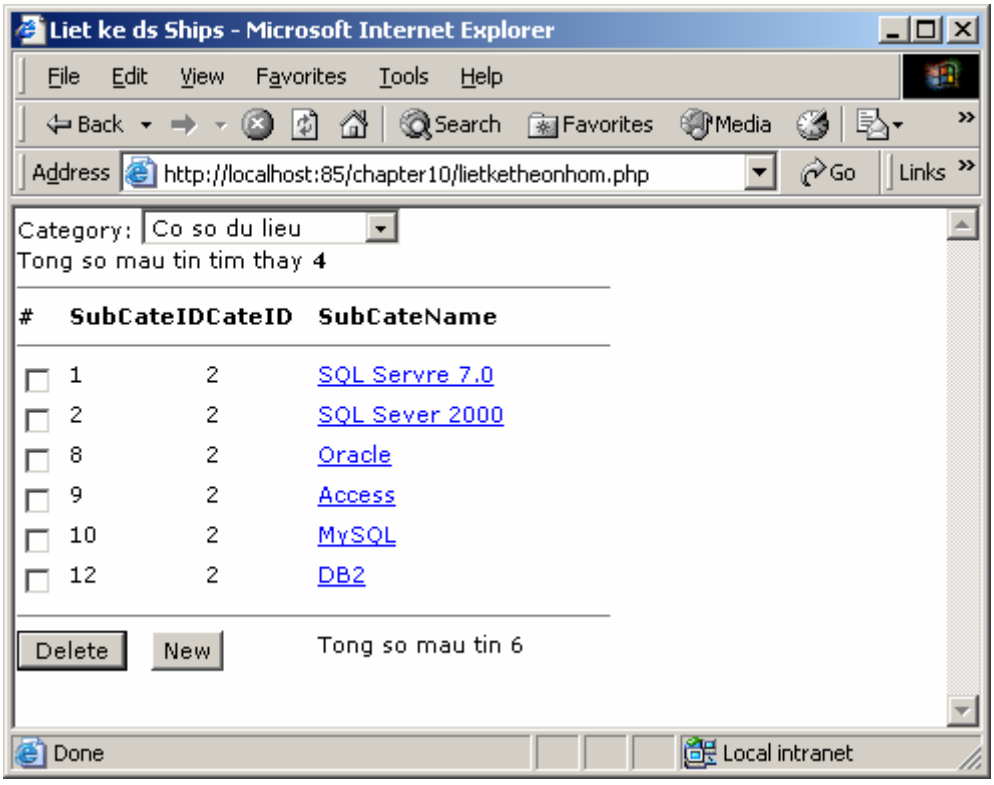
Cho biết bạn submit từ trang nào và loại xoá nhiều mẩu tin hay một mẩu tin đối với bảng tương ứng. Mục đích của vấn đề này là trang delete sử dụng chung cho nhiều bảng khác nhau và từ trang liệt kê (xoá nhiều) hoặc từ trang edit (1 mẩu tin cụ thể).

Ngoài ra, chúng ta khai báo <input type=hidden name=chon value=""> để nhận giá trị chọn trên cách checkbox bằng cách khai báo đoạn javascript như sau:

```
<script>
     function calculatechon()
```

```
        {
          var strchon="";
          var alen=document.frmList.elements.length;
          var buttons=1;

          alen=(alen>buttons)?document.frmList.chkid.length:0;
          if (alen>0)
          {
            for(var i=0;i<alen;i++)
            if(document.frmList.chkid[i].checked==true)
                    strchon+=document.frmList.chkid[i].value+",";
          }else
          {
            if(document.frmList.chkid.checked==true)
                    strchon=document.frmList.chkid.value;
          }
          document.frmList.chon.value=strchon;
          return isok();
        }

</script>
```

Tuy nhiên, do nhiều loại sản phẩm thuộc các nhóm sản phẩm khác nhau, chính vì vậy bạn khai báo danh sách nhóm sản phẩm trên thẻ select cho phép người sử dụng liệt kê sách theo nhóm sản phẩm như hình 10-2.



Hình 10-2: Liệt kê danh sách loại sách

Để liệt kê danh sách nhóm trong bảng tblCategories, bằng cách khai báo phương thức nhận chuỗi SQL dạng Select và giá trị mặc định trả về nhiều phần tử thẻ option trong tập tin database.php như ví dụ 10-2.

```
function optionselected($stSQL,$item,$links)
{
     $results = mysql_query($stSQL, $links);
        $totalRows=mysql_num_rows($results);
     $strOption="<option value=\"\" selected>";
     $strOption .="--Select--</option>";
     if($totalRows>0)
     {
      while ($row = mysql_fetch_array ($results))
       {
        $strOption .="<option value=\"" ;
        $strOption .=$row["ID"]."\"";
        if($row["ID"]==$item)
       $strOption .=" selected ";
        $strOption .= ">".$row["Name"];
        $strOption .="</option>";
       }
     }
     return $strOption;
}
```

Sau đó, gọi phương thức này trong trang lietketheonhom.php như ví dụ 10-3.

```
<?php
  require("dbcon.php");
  require("database.php");
  $id="";
  if (isset($cateid))
    $id=$cateid;
  $stSQL ="select CateID As ID, CateName as Name from tblCategories ";
  $result = mysql_query($stSQL, $link);
  $totalRows=mysql_num_rows($result);
  $strOption=optionselected($stSQL,$id,$link);
?>
<form name=frmMain method=post>
  <tr>
       <td align=left colspan=4>
    Category: <select name=cateid onchange="document.frmMain.submit();">

    <?=$strOption?>
    </select></td>
  <td align=right> </td>
  </tr>
</form>
```

Lần đầu tiên bạn có thể chọn mặc định một nhóm hoặc liệt kê tất cả, khi người sử dụng chọn nhóm sản phẩm nào đó thì trang lietketheonhom.php sẽ liệt kê danh sách loại sách của nhóm sách đó. Để làm điều này, bạn khai báo thẻ form với thẻ select như ví dụ 10-4.

```
<form name=frmMain method=post>
  <tr>
       <td align=left colspan=4>
    Category: <select name=cateid onchange="document.frmMain.submit();">

    <?=$strOption?>
    </select></td>
  <td align=right> </td>
  </tr>
</form>
```

Khi người sử dụng chọn các mẩu tin như hình 10-2 và nhấn nút Delete, dựa vào giá trị của nút có tên action (trong trường hợp này là Delete), bạn có thể khai báo biến để lấy giá trị chọn bằng cách khai báo như ví dụ 10-5.

```
$strid=$chon;
```

```
$strid=str_replace(",","','",$strid);
```

Dựa vào thẻ hidden khai báo trong các trang trình bày danh sách (chẳng ạn lietketheonhom.php) mẩu tin như sau:

```
<input name="from" type=hidden value="subcategories">
```

Bạn có thể biết từ trang nào gọi đến trang dodelete.php để quay trở về khi thực hiện xong tác vụ xử lý.

Ngoài ra, dựa vào giá trị của nút action để thực hiện phát biểu SQL. Chẳng hạn, trong trường hợp này nếu người sử dụng nhấn hút Delete thì bạn khai báo như ví dụ 10-6 sau:

```
switch($strfrom)
   {
   case "subcategories":
      $stSQL ="delete from tblsubcategories where SubCateID in('".$strid."')";
      $strlocation="Location:lietketheonhom.php";
      break;
   case "categories":
      $stSQL ="delete from tblcategories where CateID in('".$strid."')";
      $strlocation="Location:nhom.php";
      break;
   }
```

Sau đó, bạn có thể thực thi phát biểu SQL vừa khai báo ở trên như ví dụ 10-7.

```
if($stSQL!="")
{
   $result = mysql_query($stSQL, $link);
}
```

Lưu ý rằng, bạn cũng nên khai báo try catch trong khi làm việc với cơ sở dữ liệu. Ngoài ra, bạn cũng phải xác nhận trước khi thực thi hành động xoá mẩu tin chọn bằng cách khai báo đoạn Javascript như sau:

```
<script>
function isok()
 {
  return confirm('Are you sure to delete?');
 }
</script>
```

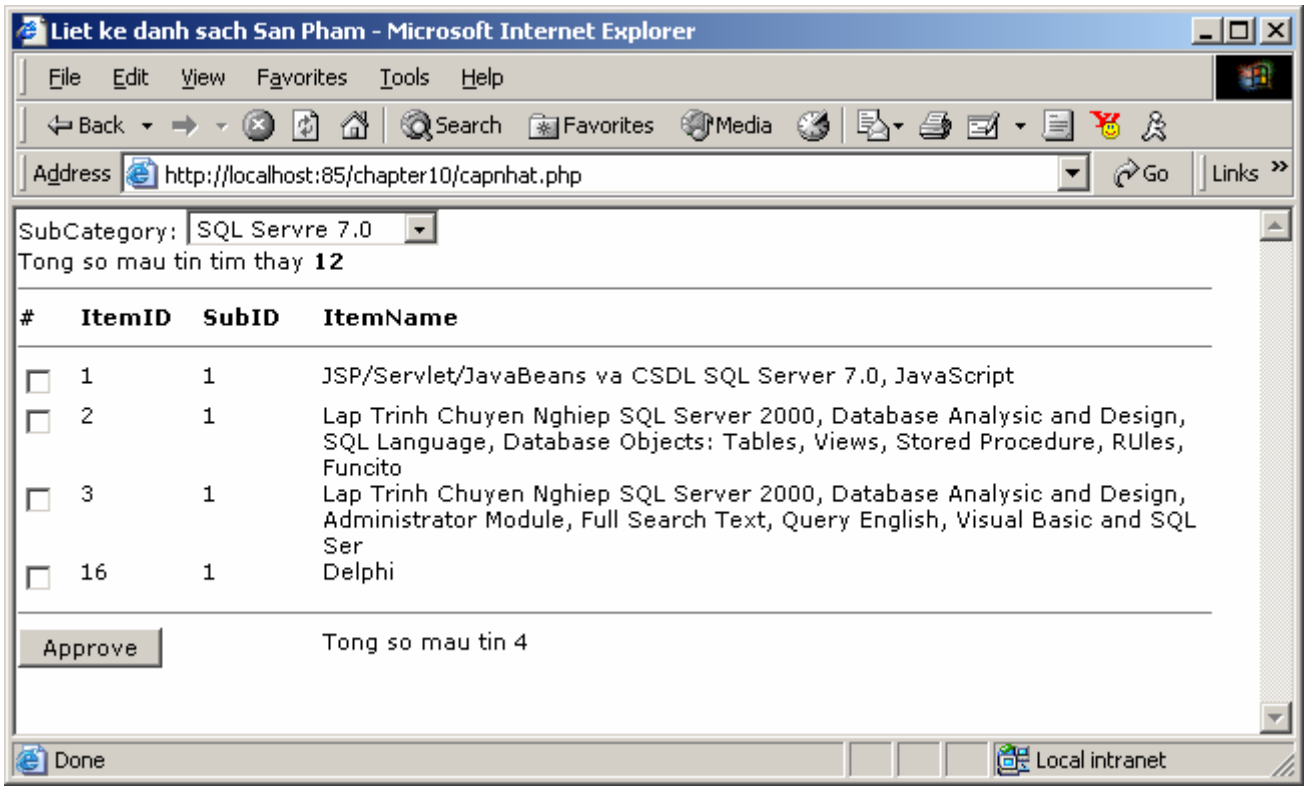
Sau đó gọi trong biến cố onsubmit của form như sau:

```
<form action=dosql.php method=post onsubmit="return calculatechon();">
```

2. CẬP NHẬT NHIỀU MẨU TIN

Tương tự như trường hợp Delete, khi bạn duyệt (approval) một số mẩu tin theo một cột dữ liệu nào đó, chẳng hạn, trong trường hợp này chúng ta cho phép sử dụng những sản phẩm đã qua sự đồng ý của nhà quản lý thì cột dữ liệu Activate của bảng tbltems có giá trị là 1.

Để làm điều này, trước tiên bạn liệt kê danh sách sản phẩm như hình 10-3.



Hình 10-3: Liệt kê danh sách sản phẩm duyệt hay chưa

Tương tự như trong trường hợp delete, bạn khai báo trang doUpdate như sau:

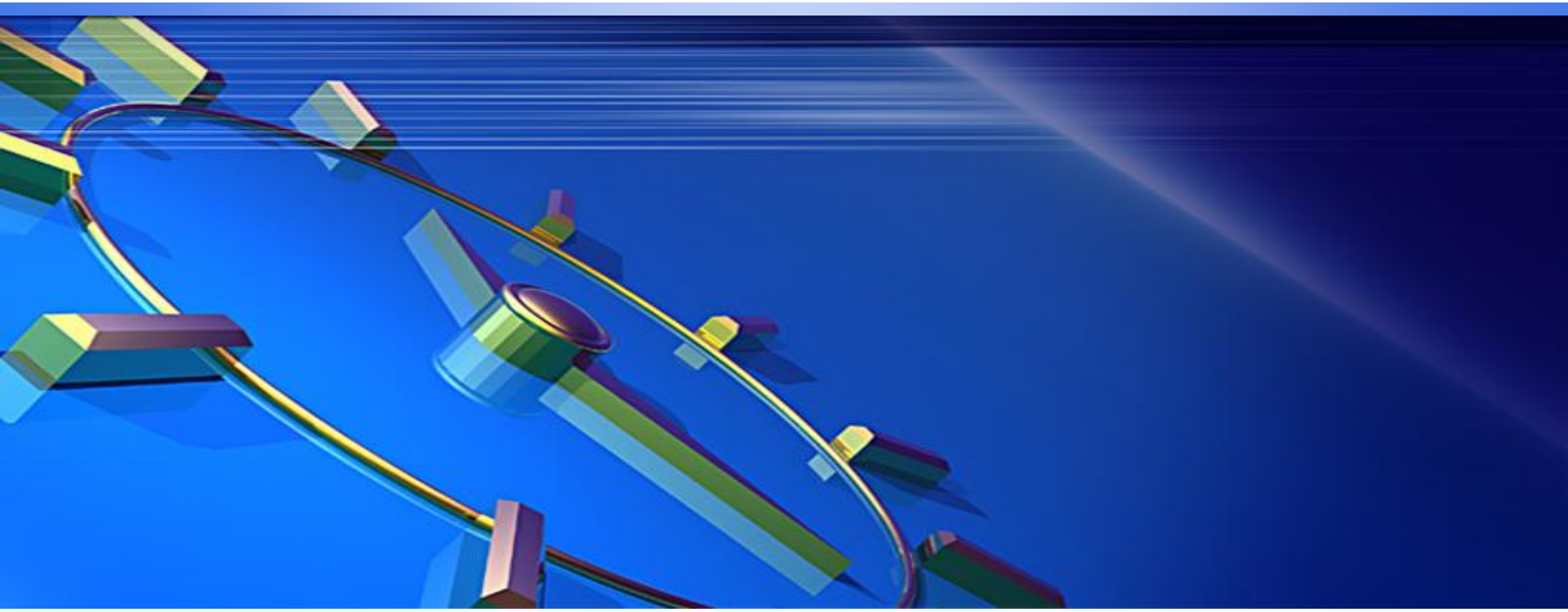
```
<HTML>
<HEAD>
<TITLE>::Welcome to PHP and mySQL</TITLE>
</HEAD>
<BODY>
<h3>Cap nhat mau tin</h3>
<?php
require("dbcon.php");
$strid=$chon;
$strid=str_replace(",","','",$strid);
$strfrom="";
if(isset($from_))
     {
        $strfrom=$HTTP_POST_VARS{"from_"};
     }
$strtype="";
if(isset($type))
     {
        $strtype=$HTTP_POST_VARS{"type"};
     }

$stSQL="";
if($strfrom<>"")
{
     switch($strfrom)
        {
        case "items":
           $stSQL ="update tblItems set Activate=1 where ItemID
in('".$strid."')";
```

```
            break;
          }
        if($stSQL!="")
        {
          $result = mysql_query($stSQL, $link);
          if($result)
             $affectrow=mysql_affected_rows();
          mysql_close($link);
        }
}
?>
So mau tin cap nhat <?= $affectrow?>
</BODY>
</HTML>
```

3. KẾT LUẬN

Trong bài này, chúng ta tìm hiểu chức năng xoá, cập nhật nhiều mẩu tin bằng cách sử dụng thẻ input loại checkbox cùng tên và khác giá trị, bài kế tiếp chúng ta tiếp tục tìm hiểu về chức năng đăng nhập trong PHP.

LẬP TRÌNH WEB



Giáo viên: Trần Cao Nhân

Ngôn ngữ lập trình PHP

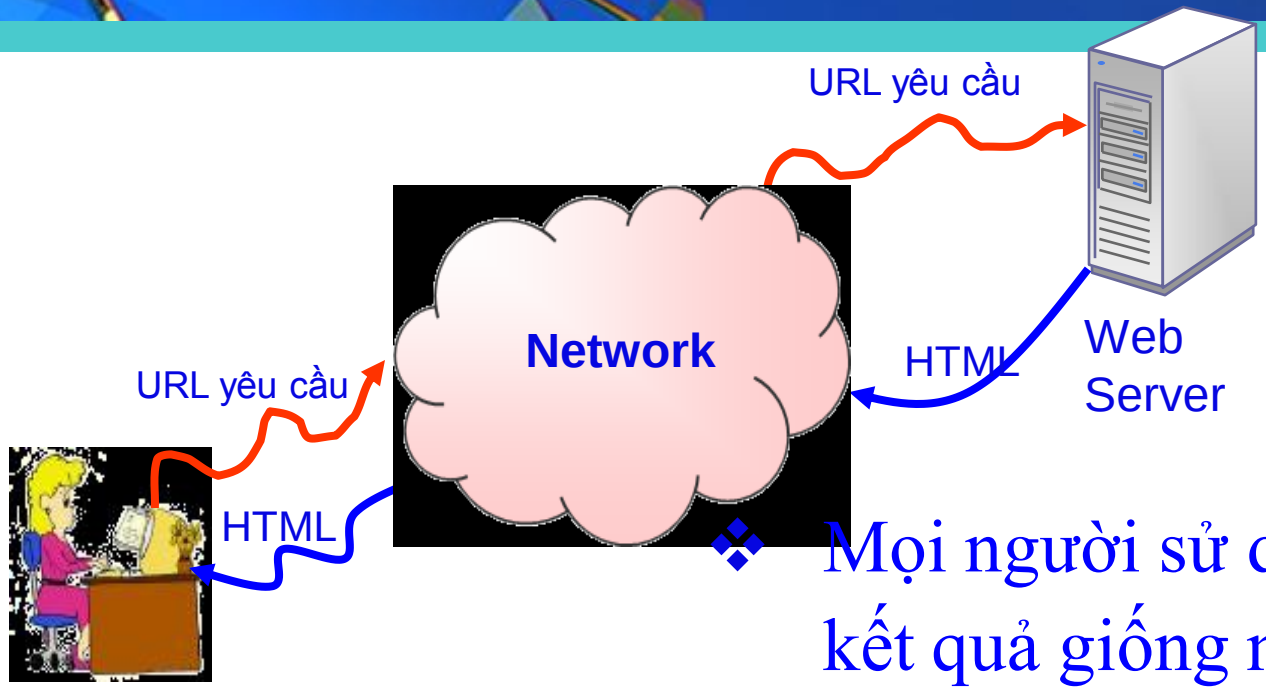
2 Ngôn ngữ Web lập trình PHP

Một số khái niệm

Web tĩnh là gì?

- Là những trang văn bản bao gồm chữ viết, hình ảnh, video …
- Tốc độ nhanh, dung lượng nhỏ, không kết nối CSDL.
- Thay đổi thông tin rất khó, số trang nhiều.

Cách làm việc của web tĩnh

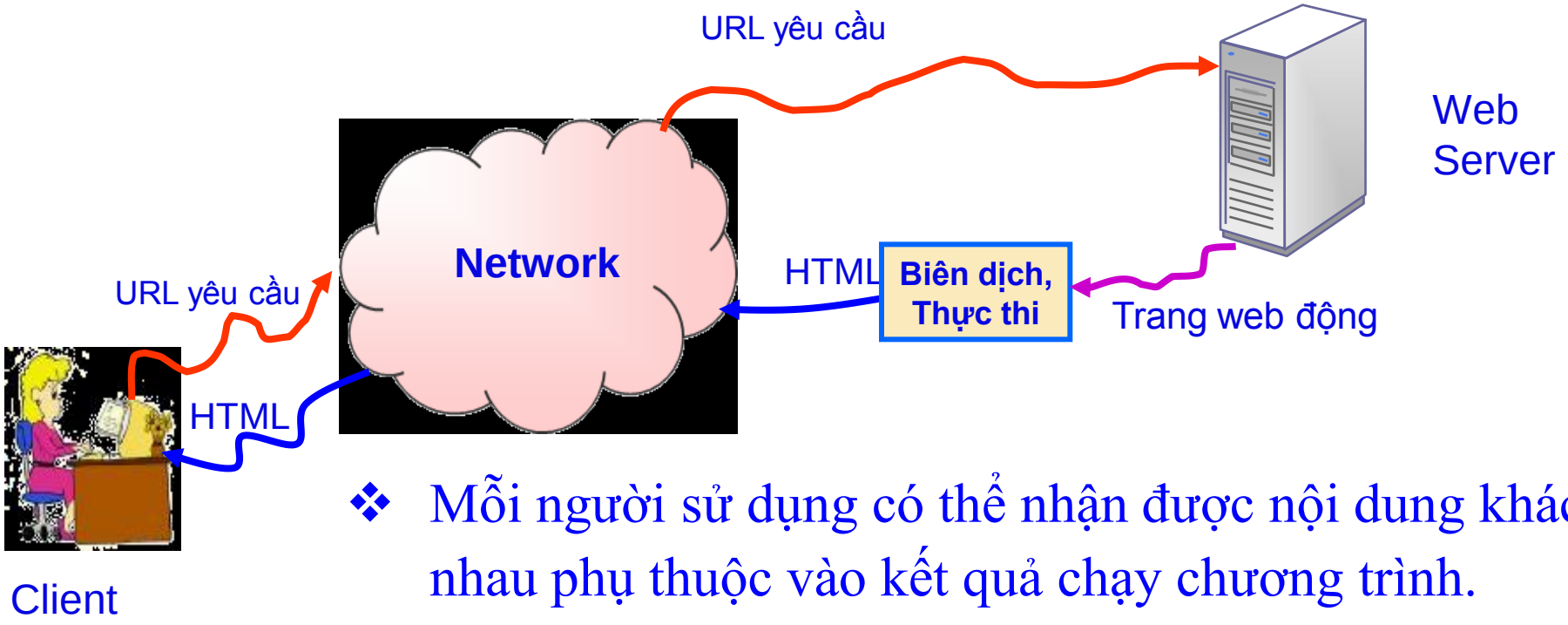


- Mọi người sử dụng nhận được kết quả giống nhau.
- Trang web được viết bằng HTML, chỉ thay đổi khi có sự thay đổi của người xây dựng
- Khả năng tương tác yếu
- Webserver hoạt động giống 1 file server.

Web động là gì?

- Là một trang mạnh và linh hoạt vì:
  - Cho phép người dùng quản lý nội dung.
  - Có bộ nhớ, cho phép người dùng đăng ký và đăng nhập.
  - Dễ duy trì, cập nhật và phát triển.
  - Nội dung được lưu trữ trong CSDL.
  - Thể hiện được tính thương mại điện tử.

Cách làm việc của web động



- Mỗi người sử dụng có thể nhận được nội dung khác nhau phụ thuộc vào kết quả chạy chương trình.
- Trang web viết bằng HTML + Ngôn ngữ lập trình phía server. Có thể được thay đổi bởi người sử dụng.
- Khả năng tương tác mạnh.

PHP là gì?

- PHP = PHP: Hypertext Preprocessor, tên gốc là Personal Home Pages.
- PHP là ngôn ngữ lập trình các ứng dụng web.
- Bộ biên dịch PHP là phần mềm mã nguồn mở.
- Thường kết nối với hệ quản trị CSDL MySQL
- PHP là ngôn ngữ nhúng.

Nhúng mã lệnh PHP vào HTML

- Có thể nhúng mã PHP vào mọi vị trí trong tài liệu HTML.
- Đoạn mã PHP được đặt giữa cặp thẻ:

```
<?php
// Mã lệnh PHP
?>
```

Đặc điểm của PHP

- Có khản năng đối tượng.
- Thông dịch.
- Phân biệt **CHỮ HOA** và **chữ thường**.
- Lệnh kết thúc bằng dấu chấm phẩy " ; "

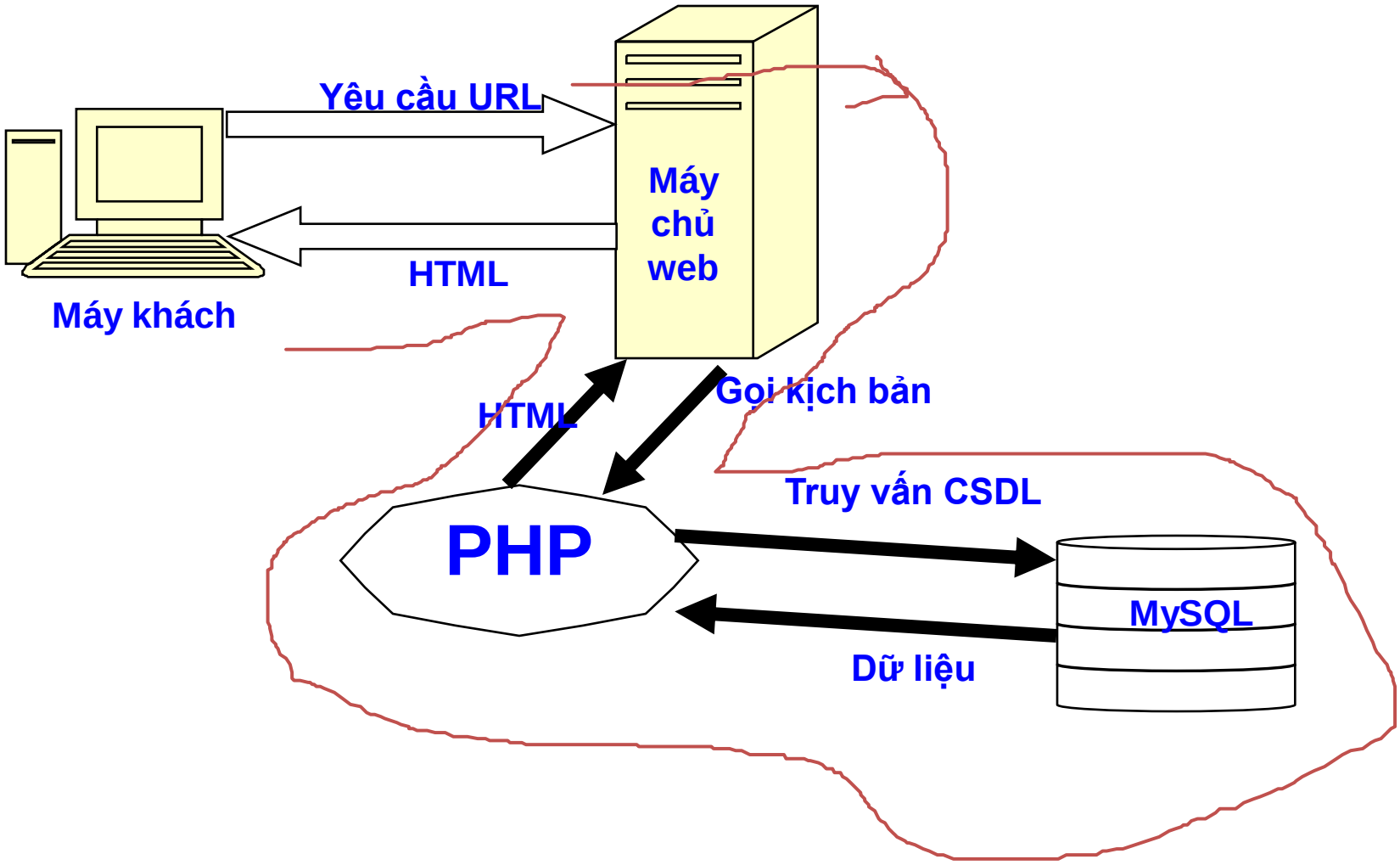
Tạo sao cần dùng PHP

- PHP dễ học, dễ viết.
- Có khả năng truy xuất hầu hết CSDL có sẵn.
- Thể hiện được tính bền vững, chặn chẽ, phát triển không giới hạn.
- PHP miễn phí, mã nguồn mở.
- Điểm mạnh của PHP và MySQL trên hệ điều hành Linux.

MySQL là gì?

- MySQL là hệ quản trị cơ sở dữ liệu mã nguồn mở tốt nhất và phổ biến hiện nay.
- Có khả năng thực thi hoàn hảo, linh động và đang tin cậy.
- Dễ nắm bắt, giá rẽ hoặc miễn phí.
- MySQL là một ứng dụng mã nguồn mở.
- MySQL xử lý 6000 bảng và 5 tỉ mẫu tin

Cơ chế làm việc PHP & MySQL



Ngôn ngữ lập trình PHP

Cú pháp PHP

Cú pháp mã lệnh PHP có 4 dạng sau:

Dạng 1: Dạng chuẩn

<?php Mã lệnh PHP **?>**

Dạng 2: Dạng ngắn gọn

<? Mã lệnh PHP **?>**

Nhắc nhở: Nên dung dạng chuẩn.

Cú pháp PHP

Dạng 3: Cú pháp giống với ASP.

<% Mã lệnh PHP **%>**

Dạng 4: Cú pháp bắt đầu bằng script

<script language=php>

.....Mã lệnh PHP

</script>

Xuất giá trị ra trình duyệt web

- **Hàm echo(<thông tin>);**

  Trong đó: thông tin có thể là hằng, biến, biểu thức hay hàm.

  Có thể sử dụng **echo** <thông tin>;

  Ví dụ:

```
echo("Chào các bạn");
echo "Chào các bạn";
```

Xuất giá trị ra trình duyệt web

- **Hàm print(<thông tin>); giống echo**

Ví dụ:

```
<?php
print("Chào các bạn");
print "Chao các bạn";
?>
```

Viết ghi chú trong PHP

Để ghi chú trong PHP có 3 dạng sau:

Dạng 1: # *đây là ghi chú.*

Dạng này chỉ áp dụng ghi đó chỉ nằm trên một dòng văn bản

Dạng 2: // *đây là ghi chú.*

Dạng này cũng chỉ áp dụng ghi đó chỉ nằm trên một dòng văn bản

Dạng 3: /* *đây là một ghi chú dài*

*Áp dụng cho nhiều hàng */

Khái niệm biến

- Biến là một ô nhớ trong bộ nhớ chính dùng để lưu trữ giá trị
- Biến trong PHP bắt đầu bằng dấu $ theo sau là tên biến, biến có thể được khai báo khi dùng.

Quy tắc đặt tên biến

- Biến trong PHP phân biệt CHỮ HOA, chữ thường
- Tên biến bao gồm chữ cái, chữ số, dấu gạch nối (_) và phải bắt đầu bằng chữ cái hoặc dấu gạch nối.
- Biến không cần khai báo (được tự động khai báo vào lần gán giá trị đầu tiên).
- Việc sử dụng biến chưa khởi tạo sẽ gây lỗi.

Ví dụ: Khai báo và khởi gán cho tên biến.

```
<?php
    $var_number = 6;
    $var_string = "Hello world";
    $_POST["name"]="John";
    $var_arr = array("Hồ", "Diên", "Lợi");
?>
```

Khởi gán giá trị cho biến

Để khởi gán giá trị cho biết thực hiện như sau:

$name_var = <giá trị biến>;

$name_var = <biểu thức>;

Ví dụ:

```
<?php
    $soluong = 20;
    $gia = 25000;
    $t_tien = $gia * $soluong;
?>
```

Phạm vi hoạt động của biến

❖ Biến cục bộ

- Biến cục bộ là biến chỉ có ý nghĩa khi được sử dụng trong phạm vị xác định.
- Giả sử biến trong hàm là biến cục bộ, nó chỉ có ý nghĩa trong hàm.

Ví dụ biến cục bộ

```
<?php
    $a= 10; //biến toàn cục
    function test() {
        echo $a; // biến cục bộ
    }
    test(); // không cho kết quả
    echo $a; // kết quả 10
?>
```

Phạm vi hoạt động của biến

❖ **Biến toàn cục**

- Biến cục bộ là biến luôn có ý nghĩa khi sử dụng.
- Khi khai báo biến toàn cục ta sử dụng từ khóa **global**

❖ **Biến static**

- Biến static không mất giá trị khi ra khỏi phạm vi xác định, vẫn giữ giá trị khi gọi lại biến.

Ví dụ biến toàn cục

```
<?php
    $a = 10; $b = 20;
    function sum() {
        global $a, $b;
        $b = $a+$b;
    }
    sum();
    echo $b; // kết quả 30
?>
```

Ví dụ biến static

```
<?php
    function hien() {
        static $a =0;
        echo $a."</br>";
        $a++;
    }
    hien(); // kết quả 0
    hien(); // kết quả 1
    hien(); // kết quả 2
?>
```

Khái niệm hằng

- Hằng là một giá trị không đổi trong quá trình xử lý.
- Quy tắc đặt tên hằng giống quy tắc đặt tên biến, chỉ không sử dụng dấu $
- Một số hằng định nghĩa sẵn: FALSE, TRUE…

Định nghĩa và sử dụng hàm

❖ Hằng được định nghĩa bằng hàm:

define(<tên_hằng>,<giá_trị>);

Ví dụ:

```
define("PI",3.1423);
```

❖ Sử dụng hằng

```
<?php
    define("PI",3.1423);
   $r = 10;
   echo "Diện tích". 2*PI*$r*$r;
?>
```

Kiểu dữ liệu trong PHP

- Kiểu số(số nguyên, số thập phân).
- Kiểu chuỗi
- Kiểu lôgic
- Kiểu mảng
- Kiểu đối tượng

Ép kiểu và kiểm tra kiểu

- **Sử dụng cú pháp tương tự trong C**

```
$x = "123abc"; //$x là chuỗi
$x =(int) "123abc";
//$x là số nguyên=123
```

Ép kiểu và kiểm tra kiểu

Ký hiệu	Ý nghĩa kiểu
(int), (integer)	Số nguyên
(real), (double), (float)	Số thập phân
(string)	Chuỗi
(array)	Mảng
(object)	Đối tượng
(bool), (boolean)	Logic
(unset)	NULL, tương tự như gọi unset()

Toán tử toán học

Phép toán	Ví dụ	Kết quả
+	$y =2; $x=$y+2;	4
-	$y =2; $x=5-$y;	3
*	$y =2; $x=$y*5;	10
/	$y =5; $x=$y/2;	2.5
%	$y =5%3; $x =16%8;	2 0
++	$y =5; $x++;	x=6
--	$y =5; $x--;	x=4

Toán tử gán

Phép toán	Ví dụ	Tương đương
=	$x=$y;	$x=$y;
+=	$x+=$y;	$x=$x+$y;
-=	$x-=$y;	$x=$x-$y;
*=	$x*=$y;	$x=$x*$y;
/=	$x/=$y;	$x=$x/$y;
.=	$x.=$y;	$x=$x.$y;
%=	$x%=$y;	$x=$x%$y;

Toán tử so sánh

Phép toán	Ví dụ
==	5==8 kết quả false
!=	5!=8 kết quả true
<>	5<>8 kết quả true
>	5>8 kết quả false
<	5<8 kết quả true
>=	5>=8 kết quả false
<=	5<=8 kết quả true

Toán tử logic

Phép toán	Ví dụ
&&	$x=6; $y=3; ($x < 10 && $y > 1) kết quả true
||	$x=6; $y=3; ($x==5 || $y==5) kết quả false
!	$x=6; $y=3; !($x==$y) kết quả true

Toán tử @

Không phát sinh lỗi khi sử dụng toán tử @

Ví dụ:

```
<?php
    $a = 10;
    $b = 0;
    $c1 =$a/$b;
    echo  "Kết quả :".$c1; //thông báo lỗi
    $c2 = @($a/$b);
    echo  "Kết quả :".$c1
?>
```

Sự khác nhau giữa " " và ' '

Ví dụ: Sự khác nhau giữa "..." và '...'

```
<?php
    $a = 3;
    $b = 4;
    $c = $a+$b;
    echo "Tổng $a + $b =$c"; // Tổng 3 + 4 =7
    echo 'Tổng $a + $b =$c'; // 'Tổng $a + $b =$c
?>
```

Tham chiếu

- Trong PHP tham chiếu có nghĩa là lấy cùng giá một trị bằng nhiều tên biến khác nhau.
- Ký hiệu tham chiếu là &.

Ví dụ:

```
<?php
    $a = 10;
    $b = &$a;
    echo  "Kết quả :".$b;
?>
```

Các hàm kiểm tra giá trị

Hàm isset()

- Hàm isset() dùng để kiểm tra xem biến có giá trị hay không.

Ví dụ:

```
<?php
    $a = "Nguyễn Hoàng Nam";
    $b ="";
    isset($a); // trả về giá trị true
    isset($b); // trả về giá trị false
?>
```

Hàm empty()

- Hàm empty() dùng để kiểm tra biến có giá trị rỗng hay không, nếu biến có giá trị NULL.

Ví dụ:

```
<?php
    $a = "Nguyễn Nam";
    $b ="";
    empty($a); // trả về giá trị false
    empty($b); // trả về giá trị true
?>
```

Hàm is_numeric()

- Hàm is_numeric() kiểm tra biến có kiểu giá trị kiểu số hay không.

Ví dụ:

```
<?php
    $a = "Nguyễn Hoàng Nam";
    $b =123;
    is_numeric($a); // trả về giá trị false
    is_numeric($b); // trả về giá trị true
?>
```

Hàm is_int()

- Hàm is_int(<tên_biến>) hoặc is_long(<tên_biến>) kiểm tra giá trị của biến có phải là số nguyên hay không.

Ví dụ:

```
<?php
    $a = 124.5;
    $b =123;
    is_int($a); // trả về giá trị false
    is_int($b); // trả về giá trị true
?>
```

Hàm is_string()

- Hàm **is_string()** kiểm tra giá trị của biến có phải là kiểu chuỗi hay không

Ví dụ:

```
<?php
    $a = 124.5;
    $b ="Nguyễn Hoàng Nam";
    is_string($a); // trả về giá trị false
    is_string($b); // trả về giá trị true
?>
```

Hàm is_double()

- Hàm **is_double()** kiểm tra giá trị của biến có phải là kiểu số có dấu chấm động

Ví dụ:

```
<?php
    $a = 12345;
    $b =123.45;
    is_double($a); // trả về giá trị false
    is_double($b); // trả về giá trị true
?>
```

Hàm gettype()

- Hàm **gettype()** kiểm tra kiểu dữ liệu của biến, hoặc giá trị là kiểu nào: integer, string, double, array, object, class…

Ví dụ:

```
<?php
    $a = 12345;
    $b = "Nguyễn Hoàng Nam";
    gettype($a); // trả về numeric
    gettypy($b); // trả về string
?>
```

Câu lệnh điều khiển

Câu lệnh if…

```
if (<BTĐK>)
{
    // mã lệnh
}
```

Nếu <BTĐK> đúng thì thực hiện <mã lệnh>, ngược lại thì thực hiện công việc sau đó

Câu lệnh if… else

```
if (<BTĐK>)
    {
        // Thực hiện công việc nếu BTĐK đúng
    }
else
    {
        // Thực hiện công việc nếu BTĐK sai
    }
```

Toán tử ?

<tên_biến> = (<btdk>)? <value_true>: <value_false>

Ví dụ:

```
<?php
    $a = 10;
    $b =20;
    $kq = ($a>$b)?$a:$b;
    echo "Kết quả".$kq;// kết quả 20
?>
```

Câu lệnh if lồng nhau

```
if (<BTĐK1>) {
        // Thực hiện công việc nếu BTĐK1 đúng
    }
elseif (<BTĐK2>) {
        // Thực hiện công việc nếu BTĐK2 đúng
        }
    else {
        // Thực hiện công việc nếu BTĐK2 sai
        }
```

Câu lệnh switch

```
switch (gt)
{
    case gt1: <công việc 1>; break;
    case gt2: <công việc 2>; break;
    ...
    default: <công việc n+1>; break;
}
```

Câu lệnh while

while (<*BTĐK*>)

{

//mã lệnh;

}

Trong khi <BTĐK> còn đúng thì thực hiện công việc <mã lệnh>

Câu lệnh do … while

```
do
    {
    công việc nếu btđk còn đúng;
    }
while (btđk);
```

Câu lệnh for

```
for (init; condition; increment)
    {
    // mã lệnh;
    }
```

Câu lệnh foreach

```
foreach ($array as $value)
        {
        code to be executed;
        }
```

Ví dụ:

```
<?php
$a = array("2","4","6","7","9");
foreach($a as $var)
      echo $var."</br>";
?>
```

break và continue

- **Lệnh break**

  break cho phép ta thoát khởi cấu trúc điều khiển dựa trên kết quả của biểu thức logic

- **Lệnh continue**

  Khi gặp continue, các lệnh bên dưới continue tạm thời không thực hiện tiếp, khi đó con trỏ sẽ nhảy về đầu vòng lặp để kiểm tra giá trị của biểu thức điều kiện còn đúng hay không. continue thường đi kèm với một biểu thức logic

Mảng một chiều

Khái niệm mảng

- Mảng nói chung là một biến đặc biệt.
- Mảng bao gồm một dãy các ô nhớ cho phép biểu diễn thông tin dạng danh sách trong thực tế.
- Các phần tử mảng có thể có kiểu dữ liệu khác nhau.

Khai báo mảng

Cách 1: Khai báo chưa biết spt mảng

Cú pháp:

$ten_mang = array();

Ví dụ:

```
$a = array();
for ($i=0; $i<10; $i++)
    $a[$i] = $i * 2;
```

Khai báo mảng

Cách 2: Khai báo biết trước spt mảng

Cú pháp:

$ten_mang = array(<spt mảng>);

Ví dụ:

```
<?php
    $a = array(10);
    for ($i=0; $i<count($a); $i++)
        $a[$i] = $i * 2;
?>
```

Khái báo mảng

Cách 3: Khai báo biết giá trị của mảng

Cú pháp:

$ten_mang = array([khóa =>] giá_trị_1, …);

Trong đó:

- Khóa có thể là số nguyên hoặc chuỗi.
- Nếu ta không tạo giá trị khóa, thì khóa sẽ tự động phát sinh.

Khai báo mảng

Cách 4: Gán giá trị cho từng phần tử mảng

Cú pháp:

$ten_mang[ ] =<giá trị>;

Hoặc

$ten_mang[<giá trị khóa>] =<giá trị>;

Truy xuất phần tử mảng

Truy xuất đến phần tử mảng theo cú pháp:

$bien = $ten_mang[<giá trị khóa>];

Ví dụ:

$var1 = $mang[0];

$var2 = $mang[1];

$var3 = $mang[2];

$var4 = $mang[3];

Đếm số phần tử mảng

Để biết mảng có bao nhiêu phần tử ta sử dụng hàm count($name_array).

Ví dụ:

```
<?php
    $mang = array("4","6","8","0","3","5","9");
    $spt = count($mang);
    echo $spt; // kết quả 7
?>
```

Duyệt mảng

Duyệt mảng có từ khóa động

Ví dụ:

```
<?php
    $mang = array("4","6","8","0","3","5","9");
    $spt = count($mang);
    for($i = 0; $i < $spt; $i++ )
    {
        echo "\t" . $mang[$i];
    }
?>
```

Duyệt mảng

Duyệt mảng có khóa do người dùng tạo

Ví dụ:

```
<?php
    $mang = array("4","6","8","0","3","5","9");
    foreach($mang   as $vars )
    {
       echo "\t" . $vars;
    }
?>
```

Duyệt mảng

Duyệt để lấy giá trị khóa và giá trị phần tử.

Ví dụ:

```
<?php
    $mang = array("4","6","8","0","3","5","9");
    foreach($mang   as $khoa =>$vars )
    {
       echo "</br>" . [$khoa] .$vars;
    }
?>
```

Tkiếm và thay thế pt trên mảng

Tìm kiếm và thay thế một giá trị trên mảng.

```
function thay_the($mang, $gt_cu, $gt_moi)
{    $spt = count($mang);
     for ($i =0; $i < $spt; $i++ )
     {
        if($mang[$i] = $gt_cu)
         $mang = $gt_moi;
     }
}
```

Mảng hai chiều

Khai báo và khởi tạo

- Khai báo chưa biết số phần tử mảng

  $ten_mang = array(array(), array());

- Khởi gán giá trị cho từng phần tử mảng

  $ten_mang [vị trí hàng] [vị trí cột] = <giá trị>;

Truy xuất phần tử trong mảng

$bien = $ten_mang [vị trí hàng] [vị trí cột];

Ví dụ:

```
$a= array(array(1,2),array(3,4));
for ($i =0; $i< count($a); $i++)
{
    foreach($a[$i] as $var)
        echo $var;
    echo "</br>";
```

Đếm số dòng của mảng 2 chiều

Để biết mảng có bao dòng ta sử dụng hàm count($name_array).

Ví dụ:

```
<?php
$mang = array(array(1,2),array(3,4));
$spt = count($mang);
echo "Số hàng của mảng: ".$spt;
?>
```

Duyệt mảng 2 chiều

❖ Duyệt mảng có từ khóa động

```
$dong = count($mang);
for($i = 0; $i < $dong; $i++ ) {
    foreach($mang[$i]  as $vars )
    {
        echo "\t" . $vars;
    }
}
```

Xây dựng hàm trong PHP

Khai báo hàm

Để khai báo hàm ta sử dụng cấu trúc sau:

Cú pháp:

```
function ten_ham([danh sách tham số])
{
    // khối lệnh bên trong hàm
    [returm gia_trị;]
}
```

Ví dụ

```
<?php
function hien_thi()
{
echo "Chào các bạn";
}
// gọi lại hàm
hien_thi(); // Chào các bạn
?>
```

Ví dụ

```php
<?php
function tinh_tong($a, $b)
{
$c = $a + $b;
echo "Tổng:".$c;
}
// gọi lại hàm
tinh_tong(6,5); // Tổng: 11
?>
```

Ví dụ

```php
<?php
    function tinh_tong($a, $b)
    {
       $c = $a + $b;
       return $c;
    }
    // gọi lại hàm
    $t = tinh_tong(6,5);
    echo "Tổng: ".$t; //kết quả: Tổng 11
?>
```

Thư viện hàm

Hàm xử lý chuỗi

Hàm ltrim(str [,char]);

Ví dụ:

```
<?php
$st="aaaa   Hoàng Nam";
$st = ltrim($st,'a');
echo $st; //"Hoàng Nam"
?>
```

Hàm xử lý chuỗi

Hàm rtrim(str [,char]); xóa bỏ các ký tự trắng bên phải hoặc xóa bỏ các ký tự char bên phải.

Ví dụ:

```
<?php
$st="Hoàng Nam     aaaa";
$st = rtrim($st,'a');
echo $st; //"Hoàng Nam"
?>
```

Hàm xử lý chuỗi

Hàm trim(str [,char]);

Ví dụ:

```
<?php
$st="aaaa    Hoàng Nam     aaaa";
$st = trim($st,'a');
echo $st; //"Hoàng Nam"
?>
```

Hàm xử lý chuỗi

- Hàm addslashes($st): định dạng dữ liệu trong chuỗi để lưu vào CSDL.
- Để lưu chuỗi có các dấu nháy ' hay cặp ", dấu \, dấu \\ thì chúng ta dùng thêm dấu \ vào phía trước chúng như sau: \', \", \\, \\\.

Ví dụ:

```
echo addslashes($st); //Who\'re you?
$st ="Who're you?";
```

Hàm xử lý chuỗi

Hàm stripslashes($st): loại bỏ các dấu ', ", /… trong cơ sở dữ liệu

Ví dụ:

```
<?php
$st ="Who\'re you?";
echo stripslashes($st); //Who're you?
?>
```

Thư viện hàm

Hàm ucfirst($st): hàm chuyển ký tự đầu tiên của chuỗi thành hoa.

Ví dụ:

```
<?php
$st  = "nguyễn Hoàng Nam";
echo ucfirst($st); // Nguyễn Hoàng Nam
?>
```

Hàm xử lý chuỗi

Hàm ucwords($st): viết hoa kí tự đầu tiên của mỗi từ.

Ví dụ:

```
<?php
$st  = "nguyễn hoàng nam";
echo ucwords($st); // Nguyễn Hoàng Nam
?>
```

Hàm xử lý chuỗi

Hàm strtolower($st): chuyển kí tự bất kỳ thành chữ thường.

Ví dụ:

```
<?php
$st  = "NGUYỄN HOÀNG NAM";
echo strtolower($st); // nguyễn hoàng nam
?>
```

Hàm xử lý chuỗi

Hàm strtoupper($st): biến kí tự bất kỳ thành chữ hoa.

Ví dụ:

```
<?php
$st  = "nguyễn hoàng nam";
echo strtoupper($st); // NGUYỄN HOÀNG NAM
?>
```

Hàm xử lý chuỗi

Hàm strlen($st): Kết quả trả về độ dài của chuỗi

Ví dụ:

```
<?php
$st  = "Nguyễn Hoàng Bảo Nam";
echo strlen($st); // 20
?>
```

Hàm xử lý chuỗi

Hàm strcmp($str1,$str2): hàm so sánh chuỗi không phân biệt chữ hoa và chữ thường, hàm này trả về kết quả là:

Ví dụ:

```
echo strcmp('Hải','Nam'); // kết quả -1
echo strcmp('Nam','Nam') "; // kết quả  0
echo strcmp('Nam','Hải'); // Kết quả 1
```

Hàm xử lý chuỗi

Hàm strstr($st1, $st2) và strchr($s1, $st2): tìm chuỗi st2 trong st1, nếu tồn tại hàm sẽ trả về chuỗi con trong $st1 bắt đầu $st2.

Ví dụ:

```
$email = "hodienloi@hitc.edu.vn";
echo strstr($email,'@'); //@hitc.edu.vn
```

Hàm xử lý chuỗi

Hàm strpos($st1,$st2): tìm vị trí chuỗi con $st2 trong chuỗi $st1

Ví dụ:

```
$email = "hodienloi@hitc.edu.vn";
echo strpos($email,'@hitc'); // 9
```

Hàm xử lý chuỗi

Hàm str_replace($seach, $rep, $st): tìm kiếm và thay thế

Ví dụ:

```
$email = "hodienloi@hitc.edu.vn";
echo str_replace('hitc.edu.vn','yahoo.com',$email);
```

Hàm xử lý chuỗi

Hàm strrev($st): đảo ngược 1 xâu.

Ví dụ:

```
<?php
$st  = "ssalc ym olleH";
echo strrev($st); //Hello my class
?>
```

Hàm xử lý chuỗi

Hàm explode($ch, $st): tách chuỗi $st thành nhiều phần tử và gán cho phần tử mảng, tách theo ký tự $ch.

Ví du:

```
<?php
    $st = "1,2,3,4,5,6,7,4,3,5,6";
    $mang = explode(',',$st);
    print_r($mang);
?>
```

Hàm xử lý chuỗi

Hàm implode($ch, $mang): kết hợp các phần tử mảng thành chuỗi, các phần tử mảng cách nhau bởi $ch

Ví dụ:

```
<?php
    $a = array("Nguyễn","Hoàng","Bảo","Nam");
    $st = implode(' ',$a);
    echo($st);
?>
```

Hàm xử lý chuỗi

Hàm chr(number): đổi số thành ký tự trong bảng mã ASCII

Ví dụ:

```
<?php
    echo chr(97); // kết quả a
    echo chr(65); //kết quả A
?>
```

Hàm xử lý số

- Hàm abs(x): Giá trị tuyệt đố của x

Ví dụ:

```
echo abs(-5); // kết quả 5
echo abs(5); // kết quả 5
```

- Hàm round(x[,i]) : hàm làm tròn

Ví dụ: $so = 1234.567;

```
echo round($so); // Kết quả 1234
echo round($so,2); // Kết quả 1234.57
echo round($so,-2); // Kết quả 1200
```

Hàm xử lý số

- Hàm exp(x): e^x

  Ví dụ:

  echo exp(2); //kết quả 7.3890560989307

- Hàm lượng giác: sin(x) /cos(x)/tan(x)

  Ví dụ:

```
<?php
    echo sin(30*PI()/180); // Kết quả 0.5
    echo cos(30*PI()/180); // Kết quả 0.8660…
    echo tan(30*PI()/180);// Kết quả 0.5773…
?>
```

Hàm xử lý số

- Hàm sqrt(x)

  Ví dụ:

  echo "Căn bậc 2:".sqrt(9); // Kết quả 3

- Hàm rand(n1,n2)

  Ví dụ:

```
<?php
    echo  rand();
    echo  rand(10,100);
?>
```

Hàm xử lý số

- Hàm pow(a,x)

  Ví dụ:

  echo pow(10,2); // kết quả 100

- Hàm pi()

  Ví dụ:

  echo pi(); // kết quả 3.14159265358898

Hàm xử lý số

Hàm range(gt1,gt2): hàm lấy giá trị nguyên trong khoảng gt1.. gt2 và trả về mảng

Ví dụ:

```
$mang = range(10, 100);
print_r($mang);
```

Hàm xử lý số

Hàm number_format(): Định dạng số

Ví dụ:

```
<?php
$number = 1234.5678;
echo number_format($number); // 1,235
echo number_format($number, 2, ',', ' ');//1 234,57
echo number_format($number, 2, '.', '');//1234.57
?>
```

Hàm xử lý dữ liệu ngày, giờ

Hàm checkdate($month, $day, $year) kiểm tra ngày nhập vào có hợp lệ không?

Ví dụ:

```
<?php
$m = 2; $d = 30; $y = 2010;
echo checkdate($m, $d, $y); //  0
$d = 24;
echo checkdate($m, $d, $y); // 1
?>
```

Hàm xử lý dữ liệu ngày, giờ

Hàm date(format [, timestamp])

Trong đó, format

D	Ngày Mon đến Sun
d	Ngày 01 đến 31
M	Tháng Jan đến Dec
m	Tháng 01 đến 12
Y	Năm có 4 ký số
y	Năm có 2 ký số

Hàm xử lý dữ liệu ngày, giờ

Hàm getdate(): lấy *seconds, minutes, hours, mday, wday, mon, year, yday, weekday, month, 0 và trả về một mảng có chỉ số mảng là seconds,….*

Ví dụ:

```
$d = getdate();
echo $d['seconds'];
echo $d['minutes'];
echo $d['hours'];
echo $d['mday'];…….
```

Hàm xử lý dữ liệu ngày, giờ

Hàm localtime(): lấy giây, phút, giờ, ngày của tháng(Jan..Dec), tháng của năm, Years since 1900, ngày của tuần, ngày của năm và Is daylight savings time in effect của ngày hiện tại và trả về là mảng với chỉ số mảng từ 0 đến 8

Ví dụ:

```
$d = localtime();
for($i=0; $i<9; $i++)
    echo $d[$i];
```

Biểu mẫu form

Đặc điểm của form

- Form là một thành phần của trang web.
- Các đối tượng thể hiện của form:
  - TextField
  - Textarea
  - Button
  - RadioButton
  - CheckBox
  - List/Menu

Đặc điểm của form

- Các thuộc tính cơ bản của form:
  - name: tên form
  - action: hành động
  - method: phương thức
  - vị trí: _top, _parent, _self, _black
- Loại nội dung trong form
  - enctype = content-type với content-type:
    - application/x-www-form-urlenconded
    - multipart/fom-data

Action :Hành động

- Khi muốn lấy giá trị từ form đưa về trang nào thì trong action ta nhập vào trang đó.

Vi dụ:

```
<form action = "xu_ly.php" method ="POST"
    name="form1">

……….

</form>
```

Method :Phương thức

- Phương thức là hình thức chuyển giá trị trên form đến trang action.
- Có 2 phương thức:
  - POST: chuyển giá trị đến trang action và nhận giá trị ta dùng biến $_POST hoặc $_REQUEST.
  - GET: chuyển giá trị đến trang action và nhận giá trị ta dùng biến $_GET hoặc $_REQUEST

Textfield

- Dùng để nhập và hiển thị dữ liệu: có 3 loại sau:
  - Single line: nhập và hiển trên 1 dòng văn bản
  - Multi line: nhập và hiển trên nhiều dòng văn bản
  - Password: hiển thi dấu * thay cho ký tự

Textarea

- Dùng để nhập và hiển thị dữ liệu dạng multi line:
  - Num line: số dòng văn bản hiển thị
  - Wrap: Tăng số dòng bản nếu văn bản vượt quá num line

Button

- Dùng để tạo ra các nút như submit, reset:
  - Button submit: khi ta nhấn vào nút này dữ liệu trên form sẽ gửi về server.
  - Button reset: khi nhấn vào nút này các đối tượng trên form sẽ reset.

Checkbox

- Checkbox là một đối tượng có hai trạng thái on/off
- Trên form có nhiều checkbox, đồng thời ta có thể chọn nhiều checkbox.

Radiobutton/radiogroup

- Radio là một đối tượng có hai trạng thái on/off.
- Trên form có nhiều radio, các radio cùng nhóm có cùng name.
- Chỉ có thể chọn một radio trên nhóm.

List/menu

- Là danh sách nhiều mục chọn.
- Có thể chọn một hoặc nhiều mục chọn.
- Muốn chọn nhiều mục chọn thì thuộc tính "Selectons Allow Multiple"

Filefield

- Đối tượng này cho phép người dùng chọn một file để xem xét các thuộc tính hoặc upload,…

Form sử dụng phương thức POST

❖ Đặc điểm

- Biến $_POST được dùng để lấy các giá trị trên form thông qua phương thức POST.
- Thông tin được gửi từ form với phương thức này không giới hạn dung lượng thông tin gửi đi.
- Thông tin được gửi bằng phương thức POST sẽ không hiện thị lên địa chỉ URL nên người dùng không thể thấy được.

Form sử dụng phương thức POST

❖ Cách sử dụng

Cú pháp lấy giá trị của một đối tượng trên form sau khi form submit:

$_POST[‘tên điều kiển’];

Form sử dụng phương thức GET

❖ Đặc điểm

- Biến $_GET được dùng để lấy giá trị trên form bằng phương thức GET. Thông tin gửi qua không vượt quá 100 ký tự
- Thông tin được gửi đi bằng phương thức GET sẽ hiển thị trên địa chỉ URL nên người dùng có thể thấy được các thông tin này.

Form sử dụng phương thức GET

❖ Cách sử dụng

- Cú pháp lấy giá trị trên form bằng phương thức GET như sau:

```
$_GET["Tên điều khiển"];
```

Biến $_REQUEST

❖ Đặc điểm

- Biến $_REQUET chọn nội dung cả các biến $_POST, $GET, $_COOKIE
- Có thể lấy thông tin không cần xác định phương thức form.
- Độ bảo mật thông tin thấp

Biến $_REQUEST

❖ Cách sử dụng

- Cú pháp lấy giá trị bằng biến $_REQUEST

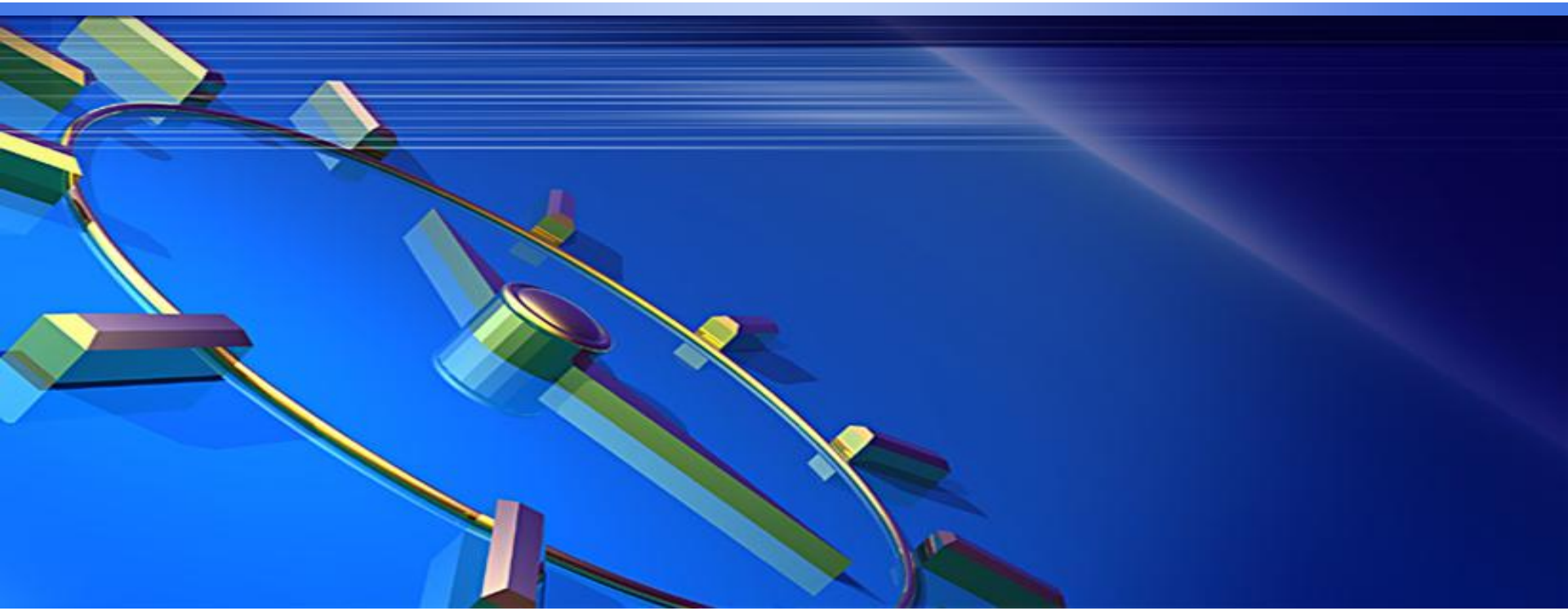
$_REQUEST["Tên điều khiển"];

Biến $_REQUEST

❖ Đặc điểm

- Biến $_FILE là một mảng chứa các thông tin của file được chọn từ bảng filefield.
- Một số thuộc tính:
  - $_FILE[điều khiển]["name"]
  - $_FILE[điều khiển]["size"]
  - $_FILE[điều khiển]["tmp_name"]
  - $_FILE[điều khiển]["type"]
  - $_FILE[điều khiển]["error"]

Nội dung tiếp theo



Giáo viên: Trần Cao Nhân

Hướng dẫn lập trình PHP

Hướng dẫn về PHP

Tác giả: Black Art gởi trong diễn đàn hackervn.net

1. Php là gì?

PHP là một ngôn ngữ lập trình kiểu script , chạy trên Server và trả về mã HTML cho trình duyệt.
VD:

```
<html>
 <head>
 <title>Example</title>
 </head>
 <body>
 <?php echo "Hi, I'm a PHP script!"; ?>
 </body>
</html>
```

Mã PHP được đặt trong kiểu tag đặc biệt cho phép ta có thể vào hoặc ra khỏi chế độ PHP

2. Cài đặt như thế nào?

Cài PHP trên nền Window với IIS version 3 hoặc 4:
- Copy php.ini - dist vào thư mục Window. Đổi tên thành php.ini
- Sửa file php.ini như sau:
- Đặt lại đường dẫn mục : 'extension_dir' chuyển tới vị trí cài đặt php hoặc nơi để các file php_*.ini. VD : c:\php
- Đặt đường dẫn chính xác đến file: browscap.ini trong thư mục Window.
- Chạy MMC , chọn Web site hoặc thư mục của ứng dụng
- mở mục Property của thư mục , chọn vào Home Directory, Virtual Directory hay Directory.
- Vào mục: Configuration->App Mappings ->Add->Executable : c:\php\php.exe %s %s. Chú ý phải có %s %s .
- Trong ô Extension, đánh đuôi file bạn muốn gắn với mã php. Thường là .php hoặc .phtml
- Đặt chế độ sercurity thích hợp. Nếu dùng hệ thống NTFS, cần đặt cho phép thực hiện với thư mục có php.exe

II.Ngôn ngữ PHP:

1/ Có 4 cách để dùng PHP

```
1.  <? echo ("this is the simplest, an SGML processing instruction\n"); ?>
2.  <?php echo("if you want to serve XML documents, do like this\n"); ?>
3.  <script language="php">
 echo ("some editors (like FrontPage) don't
 like processing instructions");
 </script>
4.  <% echo ("You may optionally use ASP-style tags"); %>
 <%= $variable; # This is a shortcut for "<%echo .." %>
```

Cách 1 chỉ có thể sử dụng nếu những tag ngắn được cho phép sử dụng. Có thể sửa short_open_tag trong cấu hình của php hoặc biên dịch file php với lựa chọn cho phép dùng các tag ngắn.
Tương tự như vậy, cách thứ 4 chỉ có tác dụng nếu asp_tag được đặt trong file cáu hình của PHP
PHP cho phép hỗ trợ các kiểu chú thích của cả C, C++ và Unix shell.
VD:

```
<?php
 echo "This is a test"; // This is a one-line c++ style comment
 /* This is a multi line comment
yet another line of comment */
 echo "Another Test"; # This is shell-style style comment
?>
```

2. Các kiểu dữ liệu:

2.1Số nguyên : Được khai báo và sử dụng giá trị giiống với C:
VD: $a = 1234; # decimal number
$a = -123; # a negative number
$a = 0123; # octal number (equivalent to 83 decimal)

$a = 0x12; # hexadecimal number (equivalent to 18 decimal)
2.2 Số thực:
VD :$a = 1.234; $a = 1.2e3;
Chú ý: Khi sử dụng các số thực để tính toán , có thể sẽ làm mất giá trị của nó. Vì vậy, nên sử dụng các hàm toán học trong thư viện chuẩn để tính toán.

2.3 Xâu :
Có 2 cách để xác định 1 xâu: Đặt giữa 2 dấu ngoặc kép (") hoặc giữa 2 dấu ngoặc đơn ('.
Chú ý: Các biến giá trị sẽ không được khai triển trong xâu giữa 2 dấu ngoặc đơn.

2.4 : Mảng :
Mảng thực chất là gồm 2 bảng : bảng chỉ số và bảng liên kết.
a. Mảng 1 chiều : Có thể dùng hàm : list() hoặc array() hoặc liệt kê các giá trị của từng phần ử trong mảng để tạo mảng . Có thể thêm các giá trị vào mảng để tạo thành 1 mảng. -->
Dùng giống trong C
Có thể dùng các hàm asort(), arsort(), ksort(), rsort(), sort(), uasort(), usort(), and uksort() để sắp xếp mảng. Tuỳ thuộc vào việc bạn định sắp xếp theo kiểu gì.
b. Mảng nhiều chiều: Tương tự C. bạn có thể dùng như VD sau:
$a[1] = $f; # one dimensional examples
$a["foo"] = $f;
$a[1][0] = $f; # two dimensional
$a["foo"][2] = $f; # (you can mix numeric and associative indices)
$a[3]["bar"] = $f; # (you can mix numeric and associative indices)
PHP3 không thể tham chiếu trự tiếp từ một mảng nhiều chièu khi ở bên trong 1 xâu:
VD :
$a[3]['bar'] = 'Bob';
echo "This won't work: $a[3][bar]";
Kết quả trả ra sẽ là: This won't work: Array[bar].
Nhưng với VD sau sẽ chạy đúng:
$a[3]['bar'] = 'Bob';
echo "This will work: " . $a[3][bar];
Với PHP4, vấn đề sẽ được giả quyết bằng cách cho mảng vào giữa 2 dấu ngoặc móc { }
VD : $a[3]['bar'] = 'Bob';
echo "This will work: {$a[3][bar]}";
PHP không yêu cầu phải khai báo trước kiểu dữ liệu cho các biến, Kiểu dữ liệu của biến sẽ phụ thuộc vào kiểu dữ liệu mà bạn gán cho nó,
VD :
$foo = "0"; // $foo là kí tự ASCII 48
$foo++; // $foo is xâu "1" (ASCII 49)
$foo += 1; // $foo là số nguyên (2)
$foo = $foo + 1.3; // $foo là số thực (3.3)
$foo = 5 + "10 Little Piggies"; // $foo là số nguyên (15)

3. Biến - giá trị:
PHP quy định một biến được biểu diễn bắt đầu bằng dấu $, sau đó là một chữ cái hoặc dấu gạch dưới.

3.1 Một số biến đã được tạo sẵn :
argv : Mảng tham số truyền cho script. Khi đoạn script chạy bằng dòng lệnh, tham số này sẽ được dùng giống như C để truy nhập các tham số trên dòng lệnh.
argc : số các tham số đươc truyền. Dùgn với argv;

PHP_SELF : tên cỷa đoạn mã script đang thực hiện. Nếu PHP đang được chạy từ dòng lệnh thì tham số này không có giá trj.

HTTP_COOKIE_VARS: một mảng các giá trị được truyền tới script hiện tại bằng HTTP cookie. Chỉ có tác dụng nếu "track_vars" trong cấu hình được đặt hoặc chỉ dẫn <?php_track_vars?>

HTTP_GET_VARS: Mảng các giá trị truyền tới script thông qua phương thức HTTP GET. Chỉ có tác dụng nếu "track_vars" trong cấu hình được đặt hoặc chỉ dẫn <?php_track_vars?>

HTTP_POST_VARS:

3.2 Phạm vi giá trị:
PHP coi 1 biến có một giới hạn. Để xác định một biến toàn cục (global) có tác dụng trong một hàm , ta cần khai báo lại. Nếu không giá trị của biến sẽ được coi như là cục bộ trong hàm.
VD :

```
$a = 1;
$b = 2;

Function Sum () {
 global $a, $b;

 $b = $a + $b;
}

Sum ();
echo $b;
```

Khi có khai báo global ở trên, $a và $b được cho biết đó là những biến toàn cục. Nếu không có khai báo global, $a và $b chỉ được coi là các biến bên trong hàm Sum(). Điều này khác với C
Một cách khác để dùng biến toàn cục trong 1 hàm là ta dung mảng $GLOBAL của PHP
VD ở trên sẽ có thể viết như sau:

```
$a = 1;
$b = 2;

Function Sum () {
 $GLOBALS["b"] = $GLOBALS["a"] + $GLOBALS["b"];
}

Sum ();
echo $b;
```

Một chú ý khác là khai báo static. Với khai báo này bên trong một hàm với 1 biến cục bộ, giá trị của biến cục bộ đó sẽ không bị mất đi khi ra khỏi hàm.
VD :

```
Function Test () {
 static $a = 0;
 echo $a;
 $a++;
}
```

Với khai báo như trên , $a sẽ không mất đi giá trị sau khi thực hiện lơi gọi hàm Test() mà $a sẽ được tăng lên 1 sau mỗi lần gọi hàm Test().

3.3 Tên biến:
Một biến có thể gắn với 1 cái tên .
VD:

```
$a = "hello";
$$a = "world";
==> $hello = "world"
```

và

```
echo "$a ${$a}";
<==> echo "$a $hello";
```

Kết quả ra sẽ là : hello world

Chú ý : bạn có thể gặp phải trường hợp không rõ ràng khi sử dụng cách này với mảng . VD : $$a[1] sẽ hiểu là bạn muốn dùng $a[1] như 1 biến hay dùng $$a như 1 biến với [1] là chỉ số?
Để tránh trường hợp này , cần có sựu phân biệt rõ bằng dấu { }. VD :${$a[1]} hoặc ${$a}[1].

3.4 Các giá trị bên ngoài phạm vi PHP
HTML Form : khi 1 form gắn với 1 file php qua phương thức POST
VD:
<form action="foo.php3" method="post">
Name: <input type="text" name="name">
<input type="submit">
</form>

PHP sẽ tạo 1 biến $name bao gồm mọi giá trị trong trường Name của Form.
PHP có thể hiểu được một mảng một chiều gồm các giá trị trong 1 form. Vì vậy, bạn có thể nhóm những giá trị liên quan lại với nhau hoặc sử dụng đặc tính này để nhận các giá trị từ 1 đầu vào tuỳ chọn.(multi select input)
Khi tính chất track_vars được đặt trong cấu hình hoặc có chỉ dẫn <?php_track_vars?> . các giá trị được submit được lấy ra qua phương thức GET và POST có thể lấy từ 2 mảng toàn cục $HTTP_POST_VARS và $HTTP_GET_VARS

IMAGE SUBMIT:
Khi dùng 1 image để thực hiện submit, có thể dùng tag như sau:
<input type=image src="image.gif" name="sub">

Khi người dùng click chuột trên ảnh, form tương ứng sẽ được gửi tới server kèm theo 2 giá trị thêm vào : sub_x và sub_y. Những biến này sẽ lưu giữ toạ độ mà người dùng đã click chuột trên ảnh. Người lạp trình có kinh nghiệm sẽ thấy rằng trên thực tế, trình duyệt gửi các giá trị có chu kỳ thời gian nhưng PHP đã tạo cảm giãc gần như việc gửi các giá trị đó là liên tục.

HTTP Cookies :
PHP hỗ trỡ HTTP cookies theo định dạng của Netscape. Cookies file lưu giữ thông tin của các trình duyệt từ xa và qua đó có thể theo dõi hay nhận biết người sử dụng. Có thể dùng Cookies bằng hàm SetCookie(). Hàm này cần được gọi trước khi thông tin được gửi tới trình duyệt. Bất kỳ cookie nào gửi tới bạn từ máy khách (client) sẽ tự động chuyển thành dữ liệu của phương thức GET và POST.
Nếu bạn muốn có nhiều giá trị trong 1 cookie, chỉ cần thêm vào dấu [ ] với tên của cookie.
VD :
SetCookie ("MyCookie[]", "Testing", time()+3600);
Chú ý rằng cookie sẽ thay thế cho cookie cùng tên, trừ trường hợp khác đường dẫn hoặc miền.

BIẾN MÔI TRƯỜNG:
PHP tự động tạo biến cho các biến môi trường như 1 biến bình thường của PHP
VD :
echo $HOME; /* Shows the HOME environment variable, if set. */

Vì thông tin tới qua các phương thức GET, POST , Cookie cũng tự đông tạo các biến PHP, thỉnh thoảng bạn nên đọc 1 biến từ môi trường để chắc chắn rằng bạn có đúng version. Hàm getenv() và putenv() giúp bạn đọc và ghi với các biến môi trường.

DẤU CHẤM TRONG TÊN BIẾN:
Bình thường, PHP không thay đổi tên biến khi biến đó được truyền vào đoạn script. Tuy nhiên, nên chú ý rằng dáu chấm (.) không phải là một ký hiệu hợp lệ trong tên biến đối với PHP. Vì vậy, PHP sẽ tự động thay thế các dấu chấm bằng dấu gạch dưới.(_)

4. CONSTANTS

PHP định nghĩa sẵn một vài hằng số:
FILE : tên của script file đang được thực hiện.
LINE : số dòng của mã script đang được thực hiện trong script file hiện tại.
_PHP_VERSION_ : version của PHP
_PHP_Ó : tên hệ điều hành mà PHP đang chạy
TRUE
FALSE
E_ERROR : báo hiệu có lỗi
E_PARSE : báo lỗi sai khi biên dịch
E_NOTICE : Một vài sự kiện có thể là lỗi hoặc không.
E_ALL :

Có thể định nghĩa một hằng số bằng hàm define()
VD :

```
<?php
define("CONSTANT", "Hello world.");
echo CONSTANT; // outputs "Hello world."
?>
```

5. BIỂU THỨC

Biểu thức là một phần quan trọng trong PHP. Phần lớn mọi thứ ban viết đều được coi như 1 biểu thức. Điều này có nghĩa là mọi thứ đều có 1 giá trị.
Một dạng cơ bản nhất của biểu thức bao gồmcác biến và hằng số.
PHP hỗ trợ 3 kiểu giá trị cơ bản nhất: số nguyên, số thực, và xâu. Ngoài ra còn có mảng và đối tượng. Mỗi kiểu giá trị này có thẻ gán cho các biến hay làm giá trị trả ra của các hàm.
Bạn có thể thao tác với các biến trong PHP giống như với trong C.
VD

```
$b = $a = 5;  /* assign the value five into the variable $a and $b */
$c = $a++;  /* post-increment, assign original value of $a
(5) to $c */
$e = $d = ++$b; /* pre-increment, assign the incremented value of
$b (6) to $d and $e */

/* at this point, both $d and $e are equal to 6 */

$f = double($d++);  /* assign twice the value of $d before
the increment, 2*6 = 12 to $f */
$g = double(++$e);  /* assign twice the value of $e after
the increment, 2*7 = 14 to $g */
$h = $g += 10;  /* first, $g is incremented by 10 and ends with the
value of 24. the value of the assignment (24) is
then assigned into $h, and $h ends with the value
of 24 as well. */
```

6.CÁC CẤU TRÚC LỆNH:

6.1 If ....else....else if:

```
if (điều kiện)  { do something; }
elseif  ( điều kiện )  { do something;}
else { do something;}
```

6.2 Vòng lặp :

```
while ( DK) { ...}
do { .....} white ( DK );
for (bieuthuc1; bieuthuc2; bieu thuc3) {.....}
```

PHP 4 only :

```
foreach(array_expression as $value) statement
foreach(array_expression as $key => $value) statement
```

6.3 break và continue:

break : thoát ra khỏi vòng lặp hiện thời
continue : bỏ qua vòng lặp hiện tại, tiếp tục vòng tiếp theo.

6.4 switch
switch (tên biến) {
case trường hợp 1: ..... break;
case trường hợp 2: ..... break;
case trường hợp 3: ..... break;
default :
}

7 HÀM:

Dùng giống với C++. Ngoại trừ bạn không cần phải khai báo kiểu cho tham số của hàm:

7.1 Tham trị :

VD:
```
function takes_array($input) {
 echo "$input[0] + $input[1] = ", $input[0]+$input[1];
}
```

7.2 Tham biến:

```
function add_some_extra(&$string) {
$string .= 'and something extra.';
}
```

7.3 Tham số có giá trị mặc định:

```
function makecoffee ($type = "cappucino") {
return "Making a cup of $type.\n";
}
```
Chú ý : khi sử dụng hàm có đối số có giá trị mặc định, các biến này sẽ phải nằm về phía phải nhát trong danh sách đối số.
VD : Sai
```
function makeyogurt ($type = "acidophilus", $flavour) {
 return "Making a bowl of $type $flavour.\n";
}
```
Đúng ;
```
function makeyogurt ($flavour, $type = "acidophilus") {
 return "Making a bowl of $type $flavour.\n";
}
```

7.4 Giá trị trả lại của hàm:

Có thể là bất kỳ giá trị nào, Tuy vây, không thể trả lại nhiều giá trị riêng lẻ nhưng có thể trả lại một mảng các giá trị.
VD
```
function small_numbers() {
 return array (0, 1, 2);
}
```

Để trả lại một tham trỏ, bạn cần có dấu & ở cả khai báo của hàm cũng như ở giá trị trả lại.
VD :
```
function &returns_reference() {
 return &$someref;
}

$newref = &returns_reference();
```

7.5 Hàm biến:

PHP cho phép sử dụng hàm giá trị Nghĩa là khi một biến được goi có kèm theo dấu ngoặc đơn , PHP sẽ tìm hàm có cùng tên với giá trị biến đó và thực hiện

VD

```
<?php
function foo() {
 echo "In foo()<br>\n";
}

function bar( $arg = '' ) {
 echo "In bar(); argument was '$arg'.<br>\n";
}

$func = 'foo';
$func();
$func = 'bar';
$func( 'test' );
?>
```

8. CÁC TOÁN TỬ:

PHP có các toán tử cho các phép số học : + - * / %
Các toán tử logic : and or xor ! && ||
Toán tử thao tác với bit : & | ^ ~ << >>
Toán tử so sánh : ==, != ,< ,> ,<=, >=, ==== (bằng va cùng kiểu - PHP4 only), !== (khác hoặc khác kiểu - PHP4 only)
Toán tử điều khiển lỗi : @ - khi đứng trước 1 biểu thức thì các lỗi của biểu thức sẽ bị bỏ qua và lưu trong $php_errormsg
VD:

```
<?php
/* Intentional SQL error (extra quote): */
$res = @mysql_query ("select name, code from 'namelist") or
 die ("Query failed: error was '$php_errormsg'");
?>
```

Toán tử thực thi : ` `- PHP sẽ thực hiện nội dung nằm giữa 2 dấu ` như 1 lệnh shell. Trả ra giá trị là kết quả thực hiện lệnh
VD :

```
$output = `ls -al`; //liệt kê các file bằng lệnh Linux
echo "<pre>$output</pre>";
```

9 LỚP VÀ ĐỐI TƯỢNG:

Class: là tập hợp các biến và hàm làm việc với các biến này. Một lớp có định dạng như sau:

```
<?php
class Cart {
 var $items;  // Items in our shopping cart

 // Add $num articles of $artnr to the cart

 function add_item ($artnr, $num) {
 $this->items[$artnr] += $num;
 }

 // Take $num articles of $artnr out of the cart
function remove_item ($artnr, $num) {
 if ($this->items[$artnr] > $num) {
 $this->items[$artnr] -= $num;
 return true;
 } else {
 return false;
 }
 }
}
?>
```

Lớp Cart ở đây là một kiểu dữ liệu, vì vậy bạn có thể tạo một biến có kiểu này với toán tử new
VD:
```
$cart = new Cart;
$cart->add_item("10", 1);
```

Lớp có thể được mở rộng bằng những lớp khác. Lớp mới thu được có tất cả những biến và hàm của cá lớp thành phần. Thực hiện việc thừa kế này bằng từ khoá "extends". Chú ý : kế thừa nhiều lớp 1 lúc không được chấp nhận.
VD :
```
class Named_Cart extends Cart {
 var $owner;

 function set_owner ($name) {
 $this->owner = $name;
 }
}
```

Các hàm khởi tạo của lớp được gọi tự động khi bạn gọi toán tử new.
Tuy nhiên, các hàmkhởi tạo của lớp cha sẽ không được gọi khi hàm khởi tạo của lớp con được gọi. Hàm khởi tạo có thể có đối số hoặc không,

10. THAM CHIẾU:

Tham chiếu trong PHP có nghĩa là lấy cùng 1 giá trị bằng nhiều tên biến khác nhau. Khác với con trỏ trong C, tham chiếu là một bảng các bí danh. Chú ý : trong PHP, tên biến và nôi dung của biến là khác nhau. Vì vậy, cùng 1 nội dugn có thể có nhiều tên khác nhau.
Tham chiếu PHP cho phép bạn tạo 2 biến có cùng nôi dung.
VD :
$a = & $b; -- > $a, $b trỏ tới cùng 1 giá trị.
Tham chiếu truyền giá trị bằng tham chiếu. Thực hiện việc này bằng cách tạo một hàm cục bộ và truyền giá trị được tham chiếu
VD:
```
function foo (&$var) {
 $var++;
}

$a=5;
foo ($a);
```
--> Kết quả : $a = 6;
Giá trị trả lại của một hàm bằng tham chiếu rất tiện khi bạn muốn sử dụng hàm để tìm 1 giá trị trong 1 phạm vi nào đó.
VD :
```
function &find_var ($param) {
 ...code...
 return $found_var;
}

$foo =& find_var ($bar);
```

Khi bạn muốn loại bỏ mối liên kết giữa tên biến và giá trị của biến, sử dụng hàm unset()
VD :
```
$a = 1;
$b =& $a;
unset ($a);
```

11. THAO TÁC VỚI ẢNH:

PHP không bị giới hạn với mã HTML được trả lại cho trình duyệt. Vì vậy, có thể dùng PHP để tạo và thao tác với các file ảnh có định dạng khác nhau, bao gồm :gif, png, jpg, wbmp,

and xpm. PHP có thể đưa các file ảnh trực tiếp đến các trình duyệt. Bạn sẽ cần biên dịch PHP với thư viện GD bao gồm các hàm thao tác với ảnh. GD và PHP có thể sẽ cần có thêm 1 số thư viện khác, tuỳ thuộc vào định dạng của file ảnh cần dùng.
VD : Tạo ảnh GIF với PHP

```
<?php
 Header("Content-type: image/gif");
 $string=implode($argv," ");
 $im = imagecreatefromgif("images/button1.gif");
 $orange = ImageColorAllocate($im, 220, 210, 60);
 $px = (imagesx($im)-7.5*strlen($string))/2;
 ImageString($im,3,$px,9,$string,$orange);
 ImageGif($im);
 ImageDestroy($im);
?>
```

giả sử VD trên trong file button.php . KHi đó, để sử dụng ta dùng tag : <img src="button.php?text">. ,

12. MySQL và PHP

Để connect tới 1 CSDL trên MySQL server rất đơn giản. Bạn chỉ cần dùng hàm mysql_connect(host, user, password) để mở 1 kết nối tới MYSQL server với kết quả là giá trị trả về của hàm (Giả sử là biến $db).
Sau đó, dùng hàm mysql_select_db(database_name, link_id) để chọn CSDL bạn muốn kết nối.
Để thực hiện một câu lệnh truy vấn, dùng hàm mysql_query(query, link_ID). Giá trị trả lại của hàm là kết quả của câu truy vấn. Nếu bỏ qua link_ID thì kết nối cuối cùng tới MySQL server sẽ được thực hiện.
Bạn có thể xem ví dụ sau sẽ hiểu rõ hơn.

```
<html>
<body>
<?php
$db = mysql_connect("localhost", "root");

mysql_select_db("mydb",$db);

$result = mysql_query("SELECT * FROM employees",$db);

printf("First Name: %s<br>\n", mysql_result($result,0,"first"));

printf("Last Name: %s<br>\n", mysql_result($result,0,"last"));

printf("Address: %s<br>\n", mysql_result($result,0,"address"));

printf("Position: %s<br>\n", mysql_result($result,0,"position"));

?>

</body>
</html>
```

Ngoài cách dùng hàm mysql_result() để lấy kết quả thực hiện , bạn có thể dùng các hàm khác : mysql_fetch_row(), mysql_fetch_array(), and mysql_fetch_object().
* mysql_fetch_row($result) : trả về một mảng các dữ liệu lấy từ 1 dòng trong CSDL. Nếu đã ở cuối CSDL, giá trị trả về là false. Bạn phải dùng chỉ số của các trường trong CSDL nếu muốn lấy dữ liệu,
VD :

```
<html>
<body>
```

```
<?php
$db = mysql_connect("localhost", "root");
mysql_select_db("mydb",$db);
$result = mysql_query("SELECT * FROM employees",$db);

echo "<table border=1>\n";
echo "<tr><td>Name</td><td>Position</tr>\n";
while ($myrow = mysql_fetch_row($result)) {
printf("<tr><td>%s %s</td><td>%s</td></tr>\n", $myrow[1], $myrow[2], $myrow[3]);
}
echo "</table>\n";
?>
</body>
</html>
```

* mysql_fetch_array (int result [, int result_type]) : trả về một mảng chứa dữ liệu lấy từ 1 dòng trong CSDL. Tương tự như mysql_fetch_row() nhưng bạn có thể lấy trực tiếp tên trường để lấy dữ liệu.
VD : Cùng một CSDL như trên, bạn có thể thấy sự khác biệt

```
<html>
<body>
<?php
$db = mysql_connect("localhost", "root");
mysql_select_db("mydb",$db);
$result = mysql_query("SELECT * FROM employees",$db);
if ($myrow = mysql_fetch_array($result)) {
 do {
 printf("<a href=\"%s?id=%s\">%s %s</a><br>\n", $PHP_SELF, $myrow["id"],
$myrow["first"], $myrow["last"]);
 } while ($myrow = mysql_fetch_array($result));
} else {
 echo "Sorry, no records were found!";
?>
</body>
</html>
```

*mysql_fetch_object : chuyển kết quả thành một đối tượng với các trường là các trường trong CSDL.
VD :

```
<?php
mysql_connect ($host, $user, $password);
$result = mysql_db_query ("database", "select * from table");
while ($row = mysql_fetch_object ($result)) {
 echo $row->user_id;
 echo $row->fullname;
}
mysql_free_result ($result);
?>
```

0. Tiêu chuẩn của HTML

Đối với bất cứ một vấn đề gì, luôn luôn có những **luật** để tuân theo. Thật may mắn, với HTML những luật này rất ít và những gì mà HTML cung cấp thì rất là lớn...

Mục đích

Ở đây chỉ là sự giới thiệu về một vài khái niệm liên quan đến HTML. Sau bài học này bạn có thể:

- Diễn tả phần quan trọng của những tiêu chuẩn HTML.
- Nhận biết được sự khác nhau giữa HTML 2.0 và HTML 3.0

Bài học

HTML, hay**H**yper**T**ext **M**arkup **L**anguage, là một sự định dạng để Web browser hiển thị những tài liệu multimedia. Bản thân những tài liệu chỉ là những tập tin văn bản đơn giản (ASCII) với những tag hoặc những đoạn mã đặc biệt mà một Web browser biết cách thông dịch và hiển thị nó trên màn hình của bạn.

Giới thiệu về những tiêu chuẩn

World Wide Web thật là *thú vị* - Nó ở khắp mọi nơi.

Hãy nhớ rằng những thứ tạo ra Web (và Internet nói chung) là phù hợp với những luật được công nhận để hầu hết loại máy tính có thể giao tiếp và chia xẻ thông tin với nhau.

Ở đâu có thể tìm thấy những tiêu chuẩn HTML được sử dụng ở đây?

Tất cả những gì mà chúng ta sẽ học trong tài liệu này là sự trợ giúp để tạo ra những tài liệu phù hợp với HTML standards.. Khi sử dụng HTML tiêu chuẩn, các bài làm của bạn sẽ "chia xẻ được" (share-able) một cách rộng rãi trong sự phát triển nhanh chóng của mạng trong tương lai. Tiêu chuẩn được sử dụng hiện nay là HTML 2.0 được hỗ trợ bởi hầu hết các Web browser.

Hiện nay, sau khi NetScape và Microsoft giới thiệu về một vài đặc điểm được coi như là vượt xa hơn những tiêu chuẩn chung của HTML trước đó, một số đặc điểm phức tạp hơn sẽ được đưa vào phiên bản HTML 3.2. Những đặc điểm này chỉ có thể được hỗ trợ bởi một số Web browser nhất định.

Điều này có nghĩa là gì ? Nếu bạn tính đến chuyện làm cho trang Web của bạn trông thấy có sức thu hút mạnh mẽ trong Netscape hoặc Explorer, thì những trang Web này có thể trở nên có vẻ rất tồi trong những Web browser khác. Hãy nhớ rằng những người đọc trang Web của bạn có thể không những đang sử dụng một loại Web browser khác mà kích cỡ màn hình và phông chữ cũng có thể không giống như trên hệ thống mà bạn sử dụng để thiết kế các trang web.

Sau cùng, có thể không phải chỉ có một mình bạn xem những trang Web mà bạn đã mất nhiều thời gian để thiết kế ra! Điều quan trọng ở đây là làm một thứ gì đó để cho mọi người cùng xem. Vì thế phần đầu tiên của những bài học sẽ đưa bạn lướt qua những đặc điểm của HTML đã được chấp nhận rộng rãi. Từ đó, bạn có thể quyết định sử dụng những đặc điểm "tốt nhất".

Những vấn đề cần được xem lại

1. HTML là gì?
2. Tại sao ta phải quan tâm đến sự khác nhau trong những tiêu chuẩn HTML?

Đi đến bài tiếp theo....

Đã đến lúc bắt đầu viết những trang Web đầu tiên! Bạn đã sẵn sàng chưa? Trong bài học kế tiếp bạn sẽ biết làm thế nào để sử dụng ba cửa sổ để viết những dòng HTML đầu tiên. Hãy bắt đầu ...

1. Tạo tài liệu HTML đầu tiên

Bạn sắp bước vào một cuộc hành trình sẽ biến bạn từ một **Internet Surfer** trở thành một **Internet Author of Multimedia!**

Mục đích

Sau bài học này bạn sẽ có thể:

- Nhận ra ý nghĩa và mục đích của những tag HTML.
- Mở ra một Workspace cho việc tạo những tài liệu HTML.
- Sử dụng một trình soạn thảo văn bản để tạo cấu trúc HTML đơn giản cho bất kỳ trang Web nào.
- Chèn những lời chú thích không được hiển thị vào trong các tập tin HTML.
- Mở tài liệu của bạn bằng Web browser để thấy nó được hiển thị như thế nào.

Bài học

Bây giờ bạn đã biết HTML là gì, chúng ta hãy bắt đầu sử dụng nó.

Tag HTML là gì ?

Khi một Web browser hiển thị một trang chẳng hạn như trang bạn đang xem ở đây, Web browser sẽ đọc từ một file văn bản đơn giản và tìm kiếm những đoạn mã đặc biệt hay những tag được đánh dấu bởi ký hiệu < và > . Dạng chung của một tag HTML là :

```
<tag_name>string of text</tag_name>
```

Lấy ví dụ, tiêu đề của phần này sử dụng tag `header` :

```
<h3>Tag HTML la gi?</h3>
```

Tag này báo cho Web browser hiển thị dòng chữ **Tag HTML là gì?** ở dạng của mức độ tiêu đề thứ ba (chúng ta sẽ xét kỹ hơn những tag này ở những bài sau). Những Tag HTML báo cho Web browser biết khi nào cần in đậm một dòng văn bản, in ngiêng nó, làm cho nó trở thành một header, hoặc làm cho nó là một Hypertext liên kết tới một trang Web khác. Điều quan trọng cần nhớ là trong tag kết thúc,

```
</tag_name>
```

có chứa ký tự "/". Ký tự "/" này báo cho Web browser biết là hiệu ứng của tag được kết thúc tại đây. Rất nhiều tag HTML được đi theo một cặp như thế này. Nếu bạn quên ký tự "/", Web browser sẽ tiếp tục thể hiện hiệu ứng của tag trong phần còn lại của văn bản và tạo ra các kết quả không mong muốn (để có kinh nghiệm, sau này bạn có thể thử xem).
LƯU Ý: một Web Browser không quan tâm tới việc bạn sử dụng chữ hoa hay chữ thường. Lấy ví dụ : <h3>. . .</h3> thì không khác gì với <H3>. . .</H3>
Không giống như việc lập trình, nếu bạn có một lỗi trong trang HTML, hệ thống của bạn sẽ không bị "Crash"; trang Web của bạn sẽ vẫn nhìn thấy được, nhưng ... sai. Có thể nhanh chóng và dễ dàng vào bên trong trang HTML và sữa chữa lại nội dung của nó.

Browser của bạn tuy nhỏ nhưng lại có một bộ từ vựng mở. Thật là thú vị với HTML khi browser của bạn không biết làm cái gì với tag đã cho, nó sẽ bỏ qua tag đó! Lấy ví dụ, trong tài liệu mà bạn đang xem đây, tag header cho phần này *thật sự* được viết như sau :

```
<wiggle><h3>Tag HTML la gi?</h3></wiggle>
```

nhưng vì browser của bạn có thể không hỗ trợ tag `<wiggle>` (do tôi tự đặt ra, có thể trong tương lai nó sẽ làm cho văn bản có dạng sóng khi thể hiện), nó sẽ bỏ qua và vẫn tiếp tục với những tag nào mà nó hiểu được. Nếu tôi là người viết ra một web browser mới, tôi sẽ quyết định thêm tag <wiggle> vào phần mềm của tôi.

Mở ra Workspace của bạn

Để hoàn tất những bài học trong phần hướng dẫn này, bạn sẽ tạo một cửa sổ web thứ hai (điều này cho phép bạn giữ cửa sổ này với những lời chỉ dẫn của bài học và một cửa sổ như

là một vùng làm việc của bạn "Workspace"); và thêm một cửa sổ thứ ba là trình soạn thảo văn bản.
Lưu ý: Đây là nơi rất tốt để lưu ý với bạn rằng chúng tôi sẽ cung cấp những chỉ dẫn chẳng hạn như là tên thực đơn và tên tập tin nhưng những tên này có thể khác đi phụ thuộc vào Web browser mà bạn đang sử dụng.

Vì vậy, bạn hãy cố thích ứng với việc chuyển qua lại giữa các chương trình và cửa sổ trên máy của bạn. Một cách khác là hãy in ra các lời chỉ dẫn trong bài học (nhưng thật sự chúng tôi không muốn tổ chức các bài hướng dẫn này theo một cấu trúc cây). Dưới đây là các bước để tạo ra workspace của bạn :

1. Từ thực đơn **File** của web browser, chọn **New Window** hoặc **New Web Browser**. Một cửa sổ web thứ hai sẽ xuất hiện. Hãy sử dụng cửa sổ thứ nhất như là một "sách chỉ dẫn" (textbook) và cửa sổ thứ hai như là vùng làm việc của bạn (workspace) để hoàn tất những bài học HTML.

   Lưu ý: Lý do duy nhất để có hai cửa sổ ở đây là bạn có thể đọc những chỉ dẫn từ bài học và cũng có thể xem tài liệu mà bạn đang làm việc. Bạn cũng có thể đưa vào boomark trang web này và trở lại nó vào mọi lúc sử dụng thực đơn *Go* hay *History*.

2. Kế tiếp, bạn cần chuyển ra khỏi web browser và mở một chương trình soạn thảo văn bản.

   Lưu ý: Bạn cần phải chuyển qua lại giữa các cửa sổ để hoàn tất các bài học. Điều này có thể gây ra khó chịu phụ thuộc vào kích cỡ của màn hình bạn đang sử dụng. Bạn có thể chỉnh lại kích thước các cửa sổ để chúng cùng được trình bày trên màn hình để dễ dàng làm việc với chúng.

   Nếu bạn sử dụng một chương trình xử lý văn bản để tạo HTML của bạn , hãy chắc chắn rằng dạng lưu trữ là văn bản (hoặc ASCII).

Nếu bây giờ bạn mới bắt đầu, chúng tôi NHẤN MẠNH lại đề nghị bạn sử dụng một trình xử lý văn bản đơn giản như - *SimpleText* hay *TeachText* của Macintosh hay *NotePad* của Windows. Tại sao không nên sử dụng sự trợ giúp điêu luyện của các trình xử lý văn bản HTML? Thực tế sự thiết kế của tài liệu mong muốn các bạn hiểu được những khái niệm cần thiết rồi SAU ĐÓ mới sử dụng đến sự trợ giúp mà các trình xử lý văn bản HTML mang lại.

Cũng vì vậy, sẽ giúp cho bạn được nhiều hơn khi bạn tạo ra một directory/folder mới trong máy của bạn để giữ lại các kết quả bạn tạo ra. Bạn có thể gọi nó là `workarea` hay `myspace` hay bất kỳ từ nào bạn thích; hãy lưu lại tất cả các tập tin mà bạn tạo ra trong vùng này. Điều này sẽ làm cho cuộc sống của bạn tốt hơn ... ít nhất là trong khi làm việc với tài liệu này!

Tạo tài liệu HTML của bạn

Một tài liệu HTML gồm hai phần riêng biệt, phần đầu (head) và phần thân (body). Phần **đầu** chứa đựng những thông tin về tài liệu và không được hiển thị lên màn hình. Phần **thân** thì chứa đựng mọi thứ được trình bày lên màn hình như là một phần của trang Web.

Cấu trúc cơ bản của một trang HTML là:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
<html>
<head>
<!-- header info used to contain extra information about
    this document, not displayed on the page -->
</head>
```

```
<body>

<!-- all the HTML for display -->
      :    :
      :    :
      :    :
</body>
</html>
```

Dòng đầu tiên hết:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
```

thì không cần thiết lắm, nhưng là mã để báo cho browser biết phiên bản nào của HTML được sử dụng trong trang hiện thời. Để có nhiều thông tin hơn, hãy xem trang W3C Reference Specification.

Toàn bộ công việc để thành lập một trang HTML nằm trong cặp tag `<html>...</html>`. Bên trong nó là cặp tag `<head>...</head>` và sau đó là `<body>...</body>`. Trang Web của bạn có thể được hiển thị tốt trên hầu hết máy tính mà không cần những tag này. Tuy nhiên bằng cách sử dụng chúng, trang Web của bạn sẽ hoàn toàn tương thích với chuẩn quốc tế của HTML và chắc chắn rằng sẽ tương thích với những Web browser khác trong tương lai.

Đây là một thói quen tốt giống như là việc bạn đánh răng hằng ngày vậy.

Cũng lưu ý rằng những tag **chú thích** được bao bởi `<!-- blah blah blah -->`. Dòng văn bản giữa những tag này KHÔNG được hiển thị trong trang Web nhưng để đặt những thông tin có thể hữu ích cho chính bạn hoặc bất kỳ người nào cần phải xem dạng HTML của những trang Web. Khi trang Web của bạn trở nên phức tạp (giống như bạn sẽ thấy sau này khi thêm vào các bảng, frame và những vấn đề khác của hơn 20 bài học), những chú thích này sẽ trở nên hữu ích khi bạn cần cập nhật một trang Web mà bạn đã tạo ra cách đây khá lâu.

Sau đây là những bước tiếp theo cho việc tạo file HTML đầu tiên. Bạn vẫn sẵn sàng chứ?

1. Nếu chưa mở, hãy mở trình soạn thảo văn bản của bạn.
2. Chuyển đến trình soạn thảo văn bản.
3. Bạn hãy nhập những dòng văn bản sau (không cần đánh vào phím RETURN cuối mỗi dòng, web browser sẽ tự động điều chỉnh lại các văn bản):
4.
5. `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">`
6. `<html>`
7. `<head>`
8. `<title>Volcan`

2. Điều chỉnh một tài liệu HTML

Bây giờ bạn đã có tài liệu HTML đầu tiên của bạn, bạn sẽ học cách điều chỉnh tài liệu và nhìn thấy sự cập nhật trong tài liệu của bạn.

Mục đích

Sau bài học này bạn sẽ có thể:

- Mở lại workspace cho trang Web của bạn.
- Thực hiện các thay đổi trong tài liệu HTML bằng cách sử dụng trình soạn thảo.
- Nạp lại tài liệu trong Web browser để thấy được sự thay đổi.

Bài học

Mở lại workspace của bạn

Lưu ý: Nếu bạn không có tài liệu từ bài học trước, bạn có thể download lại ngay bây giờ.

Để hoàn tất bài học này bạn cần tạo nên cửa sổ thứ hai và mở lại cửa sổ soạn thảo văn bản bạn đã sử dụng trong bài học trước. Sau đây là những bước cần thiết để làm công việc đó:

1. Nếu chưa có, tạo cửa sổ mới bằng cách chọn **New Window** từ thực đơn **File** .
2. Sử dụng **Open File...** từ thực đơn **File** để tìm và mở lại tập tin HTML bạn đã tạo ra trong ví dụ trước.
3. Mở lại chương trình soạn thảo văn bản.
4. Trong chương trình soạn thảo, mở tập tin ("Volc.htm") bạn đã tạo ra trong bài học trước.

   Chú ý: Nếu bạn sử dụng Windows, tập tin của bạn nên có tên là `"VOLC.HTM"`.

   Từ nay chúng tôi xem như bạn dễ dàng mở được workspace của bạn.

Tạo sự thay đổi trong tài liệu HTML.

1. Chuyển đến cửa sổ trình soạn thảo.
2. Sau đoạn văn bản bạn đã đánh từ bài học trước, nhấn ENTER một vài lần và đánh thêm đoạn văn sau:
3.
4. A volcano is a location where magma,
5. or hot melted rock from within a planet,
6. reaches the surface. It may happen violently,
7. in a massive supersonic explosion, or more
8. quietly, as a sticky, slow lava flow.

   Lưu ý rằng đoạn văn bản này phải ở **trên** những tag `</body>` và `</html>` nằm ở cuối tập tin HTML.

9. Chọn **Save** từ thực đơn **File** để cập nhật sự thay đổi trong tập tin HTML của bạn.

Nạp lại tài liệu trong Web Browser

Trở lại browser workspace của bạn, nơi mà phiên bản trước của tập tin được hiển thị. Lưu ý rằng những dòng chữ bạn mới nhập vào chưa được nhìn thấy. Để thấy được sự thay đổi , sử dụng button **Reload** hoặc thực đơn có trong web browser của bạn. Yêu cầu này ra lệnh cho Web browser đọc lại cùng một tập tin HTML và hiển thị lại nó cùng với các sự thay đổi đã được tạo ra. Bạn sẽ nhìn thấy những dòng chữ bạn mới nhập vào.

Lưu ý rằng Web browser sẽ bỏ qua những dòng trống mà bạn đã nhập. Mặc dầu vậy, có thể bạn vẫn muốn sử dụng những dòng trắng (blank line), những khoảng trắng, và cả những tag chú thích mà chúng ta đã thấy trong bài 1 để làm cho tập tin HTML của bạn dễ đọc hơn trong trình soạn thảo văn bản.

Sử dụng Drag và Drop!

Có thể có một cách dễ dàng hơn để nạp và xem các trang HTML của bạn. Bạn cần phải sắp xếp lại trong máy của bạn để có thể nhìn thấy icon cho các tập tin HTML của bạn bên ngoài cửa sổ web browser. Sau đó click và drag icon của tập tin `"Volc.htm"` hay `"Volc.htm"`

thẳng vào cửa sổ web browser của bạn. Đúng vậy! trang của bạn sẽ được hiển thị nếu máy tính của bạn có hỗ trợ hoạt động drag và drop (Chúng tôi biết chắc rằng Macintosh System 7.5 và Windows 95 có hỗ trợ hoạt động này.)

Kiểm tra công việc của bạn

Hãy so sánh tài liệu của bạn với ví dụ mẫu sẵn có. Nếu trang Web của bạn trông khác với mẫu, hãy xem lại văn bản mà bạn đã đánh vào trong trình soạn thảo văn bản. Hãy chắc chắn rằng nó phù hợp với những chỉ dẫn trong đoạn **Tạo sự thay đổi trong tài liệu của bạn** của bài này.

Xem lại

Xem lại những chủ đề sau:

1. Làm thế nào để mở lại workspace của bạn?
2. Những bước nào được sử dụng để thay đổi nội dung tài liệu HTML của bạn?
3. Làm thế nào hiển thị và nhìn thấy sự thay đổi trong Web browser?

Thực tập tự do

Tương tự như trong bài học, hãy điều chỉnh tài liệu HTML riêng của bạn đã tạo ra trong bài học vừa rồi. Thêm một vài câu và hãy kiểm tra trong browser của bạn khi nạp lại nó.

Đi đến phần tiếp theo....

Bây giờ bạn đã biết về quá trình soạn thảo, chúng ta sẽ thêm những tiêu đề to và có sức hấp dẫn hơn vào tài liệu HTML của bạn

3. Sáu mức tiêu đề

Như là bạn đã nhìn thấy trong trang Web này, các phần tiêu đề ("Sáu mức tiêu đề", "Mục đích", "Mục Lục", ...) xuất hiện ở những kích cỡ khác nhau và, có lẽ ở cả những phông chữ và màu sắc khác nhau nữa. HTML cung cấp những đoạn mã cho việc thiết kế phần tiêu đề theo sáu mức độ khác nhau. Browser tự xác định chính xác phông chữ và kích cỡ để hiển thị.

Mục đích

Sau bài học này bạn có khả năng:

- Nhận dạng được những mức độ khác nhau của các tiêu đề trong HTML và các tag liên quan tới chúng.
- Đặt những mức tiêu đề khác nhau vào trong tài liệu HTML và nhìn thấy sự thay đổi của chúng trong browser của bạn.

Bài học

Những tiêu đề của HTML

Những tiêu đề được thực hiện trong HTML bằng cách đặt đoạn văn bản giữa những tag tiêu đề (heading tag). Dạng các tag tiêu đề của HTML là:

<hN>Text to Appear in Heading</hN>

N là một số có giá trị từ 1 đến 6 chỉ định kích cỡ tiêu đề. Sau đây là một vài ví dụ cho những kích cỡ khác nhau của tiêu đề :

Heading Level 1

Heading Level 2

Heading Level 3

Heading Level 4

Heading Level 5

Heading Level 6

Các mức độ tiêu đề được sắp xếp từ 1 (quan trọng nhất) đến 6 (ít quan trọng nhất). Các tiêu đề nên có tính tương đối, đánh số thứ tự và đặt tại các vị trí tương tự nhau.

Đặt những tiêu đề HTML vào tài liệu của bạn

Lưu ý: Nếu bạn không có tài liệu từ bài học trước, bạn có thể download lại ngay bây giờ.

1. Mở lại workspace của bạn (nếu nó chưa được mở).
2. Đến cửa sổ soạn thảo văn bản.
3. Mở lại tập tin HTML đã tạo trong bài 2, `"Volc.htm"`.
4. Trước hết, chúng ta sử dụng tag <h1> để trình bày tiêu đề như là header lớn nhất, `<H1>`. Nhập dòng sau vào phần thân của HTML và **sau** các tag `</head><body>`:
5.
6. <h1>Volcano Web</h1>
7. Bên dưới những dòng văn bản đã nhập, tạo những đề mục khác nhau cho những phần sau này của trang *Volcano Web* của bạn.

   Nhập những tiêu đề sau trong phần thân của trang Web của bạn (chú ý một số sử dụng </h3> còn một số sử dụng </h2>):

   <h2>Introduction</h2>

   <h2>Volcano Terminology</h2>

   <h2>Volcanic Places in the USA</h2>

   <h3>Mount St Helens</h3>

   <h3>Long Valley</h3>

   <h2>Volcanic Places on Mars?</h2>

   <h2>Research Project</h2>

   <h3>References</h3>

8. Lưu trữ (save) lại sự thay đổi trong trình soạn thảo.
9. Trở lại web browser của bạn, **Mở** và **Nạp lại** tập tin HTML.

Chú ý rằng trên máy tính bạn đang sử dụng hiện nay, bạn có thể cài đặt cho browser của bạn phông và/hay kích thước của các tiêu đề. Ví dụ như, bạn có thể có một browser thể hiện tag *h1* theo phông Times có 36 điểm; tag *h2* theo phông Helvetica có 24 điểm, v.v... Các mã HTML chỉ đưa ra các loại của tiêu đề (từ h1 đến h6); còn việc thể hiện chúng thì được điều khiển bởi người sử dụng thông qua trình web browser.

Kiểm tra công việc của bạn

So sánh công việc của bạn với ví du mẫu sẵn có. Nếu có một vài tiêu đề xuất hiện không đúng, hãy kiểm tra lại tag bắt đầu và tag kết thúc của mỗi tiêu đề.

Như là một phần bài tập đề nghị, hãy nhìn xem chuyện gì sẽ xảy ra khi bạn có một lỗi đánh máy sai. Mở lại tài liệu HTML của bạn trong chương trình soạn thảo và xoá đi ký tự "/" trong tag kết thúc <h1> , sau đề mục **Volcano Web**:

```
    <h1>Volcano Web<h1>
[missing "/" ----------^^^]
```

Lưu trữ sự thay đổi và nạp lại trong Web browser. Không có sự kết thúc đúng đắn của tag h1, browser của bạn sẽ thông dịch phần văn bản tiếp theo như là một phần của tiêu đề! Sau khi thử xong bạn nên trở lại tài liệu của bạn và thêm lại dấy / như ban đầu.

Xem lại

1. Những mức độ khác nhau của tiêu đề trong HTML là gì?
2. Những tag liên quan đến những mức độ này là gì?
3. Những bước nào được sử dụng trong việc đặt tiêu đề vào trong tài liệu HTML ?
4. Những gì xảy ra nếu bạn quên dấu "/" tại cuối của một tag tiêu đề?

Thực tập tự do

Hãy thêm vào ít nhất ba tiêu đề với các cấp độ khác nhau vào tài liệu WWW của riêng bạn.

4. Chia văn bản ra thành nhiều đoạn

Cho đến bây giờ bạn đã tạo và sửa đổi tài liệu HTML, và chắc bạn đã cảm thấy thoải mái với việc soạn thảo và nạp lại văn bản trong Web browser. Vì vậy bây giờ bạn sẽ cảm thấy dễ dàng với bài học **nhanh** này: thêm vào các dấu chia đoạn văn (paragraph break).

Mục đích

Sau bài học này bạn có thể:

- Nhận dạng những tag chia đoạn trong HTML.
- Chép một đoạn văn bản từ Web page và dán (paste) nó vào một tài liệu khác.
- Chèn các dấu chia đoạn vào trong văn bản HTML và nhìn sự thay đổi trong Web browser của bạn.

Bài học

Chia đoạn trong HTML

Trước đây chúng ta đã biết rằng một Web browser sẽ bỏ qua tất cả ký tự XUỐNG DÒNG (carriage returns) được đánh trong trình soạn thảo. Nhưng, khi browser nhìn thấy tag chia đoạn, nó chèn một dòng trống và bắt đầu một đoạn mới. Mã HTML cho công việc chèn một sự chia đoạn là :

```
<p>
```

Lưu ý rằng tag này là một đặc biệt vì nó không cần tag kết thúc; vì thế bạn không cần sử dụng:

```
</p>
```

Trong bài sau chúng ta sẽ thấy trường hợp tag có sử dụng tag để kết thúc.

Cũng lưu ý rằng tag <h> có sự gắn liền với sự chia đoạn nên không cần thiết đặt tag trước một tag tiêu đề như ví dụ dưới đây:

```
<p>
<h2>Blah Blah Blah Blah</h2>
```

Chèn các dấu chia đoạn

Lưu ý : Nếu bạn không có tài liệu từ bài học trước, bạn có thể download lại ngay bây giờ.

Hãy theo sự chỉ dẫn dưới đây để chèn và xem lại một sự ngắt đoạn trong tài liệu HTML của bạn.

1. Mở lại workspace của bạn (nếu nó chưa được mở).
2. Đi đến cửa sổ soạn thảo văn bản.
3. Mở tài liệu làm việc của bạn `Volc.htm` trong trình soạn thảo văn bản (nếu nó chưa được mở).
4. Trước hết chúng ta di chuyển câu (`"A volcano is ..."`) đến dưới tiêu đề **Introduction**. Hãy sử dụng chuột để **cắt** và **dán** đoạn văn bản này vào vị trí thích hợp.
5. Sau những câu này, chúng ta muốn thêm vào một đoạn nữa. Nhưng thay vì phải đánh lại nó, từ trang Web này hãy sử dụng chuột để chọn và **chép** (copy) vào clipboard các đoạn văn:
6.
7. Volcanoes have been a part of earth's history
8. long before humans. Compare the history of human
9. beings, a few million years in the making, to that of
10. the Earth, over four billion years in the making.
11.
12. Bây giờ, hãy trở lại tài liệu HTML trong trình soạn thảo và **dán** (paste) nó vào sau những câu có sẵn bên trong đề mục `<h2>Introduction</h2>`.
13. Lưu trữ sự thay đổi trong trình soạn thảo.
14. Trở lại Web browser của bạn.
15. Nếu tài liệu của bạn chưa có, sử dụng lệnh **Open Local...** từ thực đơn **File** để mở tài liệu.
16. Chọn **Reload** từ thực đơn **File**. Bây giờ bạn sẽ nhìn thấy hai câu của phần Introduction. Bây giờ chúng ta muốn đặt một dấu phân đoạn *giữa* các câu này.
17. Một lần nữa trở lại tài liệu HTML trong trình soạn thảo.
18. Sau câu thứ hai của phần `<h2>Introduction</h2>` (tức là câu có kết thúc là " as a sticky, slow lava flow.") nhấn RETURN (thật ra không cần thiết nhưng điều này làm cho bản HTML dễ đọc hơn khi bạn làm việc với nó), và bây giờ hãy nhập tag *phân đoạn* như sau :
19.
20.

    Lúc này đoạn văn bản trông giống như sau:

```
<h2>Introduction</h2>
A volcano is a location where magma,
or hot melted rock from within a planet, reaches the
surface. It may happen violently, in a massive supersonic
explosion, or more quietly, as a sticky, slow lava flow.
<p>
Volcanoes have been a part of earth's history long before
humans. Compare the history of human beings, a few million
years in the making, to that of the Earth, over four
billion years in the making.
```

21. Lưu trữ sự thay đổi trong trình soạn thảo.
22. Trở lại Web browser và **nạp lại** tài liệu. Hai câu của mục introduction bây giờ sẽ ở trên hay đoạn văn khác nhau.

Những dạng phân đoạn khác

Để phân chia những phần chính của một trang Web, sử dụng `hard rule` hay còn gọi là tag `hr` . Tag này sẽ chèn một đường thẳng giống như bạn nhìn thấy ngay trên phần tiêu đề của mục này.

Dạng HTML cho tag `hard rule` là:

```
<hr>
```

Bây giờ chúng ta hãy sử dụng nó! Đặt một tag `hr` ngay trên tiêu đề **Introduction**. Công việc này giúp cho việc tách câu mở đầu của bài học ra khỏi những phần đi sau nó.

Và cuối cùng, tag <br> sẽ đẩy văn bản xuống dòng mới như tag <p> nhưng không chèn thêm một dòng trống. Bạn có thể phải sử dụng tag này khi tạo một danh sách (list), khi viết những dòng của một bài thơ, v.v... Hãy so sánh sự khác nhau giữa việc sử dụng tag <br> và tag <p> trong hai ví dụ sau đây:

Chỉ sử dụng các tag <p>	
HTML	**Kết quả**
And then, we could all see at once the brilliant purpose of the paragraph tag. **<p>** Moving on... **<p>** the more tags you write, the better you will feel?	And then, we could all see at once the brilliant purpose of the paragraph tag. Moving on... the more tags you write, the better you will feel?
**Sử dụng cả tag <p> và tag **	
HTML	**Kết quả**
And then, we could all see** ** at once the brilliant purpose** ** of the paragraph tag. **<p>** Moving on... ** ** the more tags you	And then, we could all see at once the brilliant purpose of the paragraph tag. Moving on... the more tags you write, the better you will feel?

write,**
**
the better you will feel?

Tag <br> còn có thể được sử dụng cho những cách trình bày khác nhau cho phần tiêu đề của bạn. Nếu như bạn chú ý, bạn sẽ nhận thấy là các tag **tiêu đề** <h1>, <h2>, . . . tự động chèn thêm các dòng trắng trên và dưới dòng chữ của tag tiêu đề. Một vài tác giả của những trang Web thích tự điều khiển những dòng trắng này.

Tên đề mục dùng tag tiêu đề	
HTML	**Kết quả**
and in the end it was all for naught. ** **header tag** ** Later, sir Longhorn declared that all makers of cheese would have to be certified before commencing production.	and in the end it was all for naught. **The New Cheese Edict** Later, sir Longhorn declared that all makers of cheese would have to be certified before commencing production.

Tên đề mục dùng tag và 	
HTML	**Kết quả**
and in the end it was all for naught. **<p>** ****The New Cheese Edict** ** Later, sir Longhorn declared that all makers of cheese would have to be certified before commencing production.	and in the end it was all for naught. **The New Cheese Edict** Later, sir Longhorn declared that all makers of cheese would have to be certified before commencing production.

Ở đây sự khác nhau có vẻ tầm thường quá, nhưng thực tế điều này tạo ra các khả năng để sau này chúng ta học cách tạo ra tiêu đề có màu và kích cỡ khác nhau. Ví dụ như hãy xem trang Web Pages That Don't Look Like Web Pages

Kiểm tra lại công việc của bạn

Nếu bạn thích, hãy so sánh trang Web bạn đang thực hiện với ví dụ mẫu. Nếu tài liệu bạn tạo ra khác với ví dụ mẫu, hãy kiểm tra lại cách bạn đánh vào các dấu chia đoạn <p>. Hãy chắc chắn rằng bạn đã làm đúng theo sự hướng dẫn trong mục **Chèn các dấu chia đoạn** của bài này.

Xem lại

1. Tag HTML nào dùng cho việc phân đoạn?
2. Những bước nào bạn sử dụng cho việc chèn thêm một sự phân đoạn trong tài liệu của bạn?
3. Làm thế nào trình bày và nhìn thấy được sự thay đổi đó trong Web browser ?
4. Điểm thêm*- Tag <hr> là gì? Tag <br> là gì?

Thực tập tự do

Sử dụng `paragraph`, tag `hard rule`, hay tag `br` để tạo ra các đề mục hay đoạn văn trong tài liệu của bạn.

Đi đến phần tiếp theo....

Bạn có nhớ phông chữ đầu tiên không ?
Làm thế nào để thêm một **kiểu mẫu...** (style) khác vào văn bản của bạn.

Giống như một trình soạn thảo văn bản, HTML có thể báo cho Web browser hiển thị một phần nào đó của văn bản ở dạng in *nghiêng* hoặc in **đậm** hoặc ở dạng in **kết hợp** cả hai chế độ.

Mục đích

Sau bài học này bạn có khả năng:

- Nhận dạng các tag HTML cho các kiểu mẫu văn bản: **Đậm**, *nghiêng*, hoặc ở dạng chữ đánh máy (`mono-spaced`).
- Nhập văn bản trong tài liệu HTML theo nhiều kiểu mẫu khác nhau và nhìn thấy sự thay đổi của chúng trong Web browser của bạn.

Bài học

Các tag Style của HTML

HTML cung cấp một vài tag cho việc định dạng style cho văn bản. Hãy thận trọng và suy nghĩ cẩn thận trước khi dùng những kiểu này để định dạng cho văn bản; sử dụng quá nhiều có thể làm cho văn bản trở nên khó đọc hơn ...

Các tag Style	
HTML	**Kết quả**
`<b>This is Bold...</b>`	**This is Bold**
`<i>This is Italic...</i>`	*This is Italic*
`<tt>This is Typewriter...</tt>`	This is Typewriter

Lưu ý rằng bạn có thể kết hợp các tag này để định dạng miễn là chúng được lồng vào nhau một cách chính xác, ví dụ như cả hai tag bắt đầu và kết thúc in nghiêng đều phải *ở bên trong* các tag in đậm. Cũng lưu ý là thứ tự các tag không quan trọng.

`<i><b>This is Bold AND Italic</b></i>`	***This is Bold AND Italic***
`<b><i>And So is This</i></b>`	***And So is This***

Hơn nữa, bạn có thể áp dụng những kiểu vào trong phần tiêu đề. Lưu ý về cách đặt các tag style đóng và mở xung quanh những từ muốn tạo kiểu còn những tag đề mục thì bao quanh toàn bộ đề mục.

`blah blah blah`	blah blah blah
`<h2><i>New</i> and <tt>Improved!</tt></h2>`	***New* and `Improved!`**
`blah blah blah`	blah blah blah

Áp dụng *Style* vào trong tài liệu của bạn

Lưu ý: Nếu bạn chưa có tài liệu từ các bài học trước, bạn có thể download ngay bây giờ.

Theo những bước sau để áp dụng các tag style cho văn bản HTML của bạn:

1. Mở lại workspace của bạn (nếu nó chưa được mở).
2. Trở lại tài liệu HTML "`Volc.htm`" của bạn trong chương trình soạn thảo.
3. Tìm từ "volcano" trong câu đầu tiên của phần Introduction. Chúng ta sẽ làm từ này trở nên in đậm để đánh dấu đây là từ quan trọng.
4. Chèn thêm các tag để làm từ này trở nên in đậm :
5.
6. volcano
7. Bây giờ chúng ta sẽ sửa đổi đoạn văn thứ hai với những tag in đậm và in ngiêng để nhấn mạnh. Hãy thêm tag ... và ... quanh từ `billion` đó để đoạn này trông giống như sau :
8.
9. <p>
10. Volcanoes have been a part of earth's history long
11. before humans. Compare the history of human beings,
12. a few million years in the making, to that of the
13. Earth, over four billion years in the
14. making.
15. Cuối cùng, chúng ta sẽ sử dụng tag `typewriter` để chỉ ra một từ đặc biệt. Bên dưới đề mục **Volcano Terminology**, nhập vào đoạn sau:
16.
17. The study of volcanoes, or <tt>Volcanology</tt>,
18. includes many odd terms.
19. **Lưu trữ** trong trình soạn thảo và **Reload** trong Web browser của bạn.

Kiểm tra lại công việc của bạn

Hãy xem Ví dụ mẫu để nhìn thấy sự thay đổi. Điều quan trọng khi sử dụng các tag style là sự kết thúc đúng đắn các tag này bằng tag </> thích hợp. Nếu không, tất cả phần văn bản còn lại sẽ bị tác động của kiểu đã chọn. Khi nhìn sẽ có vẻ kỳ quặc.

Xem lại

1. Các tag style của HTML là gì?
2. Những tag nào được sử dụng cho việc định dạng các kiểu mẫu văn bản?
3. Những bước nào bạn sử dụng cho việc áp dụng kiểu vào trong tài liệu HTML ?
4. Điểm thêm*- Các tag này tạo ra tiện lợi khi tạo trang web như thế nào?

Thực tập tự do

Thử sử dụng các tag style để **làm đậm**, *làm nghiêng*, và dùng kiểu chữ đánh máy `typewriter` vào văn bản trong trang web của bạn. Hãy quan sát khi bạn có thể sử dụng thành công tổ hợp các kiểu mẫu.

6. Danh sách, Danh sách, và Danh sách

Danh sách được dùng để trình bày các mục thông tin thành một dạng dễ đọc hơn. Chẳng hạn như, ngay sau phần tiêu đề kế tiếp dưới đây có một danh sách.

Mục đích

Sau bài học này bạn có khả năng:

- Nhận dạng được mã cho việc tạo thành những danh sách có thứ tự (ordered list), không có thứ tự (unordered list) và danh sách lồng nhau (nested list) của một trang web.
- Đặt những loại danh sách khác nhau vào trong tài liệu HTML và nhìn thấy sự thay đổi của chúng trong trang Web.

Bài học

Có rất nhiều trang Web trình bày danh sách các mục. Có những mục bắt đầu bằng một "bullet" (unordered) hoặc là một danh sách có đánh số theo thứ tự (ordered). Những danh sách này được định dạng dễ dàng trong HTML, và chúng cũng có thể được lồng vào nhau (danh sách trong danh sách) tạo thành một dạng phân cấp tài liệu. Danh sách cũng được sử dụng cho việc tạo một bảng chỉ mục hay là một bảng nội dung của một dãy các tài liệu hoặc các chương.

Danh sách không có thứ tự

Danh sách không có thứ tự (unordered list) sử dụng tag `ul` được xuất hiện như là một danh sách các mục với những dấu "bullet" hoặc những ký hiệu đánh dấu ở trước. Ký hiệu dùng để đánh dấu tùy thuộc vào từng loại phiên bản của Web browser và phông để trình bày các ký tự thông thường (ví dụ như, trong Macintosh các bullet là ký tự option-8 của phông Times có dạng là một hình vuông nhỏ, còn trong Geneva nó lại là một dấu chấm tròn lớn).

Sau đây là ví dụ cho một danh sách không có thứ tự:

My Unordered List:

- Item 1
- Item 2
- Item 3

Và dạng HTML cho kết quả trên là:

```
<B>My Unordered List:</B>
<ul>
<li>  Item 1
<li>  Item 2
<li>  Item 3
</ul>
```

Các tag <ul> đánh dấu sự bắt đầu và kết thúc của danh sách, còn tag <li> chỉ ra từng mục cho một danh sách.

Danh sách có thứ tự

Danh sách có thứ tự là danh sách mà browser sẽ đánh số thứ tự cho mỗi mục trong danh sách, thường là bắt đầu bằng "1." Lưu ý rằng, sự khác nhau đó chính là do một thay đổi nhỏ từ tag `ul` thành tag `ol`. Lấy ví dụ tương tự như ví dụ trên:

My Ordered List:

1. Item 1

2. Item 2
3. Item 3

Và dạng HTML cho ví dụ trên là:

```
<B>My Ordered List:</B>
<ol>
<li>  Item 1
<li>  Item 2
<li>  Item 3
</ol>
```

Danh sách lồng nhau

Danh sách có thứ tự và danh sách không có thứ tự có thể lồng vào nhau theo nhiều mức độ khác nhau tùy thuộc vào từng loại Web browser. Vấn đề bạn cần quan tâm chính ở đây là việc kiểm tra rằng mỗi danh sách được kết thúc một cách chính xác và trật tự lồng vào nhau là đúng.

Bước đầu bạn có vẻ thấy phức tạp với các tag <ol>   , nhưng hãy cố gắng nhớ cấu trúc cơ bản như sau :

```
<ul>        <ol>
  <li>        <li>
  <li>        <li>
</ul>       </ol>
```

Sau đây là một ví dụ với một danh sách không có thứ tự với các mức danh sách con khác của nó:

Nested Unordered List

- This is the first item
- This is the second item
  - This is the first sub item of the second item
    - And this is a sub item of a sub item
    - Getting lost yet?
  - This is the second sub item of the second item
  - This is the third sub item of the second item
- This is the third item

Hãy chú ý sự thay đổi của các dấu đánh dấu cho các mức khác nhau của danh sách.

Và đây là đoạn mã HTML cho ví dụ trên:

```
<ul><B>Nested Unordered List</B>
<li>This is the first item
<li>This is the second item
<ul>
<li> This is the first sub item of the second item
<ul>
<li> And this is a sub item of a sub item
<li> Getting lost yet?
</ul>
<li> This is the second sub item of the second item
```

```
<li> This is the third sub item of the second item
</ul>
<li>This is the third item
</ul>
```

Những danh sách lồng nhau - Trộn chúng lại với nhau

Bạn không những có thể lồng các danh sách có thứ tự vào trong danh sách có thứ tự mà còn có thể trộn lẫn các loại danh sách. Đừng quá ngạc nhiên! Bắt đầu HTML trông có vẻ hơi ngớ ngẩn, nhưng bạn hãy xem thử cách mà các danh sách lại có những danh sách khác bên trong của nó.

Lấy ví dụ, danh sách có thứ tự dưới đây bao gồm một danh sách không có thứ tự bên trong nó:

Nested Unordered List

1. This is the first item
2. This is the second item
    - This is the first sub item of the second item
        1. An this is a numbered sub item of a sub item
        2. An this is another numbered subItem of a sub item
        3. Getting lost yet?
    - This is the second sub item of the second item
    - This is the third sub item of the second item
3. This is the third item

Và đây là đoạn mã HTML cho ví dụ trên. Hãy chú ý sự sắp xếp trong cách viết HTML để dễ đọc hơn:

```
<B>Nested Unordered List</B>
 <ol>
   <li>This is the first item
   <li>This is the second item
     <ul>
     <li> This is the first sub item of the second item
       <ol>
       <li> An this is a numbered subItem of a sub item
       <li> An this is another numbered sub item of a sub item
       <li> Getting lost yet?
       </ol>
     <li> This is the second sub item of the second item
     <li> This is the third sub item of the second item
     </ul>
   <li>This is the third item
 </ol>
```

Đặt những danh sách vào trong tài liệu HTML của bạn

Lưu ý: Nếu bạn chưa có tài liệu từ bài học trước, hãy download ngay bây giờ.

Sử dụng tag `list`, bạn sẽ thêm một danh sách có thứ tự và một danh sách không có thứ tự vào trang web *Volcano Web* của bạn.

1. Mở lại workspace của bạn (nếu nó chưa được mở).
2. Mở lại tài liệu HTML trong trình soạn thảo.

3. Bên dưới đề mục `Volcano Terminology` chúng ta sử dụng một danh sách không có thứ tự để trình bày một số thuật ngữ kỹ thuật được sử dụng trong việc nghiên cứu núi lửa. Hãy đến mục này trong tài liệu HTML của bạn.
4. Trước hết thêm câu sau :
5.
6. How many do you know?
7. Và bây giờ, hãy đánh vào dạng HTML để tạo ra danh sách các thuật ngữ:
8.
9.
10. caldera
11. vesicularity
12. pahoehoe
13. rheology
14. lahar
15.
16. Bây giờ chúng ta sử dụng một danh sách có thứ tự để xác định những phần yêu cầu của bài học chúng ta đang xây dựng. Bên dưới đề mục `Research Project`, thêm những dòng sau (Hướng dẫn: lúc này bạn nên sử dụng chức năng cắt và dán từ trong trang web trừ phi bạn có thích thú và muốn tự đánh lại đoạn văn bản!):
17.
18. Your mission is to find information and report on a
19. volcano, other than the ones listed above, that has erupted
20. in the last 100 years. Your reports must include:
21.
22. Type of volcano
23. Geographic location
24. Name, distance, and population of nearest major city
25. Dates of most recent and most destructive eruptions.
26. Other events associated with the recent eruptions
27. (earthquakes, floods, mudslides, etc)
28.
29. <p>
30. Then, write a one page description on the major hazards to
31. humans in the vicinity of this volcano. Speculate on what
32. you would do if you were in charge of minimizing the risk
33. to the population.
34. **Lưu trữ** và **Reload** trong Web browser của bạn.

Kiểm tra lại công việc của bạn

Có thể bạn muốn so sánh tài liệu của bạn với ví dụ mẫu cho phần bạn vừa làm. Nếu danh sách của bạn khác với danh sách trong ví dụ mẫu, hãy kiểm tra lại những gì bạn đã đánh trong trình soạn thảo. Hãy chắc chắn rằng nó phù hợp với các hướng dẫn trong mục **Đặt những danh sách vào trong tài liệu HTML của bạn** trong bài này.

Xem lại

1. Danh sách có giá trị như thế nào trong trang Web ?
2. Tag HTML nào được dùng cho danh sách không có thứ tự?
3. Tag HTML nào được dùng cho danh sách có thứ tự?
4. Phải làm thế nào để tạo ra danh sách lồng nhau ?
5. Những bước nào được sử dụng trong việc thêm một danh sách vào tài liệu HTML của bạn?

Thực tập tự do

Có thể bạn muốn có kinh nghiệm trong việc chuyển đổi một danh sách có thứ tự thành một danh sách không có thứ tự. Đồng thời bạn cũng muốn thêm một danh sách có thứ tự hay không có thứ tự vào trong tài liệu WWW của bạn. Hãy chắc chắn rằng nó được thể hiện đúng trong browser của bạn. Bạn có tạo được các danh sách có danh sách con không ?

7. Thêm hình ảnh vào trang Web

Việc gởi một văn bản thông qua Internet chỉ là một cách cũ của thư tín điện tử. Người ta đã làm được điều đó trong nhiều thập kỷ! Khi bạn có thể bao gồm cả **hình ảnh**, thông báo của bạn sẽ trở nên nhiều thông tin hơn. *Có phải là một bức hình sẽ đáng giá hơn cả ngàn từ không ?*

Mục đích

Sau bài học này, bạn có khả năng:

- Nhận ra những dạng hình ảnh cho World Wide Web.
- Thảo luận về những điểm chính để cân nhắc khi chèn hình ảnh vào tài liệu WWW.
- Download một tập tin hình ảnh về hệ thống của bạn.
- Sử dụng đúng những dạng HTML cho việc chèn hình ảnh trong trang web của bạn.

Bài học

Bạn sẽ cảm thấy thoải mái với bài học này! Bài này chỉ giới thiệu sơ qua về hình ảnh cho WWW. Nhưng chúng tôi muốn bạn hoạt động một chút.

Các dạng hình ảnh của Web

Có vô số dạng tập tin hình ảnh cho máy tính: PICT, GIF, TIFF, rồi thì EPS, BMP, PCX, JPEG ... Cách đọc của nó giống như một bài thơ bí ẩn. Một bài thơ tồi.

Cách thức để cho Web browser hiển thị hình ảnh là một sự định dạng của HTML để chỉ ra vị trí của tập tin hình ảnh sao cho sự định dạng đó có thể thông dịch được trong các máy tính khác nhau. Ví dụ như, thông tin theo dạng đó được nhận bởi máy Macintosh thì browser sẽ hiển thị hình ảnh theo dạng hình ảnh của máy Macintosh. Tuy nhiên, khi cũng thông tin đó gởi cho browser trên Windows, nó được hiển thị theo chế độ đồ họa của Windows.

Theo thuật ngữ kỹ thuật, chúng ta sẽ nói rằng dạng hình ảnh là **độc lập với platform** (platform independent). Bản thân HTML cũng là độc lập với platform vì các ký tự văn bản trơn của nó đều có thể hiểu được trong mọi máy tính.

Dạng chuẩn có thể hiển thị **trong** một trang web là GIF hay Graphics Interchange Format. GIF nén các thông tin của hình ảnh (giảm kích thước của tập tin) và chuyển nó thành mã nhị phân để có thể truyền qua Internet. Kỹ thuật nén theo GIF là kỹ thuật nén có hiệu quả cao nhất khi hình ảnh có những vùng liên tục có cùng một màu, và việc nén càng lớn hơn khi có sự lặp lại màu theo chiều đứng. Dạng hình ảnh khác sử dụng trong web là JPEG (được đặt tên sau khi Joint Photographic Engineering Group thiết kế ra dạng này). Trước đây, các ảnh JPEG **không** được hiển thị trong cùng trang mà được hiển thị trong một cửa sổ khác nhờ vào một chương trình "trợ giúp" được cài đặt thêm vào. Nhưng hiện nay hầu hết các browser đều hỗ trợ việc hiển thị ảnh JPEG ngay trong trang web.

Kỹ thuật nén JPEG rất có hiệu quả với các ảnh chụp mà màu sắc thay đổi liên tục trong các phần rất nhỏ của ảnh (các ảnh "grainy"). Tùy thuộc vào sự thỏa hiệp với chất lượng ảnh, JPEG cung cấp các loại hệ số nén về kích thước thật là có ấn tượng, đôi khi đạt đến 10 (tức là tập tin 1500KB giảm còn 150KB).

Để có nhiều thông tin hơn về các dạng tập tin này, hãy xem trang SITO tại Graphics File Compression. Nếu bạn muốn có một cuốn sách lớn, hãy xem Designing Web Graphics của Lynda Weinman.

Rất nhiều chương trình xử lý đồ thị có sẵn chức năng lưu trữ tập tin theo dạng GIF. Nếu không, dưới đây là một vài chương trình/trình tiện ích shareware để thực hiện việc chuyển đổi. Bạn có thể tìm thấy chúng tại các site như shareware.com, Jumbo , hay Yahoo.

Vài trình tiện ích cho Đồ Họa	
Macintosh	**Windows**
• GIFConverter • GraphicConverter • clip2gif • PhotoGIF (plugin for PhotoShop)	• WinGif 1.4 • Lview 3.1 • PaintShop Pro

Vài điểm cần biết khi sử dụng Đồ họa

Với tài liệu này, bạn không cần thiết phải sử dụng một trong các chương trình xử lý đồ họa nói trên. Chúng tôi sẽ trình bày đến các bạn cách để lưu trữ lại một bản sao của bất kỳ hình ảnh nào bạn nhìn thấy trên trang web.

Tuy vậy, vì bạn bắt đầu phát triển các trang web của bạn, bạn cũng nên làm quen với cách tạo ra các hình ảnh ở dạng GIF hay JPEG. Nếu trang web của bạn có thêm hình ảnh, bạn nên chú ý các vấn đề sau đây:

- Tăng số lượng và kích thước của hình ảnh có thể xem tốt hơn trên máy tính, nhưng chúng cũng làm cho người sử dụng phải chờ đợi hình ảnh được gởi thông qua mạng (cũng giống như có được bao nhiêu người trong chúng ta có máy Sun SparcStation tại nhà?) Chúng tôi đề nghị nên giữ kích thước của hình ảnh ít hơn 100K (với chúng tôi thì ít hơn 50B). Càng nhỏ càng tốt.
- Cũng với một ghi chú tương tự, không phải mọi người trong chúng ta đều có màn hình 21 inch! Nên giữ các hình ảnh không rộng hơn 480 điểm và không cao hơn 300 điểm để tránh cho người sử dụng phải scroll hay định lại kích thước cửa sổ browser của họ.
- Nhiều cấp độ màu làm cho ảnh có vẻ đẹp hơn nhưng với các hình GIF chúng sẽ không nén được nhiều như các khối cùng màu và đôi khi chúng không ghép nhóm được
- Một số chương trình xử lý hình ảnh có chế độ "no dithering" khi chuyển sang dạng GIF -- việc này có thể làm giảm "sự hỗn độn" (noise) trong các khối nền.
- Nhiều tông màu tối trên máy Macintosh không thể hiển thị được trên máy Windows.
- Thay vì hiển thị tất cả các hình ảnh trong trang web, có thể tạo liên kết chúng như là các hình ảnh bên ngoài để chúng chỉ được nạp về khi người xem click vào hypertext liên kết (điều này được xét kỹ hơn sau này trong bài 8a và trong bài 8d). Nếu bạn có nhiều hình cần trình bày, hãy thử chia nhỏ trang web của bạn thành một chuỗi các trang liên kết nhau.
- Một hình ảnh nhỏ (như là một "bullet" nhỏ) có thể xuất hiện nhiều lần trong một trang mà chỉ bị chậm rất ít thời gian cho mỗi lần sử dụng lại ảnh đó.
- Hiện nay nhiều browser "cache" được các hình ảnh (cất chúng lại trong máy của bạn), điều này có nghĩa là khi sử dụng cùng một tập tin trong nhiều trang web sẽ làm cho chúng được nạp vào từ ngay trên máy tính của chính bạn chứ không phải từ trên Internet.
- Điều quan trọng nhất, hãy chắc chắn rằng các hình ảnh là những thứ **tạo thêm** nghĩa cho tài liệu HTML của bạn.

Bạn có thể thiết kế ra một trang web đẹp, có nhiều hình ảnh lớn, được nạp dễ dàng từ máy tính của bạn, nhưng mà nó lại có thể chịu cảnh chậm thê thảm với các người xem sử dụng một modem tốc độ thấp qua một mạng rất bận rộn. Mạng là chỗ lúc nào cũng bị bận.

Lưu trử và Thêm Hình ảnh vào Trang Web của Bạn

Để đến được bài kế tiếp, đầu tiên bạn cần chép lại một bản sao hình ảnh của sự bùng nổ của núi lữa (hãy xem các dung nham nóng bỏng của nó!).

Chỉ cần theo các hướng dẫn trong Trung tâm Download Hình ảnh của Bài 7 rồi sau đó trở lại đây để hoàn tất bài học này

---

Kiểm tra lại công việc của bạn

Hãy kiểm tra để biết rằng tập tin `"lava.gif"` được cất vào cùng một directory/folder với tập tin văn bản HTML của bạn `"Volc.htm"`. Nếu nó không có, hãy kiểm tra lại bạn có cất nhầm vào một directory/folder khác không. Nếu đúng như vậy, chuyển nó vào lại đúng vị trí.

Xem lại

1. Hai dạng ảnh nào được sử dụng cho World Wide Web?
2. Làm thế nào để các tập tin hình ảnh được trình bày trên các máy tính khác nhau?
3. Vài điểm lưu ý nào cần xét đến khi thêm hình ảnh vào các trang web?
4. Làm thế nào để bạn cất lại hình ảnh dung nham để sử dụng trong tài liệu WWW của bạn?

Thực tập tự do

Lướt trên web và tìm một vài hình ảnh. Thử download ít nhất một hình ảnh có thể hữu ích cho trang của bạn. Vài nơi bạn có thể thử là: Yahoo Picture Archive, SpriteLib, hay Rob's Multimedia Lab

7a. Hình ảnh Inline

Toán học trong WWW là:

Text + Pictures = Multimedia
Multimedia + WWW = Global HyperMedia

Bạn có đồng ý không?

Mục đích

Sau bài học này, bạn sẽ có thể:

- Đặt một hình ảnh vào trong tài liệu HTML của bạn.
- Chọn lựa cách xắp sếp hình ảnh với văn bản bao quanh.
- Sửa đổi tag chèn hình ảnh để phục vụ cho người sử dụng dùng browser không có hình ảnh.
- Chỉ định chiều của hình ảnh.

Bài học

Chúng ta hãy nhìn xem làm thế nào để thêm các hình ảnh giống như "chữ M lớn" vào trong một trang web ...

Tag HTML cho Inline Graphics

Một hình ảnh "inline" được xuất hiện bên trong văn bản của trang web, giống như

hình "chữ M lớn" này.

Dạng HTML cho tag hình ảnh inline là:

```
<img src="filename.gif">
```

trong đó `filename.gif` là tên của một tập tin GIF *để ở cùng directory/folder* với tài liệu HTML. Từ "inline" có nghĩa là web browser sẽ hiển thị hình ảnh ngay trong văn bản.

Hãy chú ý các văn bản theo ngay sau "chữ M lớn" ở trên. Nếu chúng ta muốn "chữ M lớn" ở riêng trên một dòng thì sao? Để một hình ảnh xuất hiện trên một dòng riêng biệt,

chỉ cần thêm một tag `paragraph` trước và sau tag `image`:

```
<p> <img src="filename.gif"> <p>
```

Sự sắp xếp trong hàng của văn bản và Inline Graphics

Tag <img...> có thể thêm thuộc tính để điều khiển vị trí của văn bản so với hình ảnh trong hàng. Thuộc tính `align` được thêm vào trong tag <img....> có thể đem lại những hiệu quả sau:

1. `align=top`

```
<img align=top src="filename.gif">
```

is for Maricopa Community Colleges located in the Valley of the Sun, metropolitan Phoenix, Arizona. Our license plates say that we are the Grand Canyon State...

---

2. `align=middle`

```
<img align=middle src="filename.gif">
```

is for Maricopa Community Colleges located in the Valley of the Sun, metropolitan Phoenix, Arizona. Our license plates say that we are the Grand Canyon State...

---

3. `align=bottom` (mặc định)

<img align=bottom src="filename.gif">

is for Maricopa Community Colleges located in the Valley of the Sun, metropolitan Phoenix, Arizona. Our license plates say that we are the Grand Canyon State...

Chú ý cách sắp xếp văn bản trên mỗi dòng...(có thể thu hẹp hay mở rộng bề rộng của cửa sổ browser WWW để xem rõ sự thay đổi). Với HTML 2.0, bạn không thể có những khối văn bản ngay bên cạnh hình ảnh. Trong những bài học sau, chúng ta sẽ xem cách tạo ra những hiệu quả đó.

Đặt một hình ảnh vào trong tài liệu HTML của bạn

Lưu ý: Nếu bạn chưa có tài liệu từ bài học trước, bạn có thể download ngay bây giờ. Bạn cũng cần có hình GIF từ Trung tâm Download Hình ảnh Bài 7.

Trong bài tập này, bạn sẽ thêm hình ảnh giới thiệu núi lửa trong trang web của bạn.

1. Mở lại workspace của bạn (nếu nó chưa được mở).
2. Mở lại tài liệu `volc.htm` trong trình soạn thảo văn bản.
3. Bên trên mục `<h1>Volcano Web</h1>` đánh vào:
4. `<img src="lava.gif">`

   Dạng HTML này sẽ thêm trên đỉnh trang web của bạn hình ảnh dung nham mà bạn đã download trong bài trước.

5. Lưu trữ và reload trong web browser của bạn.

Trong việc đặt hình ảnh, bạn có thể tự hỏi tại sao chúng ta không cần đặt một tag <p> sau hình ảnh. Điều này không cần thiết bởi vì văn bản theo sau nó là một tiêu đề. Một web browser **luôn luôn** chèn một dấu ngắt đoạn trước và sau một tag <h1,h2,h3...>.

Thuộc tính `alt="..."`

Nếu trang web của bạn sẽ được xem bởi một người sử dụng browser dạng văn bản (text-only browser, ví dụ như lynx), họ sẽ không có khả năng để xem bất kỳ những hình ảnh inline nào. Hoặc đôi khi người sử dụng tắt đi việc trình bày hình ảnh inline để tiết kiệm thời gian download qua các kết nối mạng chậm. Một thuộc tính của tag <img...> cho phép thay vào vị trí của hình ảnh bằng một chuỗi văn bản mô tả nó.

Bình thường khi gặp trường hợp này, một người sử dụng text-browser sẽ thấy một vị trí giữ chỗ (place holder) để phần đầu trang web của chúng ta trông giống như sau:
[IMAGE]
Volcano Web
In this lesson you will use the Internet to research information
on volcanoes and then write a report on your results.

In this Lesson...

Cách trình bày này giúp người sử dụng biết rằng có một hình ảnh được chen vào đầu của trang này. Bạn có thể sửa đổi tag <img> để thay vì trình bày một vị trí giữ chỗ như trên, browser sẽ hiển thị một dòng văn bản. Lấy ví dụ trong bài của chúng ta, chúng ta có thể thêm "A lesson on:" bằng cách điều chỉnh lại tag <img...> như sau:

```
<img alt="A Lesson on:" src="lava.gif">
```

Thuộc tính `alt="..."` thay vị trí giữ chỗ bằng một dòng văn bản để cho một text-browser (hay khi tắt việc hiển thị hình ảnh), bài của chúng ta sẽ xuất hiện như sau:

A Lesson On
Volcano Web
In this lesson you will use the Internet to research information
on volcanoes and then write a report on your results.

In this Lesson...
Bây giờ hãy điều chỉnh tương tự như trên vào trang HTML của bạn cho tag `<img>` có hình núi lửa.

Thuộc tính chiều cao và chiều rộng

Một khả năng khác mà bạn có thể muốn thêm vào tag <img...> là hai thuộc tính để xác định kích cở hình ảnh tính bằng số điểm (pixel). Tại sao vậy? Thông thường web browser của bạn phải tự xác định những kích cở này khi nó đọc hình ảnh; việc nạp trang của bạn có thể nhanh hơn nếu bạn chỉ định những thông số này trong HTML.

Dạng để thêm những thông số này là:

```
<img src="filename.gif" width=X height=Y >
```

trong đó X là chiều rộng và Y là chiều cao của hình ảnh tính theo điểm.

Bạn có thể sử dụng nhiều chương trình tiện ích để xác định những giá trị số này. Một cách khác để tìm những kích cở của hình ảnh là nạp nó vào trong browser của bạn (bạn có thể sử dụng chức năng kéo và thả - drag and drop - biểu tượng của hình ảnh vào trong browser của bạn), khi đó chiều cao và chiều rộng sẽ được hiển thị trên thanh tiêu đề của browser.

Với ví dụ của chúng ta trong bài học này, hình `lava.gif` có 300 điểm chiều rộng và 259 điểm chiều cao. Vì vậy bạn nên điều chỉnh trong tập tin `volc.htm` thành:

```
<img alt="A Lesson on:" src="lava.gif" width=300 height=259>
```

Chú ý: Thứ tự các thuộc tính trong tag `<img>` không quan trọng.

Thông thường chúng ta sẽ tự hỏi có thể thay đổi những kích cở của hình ảnh bằng cách đưa vào những số khác với kích cở thật của hình ảnh không. Câu trả lời là **được** nhưng kết quả nhận được thì không thể biết trước. Nếu bạn cho những số lớn hơn (để làm cho hình lớn hơn) kết quả là một hình có các "khối màu" ("blocky" picture). Nếu bạn cho những số bé hơn (để làm cho hình nhỏ hơn) kết quả có thể làm biến dạng hình ảnh. Bạn có thể tự làm thử để có kinh nghiệm. Với chúng tôi, đó là một sự sai lầm!

Kiểm tra lại công việc của bạn

Bạn có thể so sánh công việc của bạn với ví dụ mẫu để biết tài liệu này nên xuất hiện như thế nào. Nếu tài liệu của bạn không giống như ví dụ mẫu, kiểm tra lại xem bạn đã đánh vào dạng hình ảnh trong trình soạn thảo như thế nào. Hãy chắc chắn rằng bạn đã đánh vào theo sự hướng dẫn trong phần **Đặt một hình ảnh vào trong tài liệu HTML của bạn** của bài này.

Nếu bạn thấy một biểu tượng hình với một dấu chấm hỏi như sau:

trước hết hãy kiểm tra lại xem tập tin HTML và hình ảnh GIF có cùng ở trên một folder/directory không. Sau đó, bạn cũng nên chắc chắn rằng tên tập tin bạn đánh trong tag `<img... >` là tương ứng với tên của tập tin GIF.

Chú ý: Một số hệ thống máy tính (UNIX) có phân biệt dạng chữ hoa và chữ thường (case sensitive) trong tên tập tin, điều này có nghĩa rằng tập tin `"lava.GIF"` KHÔNG phải là `"lava.gif"`. Một số máy tính khác (Macintosh) lại xem chúng là cùng một tập tin. Ngay cả khi máy tính của bạn không có sự khác biệt như vậy, chúng tôi vẫn đề nghị bạn nên xét lại dạng chữ trong cách đặt tên tập tin và các tham khảo chúng trong HTML của bạn. (Nhiều WWW servers ở trên UNIX).

Xem lại

1. Dạng HTML cho một hình ảnh inline là gì?
2. Tag nào bạn phải thêm vào trước một tag hình ảnh để nó xuất hiện trên một dòng riêng?
3. Làm thế nào để thêm hình ảnh núi lửa vào tài liệu của bạn?
4. Thuộc tính `alt="...."` làm gì? Thuộc tính `heighth="...."` là gì?

Thực tập tự do

Thêm một hình ảnh inline vào trang web của bạn bằng cách sử dụng một hình GIF đang có trong máy của bạn hoặc một hình được download từ Internet.

8. Tạo Liên kết bằng Anchors

Mục đích của bài này chỉ cung cấp cho ta những thông tin bổ sung... Vì thế bài này nhanh và khá dễ hiểu!

Thế nào làURL?

Sức mạnh thật sự của web là khả năng tạo ra những liên kết siêu văn bản (hypertext link) đến các thông tin liên quan. Những thông tin này có thể là những trang web khác, những hình ảnh, âm thanh, đoạn phim, animation, các chương trình, nội dung của Gopher server, một trình log-in vào máy tính ở xa (remote computer), hay một tập hợp các phần mềm, hay một "ftp" site.

WWW sử dụng một cách định địa chỉ gọi là **URL**, hay **U**niform **R**esource **L**ocator (đôi khi còn gọi là "Universal Resource Locator"), để chỉ ra vị trí của những mục như vậy. Những liên kết này thường được biểu diễn bằng những chữ màu xanh có gạch dưới và được gọi là **anchor**.

Trong những bài học tiếp theo chúng ta sẽ:

- Xem lại khái niệm của URL
- Tìm và chép URL từ web browser vào tài liệu văn bản HTML của bạn.
- Viết một anchor HTML để liên kết đến tài liệu trong cùng thư mục với tài liệu đầu tiên.
- Viết một anchor HTML để liên kết đến tài liệu khác thư mục với tài liệu đầu tiên.
- Viết một anchor HTML để liên kết đến một trang web khác trên Internet.
- Viết một anchor HTML để liên kết đến Gopher Server.
- Viết một anchor HTML để liên kết đến một phần khác của cùng một tài liệu.
- Liên kết với một hình ảnh như một "hyperlink" đến tài liệu khác.

Ồ! Nghe như có nhiều việc phải làm! Đừng lo lắng, mọi thứ điều không phức tạp như bạn nghĩ. Sau cùng, nếu không có hypertext thì chúng ta không thể gọi tài liệu này là **Soạn thảo HTML** mà phải gọi là **Soạn thảo TML**

8a. Liên kết đến tập tin cục bộ

Tài liệu của tôi có thể nói chuyện với tài liệu của tôi được không? Được chứ, ít nhất là chúng có thể liên kết với nhau được!

Mục đích

Sau bài học này, bạn có thể:

- Tạo liên kết đến tài liệu HTML trong cùng directory/folder với tài lệu chính của bạn.
- Tạo liên kết để hiển thị một hình ảnh.
- Tạo liên kết đến một tập tin trong một directory/folder khác với tài liệu chính của bạn.
- Tổ chức lại cấu trúc web của bạn.

Bài học

Bây giờ, bạn sẽ đi bước đầu tiên của "anchoring" bằng cách tạo ra một liên kết siêu văn bản đến một trang web thứ hai. Những liên kết này được gọi là "cục bộ" (local) bởi vì chúng cùng ở trên một máy tính và là tài liệu đang làm việc của bạn (chúng không cần phải thám hiểm trên Internet). Bạn sẽ thay đổi một chút về sự phát triển web site của bạn (bạn có thấy rằng điều này tạo ra một cái gì đó hơi khác một chút với từ "home gape" không?).

Lưu ý : Nếu bạn chưa có tài liệu từ bài trước, hãy download một bản sao.

Liên kết đến Tập tin cục bộ

Liên kết `anchor` đơn giản nhất là liên kết để mở tập tin HTML khác trong cùng một thư mục với trang web đang được thể hiện. Dạng HTML để làm việc này là:

```
<a href="filename.htm">text that responds to link</a>
```

Hãy xem chữ "a" là **a**nchor còn "href" là "**h**ypertext **ref**erence.

Tên tập tin phải là một tập tin HTML khác. Bất cứ văn bản nào xuất hiện sau ký hiệu > và trước ký hiệu sẽ là "siêu văn bản", nó được hiển thị bằng gạch dưới và "hyper".

Hãy theo những bước sau để xây dựng một Anchor trong tài liệu HTML của bạn để liên kết đến một tập tin cục bộ:

1. Mở văn bản HTML của bạn , `volc.htm`, trong trình soạn thảo.
2. Đầu tiên, dưới tiêu đề `Volcanic Places in the` USA đánh vào đoạn văn để giới thiệu về hai núi lửa để sử dụng cho các phần kế tiếp.

3.
4. Listed below are two places in the United States that
5. are considered "active" volcanic areas.
6. Bên dưới tiêu đề "Mount St. Helens", đánh:
7.
8. On May 18, 1980, after a long period of rest, this quiet
9. mountain in Washington provided
10. detailed observations on the mechanics
of highly explosive eruptions.

Đoạn văn "detailed observations" sẽ tạo liên kết cho người xem đến một tài liệu HTML thứ hai gọi là `msh.htm`. Tập tin HTML chưa có; chúng ta sẽ tạo ra nó trong các bước (5) và (6).

11. Lưu và đóng tài liệu HTML của bạn
12. Bây giờ mở một tài liệu **Mới** trong trình soạn thảo của bạn.
13. Nhập các dòng sau trong màn hình mới:
14. <html>
15. <head>
16. <title>Mount St Helens</title>
17. </head>
18. <body>
19. <h1>Mount St Helens</h1>
20. The towering pine trees of this once-quiet mountain
21. were toppled over like toys.
22. </body>
23. </html>
24. Lưu tập tin này với tên `msh.htm` trong cùng một directory/folder với tài liệu HTML của bạn (`Volc.htm`).
25. **Reload** `Volc.htm` trong web browser.
26. Kiểm tra liên kết siêu văn bản tại các từ "detailed observations". Khi được chọn, nó phải kết nối đến trang mới về Mount St. Helens.

Liên kết Anchor đến một hình ảnh

Trong bài 7a, chúng ta đã xét cách trình bày một hình ảnh "inline" trong trang Web. Với tag `anchor`, bạn cũng có thể tạo ra một liên kết để hiển thị một tập tin hình ảnh. Khi chọn một liên kết `anchor` , hình ảnh sẽ được download và trình bày riêng trong một trang trống hoặc cũng có thể web browser của bạn tự động gọi một trình ứng dụng để hiển thị hình ảnh.

Liên kết `anchor` đơn giản nhất là liên kết đến một tập tin trong cùng một directory/folder với tài liệu gọi nó. Dạng để tạo một liên kết siêu văn bản đến một hình ảnh giống như liên kết đến một tài liệu HTML khác ở trên

text that responds to link
trong đó `filename.gif` là tên của tập tin hình ảnh GIF.

Bây giờ theo những bước sau để thêm một liên kết với một tập tin hình ảnh trong tài liệu của bạn:

1. Trước hết, bạn cần chép một bản sao tập tin hình ảnh GIF từ Trung tâm Download Hình ảnh Bài 8a.
2. Mở tập tin `msh.htm` trong trình soạn thảo.
3. Điều chỉnh đoạn văn bản để tạo liên kết tới hình ảnh Mount St. Helens.
4. The towering pine trees of this once-quiet mountain
5. were toppled over like toys.
6. **Lưu lại** tập tin `msh.htm` và **Reload** trong web browser của bạn.
7. Bây giờ click vào liên kết mà bạn vừa tạo ra trong bước (3)

8. Một bức ảnh cây cối bị thổi ngã sẽ được thể hiện

Những liên kết đến các thư mục khác

Tag `Anchor` cũng có thể tạo liên kết đến một tài liệu HTML hoặc tập tin hình ảnh trong một directory/folder khác có liên quan tới tài liệu mà có chứa anchor. Ví dụ như, trong bài của chúng ta , chúng ta có thể lưu trữ tất cả các tập tin hình ảnh trong một directory/folder riêng gọi là `pictures`. Khi bạn tạo ngày càng nhiều tập tin HTML, sẽ tốt hơn khi giữ các tập tin hình ảnh trong một vùng riêng biệt. Hãy làm điều sau:

1. Trong hệ thống của bạn tạo một sub-directory/folder `pictures` trong cùng vị trí lưu trữ tập tin `Volc.htm`.
2. Chuyển tập tin `msh.gif` đến sub-directory/folder mới này.
3. Mở tập tin `msh.htm` trong trình soạn thảo.
4. Điều chỉnh tag `anchor` cho hình ảnh cần đọc:
5. The towering pine trees of this once-quiet mountain
6. were toppled over
   like toys.

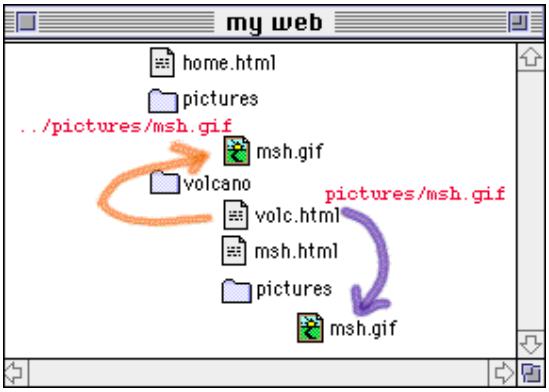
   Chú ý: Với HTML bạn có thể hướng dẫn web browser của bạn mở mọi tài liệu/hình ảnh tại thư mục cấp *thấp hơn* tài liệu hiện tại bằng cách sử dụng ký tự "/" để chỉ ra thư mục con gọi là "pictures."

7. Lưu trữ tài liệu HTML và reload lại trong browser của bạn

Nếu tất cả đều tốt, việc liên kết trong câu mô tả các cây bị thổi ngã sẽ gọi đến tập tin hình ảnh đang có trong directory/folder `pictures`.

Liên kết Anchor đến thư mục cấp cao hơn

Loại liên kết chúng ta vừa thực hiện được gọi là các liên kết "tương đối" (relative link), nghĩa là một Web browser có thể xây dựng lại đầy đủ URL dựa trên vị trí hiện hành của trang HTML và những thông tin liên kết trong các tag `<a href=...>`. Điều này rất tốt bởi vì bạn có thể xây dựng tất cả các trang web của bạn trên cùng một máy tính, kiểm tra chúng, và di chuyển chúng vào một máy khác, tất cả các liên kết tương đối vẫn giữ nguyên hiệu ứng.



Trong bài học chúng ta đã biết làm thế nào để xây dựng một siêu liên kết (hyperlink) đến một tài liệu lưu trữ trong một thư mục **thấp hơn** trang HTML đang làm việc. Lưu ý rằng bạn cũng có thể xây dựng một liên kết đến một thư mục **cao hơn** bằng cách sử dụng HTML sau:

return to home

Mỗi một lần xuất hiện của `"../"`, URL của liên kết anchor báo cho web browser đi đến một directory/folder cao hơn một cấp tương ứng với trang hiện tại; trong trường hợp cụ thể này có nghĩa là lên hai cấp thư mục và tìm tập tin `home.htm`.

Trong ví dụ của chúng ta, hãy xét trường hợp khi thư mục con `pictures` không cùng ở trong directory/folder với tập tin `volc.htm` mà ở cao hơn một cấp

Trong phần trước chúng ta đã tạo liên kết từ `volc.htm` tới tập tin `msh.gif` trong thư mục con:

```
<img src=
  "pictures/msh.gif">
```

Bây giờ, chúng ta muốn tổ chức lại cấu trúc web để thư mục `pictures` ở một mức cao hơn. Liên kết sẽ được viết như sau:

```
<img src=
  "../pictures/msh.gif">
```

Vì vậy web browser tìm kiếm thư mục `pictures` được lưu trữ cao hơn một mức so với tập tin `volc.htm`.

Sự tiện lợi của cấu trúc này là dễ dàng lưu lại một số lượng lớn hình ảnh trong directory/folder cao hơn để có thể dùng chung trong những trang web khác. Chúng ta có thể thực hiện một bài học khác về địa mạo (điểm đặc trưng tự nhiên của bề mặt quả đất) và sử dụng những hình ảnh được lưu trữ trong directory/folder này.

Bây giờ đã đến lúc tổ chức lại các tập tin HTML của chúng ta. Để thực hiện được điều này bạn cần làm quen với việc di chuyển tập tin và thư mục trên máy tính của bạn. **Hãy đọc một cách cẩn thận! Bạn có thể cảm thấy khá phức tạp, nhưng chẳng bao lâu nữa tất cả đều rõ ràng thôi!**

1. Đầu tiên, tạo directory/folder mới có tên là `volcano` (một lời khuyên là nên giữ các tên tập tin ở dạng chữ nhỏ).
2. Di chuyển hai tập tin HTML `volc.htm` và `msh.htm` vào trong directory/folder mới này.
3. Di chuyển directory/folder `pictures` (cùng với tập tin `msh.gif` bên trong) đến **cùng cấp** với directory/folder mới `volcano`. Tương tự, di chuyển tập tin `lava.gif` mà chúng ta tạo ra trong bài 7a vào thư mục các hình này.
4. Như vậy toàn bộ thư mục (`workarea`) bây giờ gồmhai thư mục con - một để giữ các tập tin HTML (`volcano`) và một giữ các hình ảnh (`pictures`):

5. + workarea **(directory)**
6.
7. + pictures **(directory)**
8. msh.gif
9. volc.gif
10.
11. + volcano **(directory)**
12. volc.htm
13. msh.htm

14. Chúng ta đã di chuyển một số thứ do vậy bây giờ chúng ta phải cập nhật lại các liên kết anchor trong các tập tin HTML. Trước hết, hãy xem lại liên kết cục bộ trong tập tin `volc.htm`:

```
15.    <h3>Mount St Helens</h3>
16.    On May 18, 1980, after a long period of rest, this quiet
17.    mountain in Washington provided <a href="msh.htm">detailed
18.    observations</a> on the mechanics of highly explosive
19.    eruptions.
```

Chú ý: Vì tập tin `msh.htm` vẫn ở cùng thư mục với volc.htm, chúng ta không cần thay đổi gì cả trong tập tin HTML này! Bạn có thấy sự liên kết tương đối là một tính chất rất hay của HTML không?

20. Nhưng bây giờ hãy xem lại liên kết đến hình Mt. St Helens đã tạo ra trong tập tin `msh.htm`:

```
21.    The towering pine trees of this once-quiet mountain
22.    were <a href="pictures/msh.gif">toppled over
       like toys</a>.
```

Mở tập tin này trong trình soạn thảo văn bản và điều chỉnh liên kết như sau:

```
The towering pine trees of this once-quiet mountain
were <a href="../pictures/msh.gif">toppled over
like toys</a>.
```

Liên kết tương đối này báo cho web browser xét lên một cấp so với directory/folder hiện hành (`volcano`) rồi tìm directory/folder có tên (`pictures`) mà ở đó có chứa ảnh `msh.gif`

23. Cuối cùng, bạn cần điều chỉnh lại tag `<img...>` tag để hiển thị hình ảnh tiêu đề. Mở tập tin `volc.htm` trong trình soạn thảo và sửa đổi dòng ngay sau tag `<body>`:

```
24.     <img alt="A Lesson on:" src="../pictures/lava.gif"
                        width=300 height=259>
```

25. Lưu lại tập tin của bạn. Bạn nên **Mở lại** tập tin `volc.htm` trong web browser của bạn và thử lại liên kết đến `msh.htm` và liên kết đến hình Mount St Helens.

Thêm một sự thay đổi nhỏ nữa

Bước thay đổi nhỏ cuối cùng này có thể không rõ ràng, ngưng chúng tôi sẽ cố gắng giải thích ngắn gọn. Điều cuối cùng bạn nên làm trong bài này là đổi tên tập tin từ `volc.htm` thành `index.htm`

Tại sao vậy? Hãy xem như bạn đã kết thúc bài này và sẵn sàng đưa web site của bạn vào World Wide Web server để mọi người cùng xem. Và chúng ta hãy giả thiết rằng địa chỉ Internet của server này tại Big University là:

```
   http://www.bigu.edu/
```

Và tập tin của bạn được cất vào thư mục sau:

```
--= top level of server: www.bigu.edu
  /courses
      /science
         /geology
```

như vậy URL cho Volcano Web có thể là:

```
   http://www.bigu.edu/courses/science/geology/volcano/volc.htm
```

Ồ, khá dài nhỉ? Bây giờ là những lời giải thích như đã hứa -- trên nhiều WWW server bạn có thể chỉ định một tên chuẩn gọi là trang web "mặc nhiên" (default) cho một thư mục và trên nhiều hệ thống tên đó là .... `index.htm`. Điều này có nghĩa rằng địa chỉ Internet:

```
   http://www.bigu.edu/courses/science/geology/volcano/
```

tương đương với

```
   http://www.bigu.edu/courses/science/geology/volcano/index.htm
```

Điều này có thể làm cho bạn nghĩ là đã tiết kiệm được nhiều năng lượng trong việc xóa bớt 20 ký tự từ URL! Thực tế, nó làm cho URL của bạn có vẻ chuyên nghiệp hơn. Nếu bạn nói có một Home page là Longhorn Cheese,

```
   http://www.cheese.com/longhorn/
```

có vẻ ít rườm ra hơn là

```
   http://www.cheese.com/longhorn/longhorn.htm
```

khi để cho mọi người đọc URL của bạn và thử đánh vào browser của họ.

Kiểm tra lại công việc của bạn

So sánh trang web với ví dụ mẫu để biết tài liệu nên xuất hiện như thế nào. Trước hết bạn sẽ thấy trang *Volcano Web*. Khi bạn click vào siêu văn bản tại `detailed observations`, web browser của bạn sẽ hiển thị một trang mới. Cuối cùng, khi bạn click vào **`toppled`**

`over like toys`, web browser của bạn sẽ hiển thị trong một cửa sổ một tập tin hình ảnh được cất trong một sub-directory/folder.

Sử dụng button **back** của browser hai lần để quay về trang này. Nếu trang web của bạn khác với mẫu, xem lại văn bản bạn đánh vào trong trình soạn thảo văn bản.

Xem lại

1. Các bước nào được sử dụng để tạo một liên kết trong tài liệu của bạn đến một tập tin cục bộ?
2. Những bước nào được sử dụng để tạo một liên kết khi muốn hiển thị hình ảnh trong cửa sổ riêng biệt?
3. Làm thế nào để tạo một liên kết đến một tập tin trong một directory/folder thấp hơn tài liệu chính của bạn? cao hơn?
4. Ý nghĩa của tập tin có tên `index.htm` trên một WWW server là gì?

Thực tập tự do

Tạo một tài liệu HTML sử dụng các định dạng HTML mà bạn đã quen biết đến giai đoạn này. Trở về trang đầu tiên và tạo một anchor để liên kết đến trang mới này.

8b. URLs - Con trỏ đến Internet

`URL? Earl? Yurl? hurl? gyrl?`
Đến lúc để liên kết đến Big Wide Web sử dụng sơ đồ định địa chỉ của web.

Mục đích

Sau khi hoàn tất bài này bạn có thể:

- Nhận dạng chức năng của Uniform Resource Locators (URLs).
- Nhân biết cấu trúc của một URL.
- Kiểm tra URL trong liên kết siêu văn bản của bất kỳ trang web nào.

Bài học

Chú ý: Trong bài này, bạn sẽ không cần tập tin văn bản HTML của bạn. Một bài học không cần gắng sức!

URL là gì?

WWW sử dụng Uniform Resource Locator (URL) để tìm vị trí của những tập tin và tài liệu của các máy tính trên Internet. Trên màn hình browser của bạn, URL cho tài liệu này được thể hiện ở phần phần trên của cửa sổ browser. URL bao gồm:

- Một sự định dạng loại Internet server;
- Một địa chỉ Internet; và
- Một đường dẫn tập tin (file path) chỉ tới mục đang xem.

URL là thứ bạn cần để xây dựng một liên kết từ trang web bạn tạo ra đến những phần thông tin khác trên Internet. Để biết chi tiết hơn, hãy xem Curling Up To URLs (v0.2)

Cấu trúc URL như thế nào?

Cấu trúc của một URL là:

type://in.ter.net.address/directory/sub-directory/.../filename

"type" chỉ ra loại của Internet server được truy xuất:

http
Một World Wide Web server. "HTTP" tượng trưng cho HyperText Transfer Protocol.

gopher
Một Internet Gopher site, thư mục các tập tin và thông tin theo dạng thực đơn.

ftp
Một anonymous File Transfer Protocol (FTP) site, truy tìm các tập tin.

telnet
Khởi tạo một trình Telnet để log on từ xa tới một máy tính khác. Khi chọn, web browser sẽ khởi động chương trình Telnet và kết nối tới site được chỉ định .

WAIS
Wide Area Indexed Server- một site dò tìm các tài liệu theo chủ đề được xác định bằng các từ khóa.

file
Một tập tin trên máy cục bộ (đĩa cứng, đĩa mềm, file server nội bộ).

Sau Type luôn luôn là "://" và địa chỉ Internet của máy từ xa. Sau đây là cấu trúc của địa chỉ này:

host.domain.domain.domain

Ví dụ như:

machine.department.college.edu
123.45.6.78
office.company.com
agency.branch.gov
machine.organization.country

Nếu URL chỉ đến mức chính của host này ("home page" của nó), thì URL được kết thúc bằng một dấu "/". Nếu bạn liên kết tới một thư mục con hoặc một tập tin, bạn phải thêm chính xác đường dẫn tới nó bằng cách sử dụng dấu "/".

Chú ý: Hầu hết các server đều kiểm tra ngữ pháp! Và cũng kiểm tra dạng chữ hoa hay chữ thường! Các tên tập tin trên các máy UNIX có phân biệt chữ hoa và chữ thường, điều này có nghĩa là tập tin có tên

SpiffyText.htm

là khác với tập tin

spiffytext.htm

Tạo kinh nghiệm với URLs

Chú ý rằng URL có thể tạo liên kết đến mọi site, thư mục, thư mục con, tập tin văn bản, hình ảnh, digital movie, hay tập tin âm thanh trên mọi Internet site đã cài đặt cho phép truy xuất tự do. Cách tốt nhất để xem các URL khác nhau là di chuyển chuột của bạn qua các liên kết siêu văn bản trong trang web nào đó- nếu bạn nhìn vào bên dưới web browser của bạn, nó sẽ trình bày URL mà bạn muốn nối đến khi bạn click vào liên kết đó. Bạn có thể vào một site lớn như Yahoo và "nhìm trộm" các URL (bạn có thấy URL của Yahoo khi bạn di chuyển mouse qua liên kết trong câu vừa rồi không?)

Còn bây giờ là một cách rất dễ để chép lại một URL của một liên kết trong một trang bất kỳ. Trước hết bạn mở thực đơn "bí mật" từ một liên kết siêu văn bản nào đó trong một trang web (Ấn và giữ chuột trong Macintosh; ấn và giữ button phải của chuột trong Windows và Unix). Từ thực đơn này, chọn **Copy This Link Location** (hay chức năng tương tự). Sau khi thả button chuột ra, chuyển đến một trình soạn thảo văn bản nào đó và chọn **Paste** từ thực đơn **Edit**. Hay quá! Bạn đã lấy được một URL từ một liên kết trong trang web (với cách này bạn có thể chép một URL ngay cả khi bạn chưa tham quan trang mà nó liên kết đến!)

Xem lại

1. Mục đích của URL cho World Wide Web?
2. Nơi nào thấy được URL trên một màn hình WWW?
3. Cấu trúc cơ bản của một URL?
4. Làm thế nào để thấy được URL của một liên kết siêu văn bản sẽ nhảy đến?

Thực tập tự do

Tìm vài site trên Internet mà bạn cảm thấy thích. Với mỗi site, ghi lại URL của nó đã được trình bày gần trên đỉnh của cửa sổ browser. Thử xem bạn có thực hiện được việc chép và dán (copy and paste) các URL vào một tài liệu văn bản hay không. Bạn sẽ sử dụng danh sách này để thêm các liên kết từ các trang web của riêng bạn đến các site mà bạn đã tìm thấy.

8c. Liên kết đến các site Internet bên ngoài

Bạn nên làm tươi lại cỗ máy của bạn bằng cách chạm vào Information SuperHighway, được không? Ở đây chúng ta sẽ mở rộng cách sử dụng của các tag `anchor` để tạo ra liên kết đến các tài nguyên có sẵn trên Internet.

Mục đích

Sau bài học này bạn có thể:

- Tạo một anchor để liên kết đến một Internet site.
- Chép nhanh URL của một site và sử dụng nó trong tài liệu HTML của bạn.

Bài học

Lưu ý: Nếu bạn chưa có tài liệu bài học trước, bạn có thể download ngay bây giờ.

HTML cho các Anchor đến Internet

Tạo liên kết đến một site trên Internet bao gồm những điều chúng ta đã xét trong bài **Liên kết với tập tin cục bộ** (Bài 8a) cùng với bài về **URLs** (Bài 8b). Dạng HTML đầy đủ cho một liên kết `anchor` đến một mục trên Internet là:

```
<a href="URL">Text to Activate Link</a>
```

Trong đó **URL** giữa hai dấu nháy là dạng đầy đủ của Uniform Resource Locator, cho biết địa chỉ của Internet site mà bạn muốn gởi đến người đọc. Chuỗi `Text to Activate Link` là dòng chữ sẽ được hiển thị như là siêu văn bản trong web browser thường (nhưng không phải luôn luôn) có dạng gạch dưới và màu xanh. Khi người đọc click trên siêu văn bản này, web browser sẽ liên kết chúng với Internet site được chỉ định bởi URL. Nên nhớ rằng một URL có thể là một liên kết đến một World Wide Web (WWW) server khác, Gopher server, FTP site, hoặc bất kỳ một tập tin văn bản, hình ảnh, âm thanh, video trên các server này.

Bây giờ chúng ta sẽ thêm một liên kết siêu văn bản đến site của NASA là site có thông tin về các núi lửa trên Sao hỏa (planet Mars). Theo những chỉ dẫn sau để thêm liên kết anchor vào tài liệu HTML của bạn:

1. Mở tập tin `index.htm` của bạn trong trình soạn thảo văn vản.
2. Bên dưới tiêu đề **Volcanic Places on Mars**, nhập văn bản sau:
3.
4.
5. Mars has its fair share of volcanic landforms,

```
including the largest known volcano in the solar system,
<a href="http://bang.lanl.gov/solarsys/raw/mars/olympus.gif">
Olympus Mons</a>
```

Chú ý: Chúng ta đã tạo sự liên kết đến hai loại thông tin khác nhau. Siêu liên kết (hyperlink) đầu tiên nối đến một trang web mô tả thông tin về Sao hỏa. Cái thứ hai là liên kết đến một ảnh GIF nhỏ của núi lửa Martian.

9. **Lưu** và **Reload** trong web browser của bạn.

Một cách dễ dàng để nhập các URL trong tag Anchor

Như là bạn đã thấy trong các trang web, URL của trang hiện tại có thể thấy được ở phần trên của cửa sổ web browser (bạn có thể tìm kiếm thực đơn để **Trình bày URL**). Ví dụ như, trong tài liệu này, URL trông giống như sau:

http://www.mcli.dist.maricopa.edu/tut/tut8c.html

Bạn có thể sử dụng chuột để chọn và **chép lại** một URL từ phần trình bày trong web browser rồi sau đó dán nó vào trong tag `anchor` của tài liệu HTML. Điều này hiệu quả hơn là viết URL ra giấy rất nhiều (đôi lúc khá dài!).

Bây giờ chúng ta sẽ thêm một số liên kết tới các site khác mà chúng ta sẽ liệt kê ra trong phần **References** của bài Volcano. Một trong các site có thể cung cấp thông tin liên quan là US Geological Survey.

Theo những bước sau:

1. Mở tài liệu HTML `index.htm` trong trình soạn thảo
2. Dưới tiêu đề "References", nhập:
3.
```
<ul>
<li> <a href="">Educational Resources from the
USGS</a>
</ul>
```

Chú ý: Chúng ta đã tạo ra liên kết siêu văn bản và vẫn cần phải thêm vào một URL giữa hai dấu nháy.

8. Nối đến US Geological Survey Education Index trong cửa sổ browser của bạn.
9. Từ trang web, sử dụng chuột để **chọn** URL đang được hiển thị trong vùng trình bày URL.
10. Từ thực đơn **Edit**, chọn **Copy** URL.
11. Bây giờ , trở lại tài liệu HTML của bạn `index.htm`
12. Click mouse vào giữa hai dấu nháy mà bạn đã đánh vào ở bước #2 và **Dán** vào tài liệu bạn đã chép trong bước #5. Tag `anchor` cuối cùng giống như sau:
13.
```
<a href="http://info.er.usgs.gov/education/index.html">
Educational Resources from the USGS</a>
```

Chú ý: Bạn vừa thiết lập xong một cấu trúc HTML sử dụng Danh sách Không có thứ tự, với mỗi mục là một liên kết siêu văn bản đến một site có cung cấp thông tin về núi lửa. Để xem lại các danh sách, hãy chọn Bài 6.

Như là phần thực tập thêm, hãy vào các site Internet dưới đây với các tài liệu về Địa chất (Geology) và/hay núi lửa (Volcanoes). Chép các URL và xây dựng ít nhất hai site trong phần **Reference**:

Các Site có Thông tin về Núi lửa:

Alaska Volcano Observatory * Cascades Volcano Observatory * Michigan Tech Volcanoes Page * NASA Earth Observing System (EOS) IDS Volcanology Team * NASA Facts: Volcanoes and Global Climate Change * Smithsonian Global Volcanism Program * Volcano Listserv * Volcano Watch Newsletter * JASON Project VI: Island Earth

Các Site có Thông tin về các Tai họa Tự nhiên (Natural Hazard):

Disaster Information * Emergency Preparedness Information eXchnage (EPIX) * NGDC Natural Hazards Data

Các Site có Thông tin Tổng quát về Địa chất:

Earth Resources Observation Systems (EROS) Data Center * EINet Galaxy * Environmental Internet Catalog * NASA Internet Connections * NOAA Environmental Information Server * Planet Earth Home Page * Rice University * United States Geological Survey * Whole Internet Catalog * WWW Virtual Library

---

Kiểm tra lại công việc của bạn

So sánh trang web của bạn với ví dụ mẫu để biết tài liệu nên xuất hiện như thế nào. Nếu trang web của bạn không giống như ví dụ mẫu hay các liên kết không nối đến đúng site ở xa, hãy kiểm tra lại văn bản bạn đã đánh trong trình soạn thảo. Chú ý rằng danh sách các tham khảo của bạn có thể không giống như trong ví dụ mẫu.

Xem lại

1. Địa chỉ cho một mục trên Internet là gì?
2. Những bước nào cần cho việc tạo một liên kết anchor đến NASA Internet Site?
3. Cách nhanh nào được dùng để đưa một URL vào trong liên kết anchor?
4. Hãy nói cho bạn bè về các liên kết khác mà bạn đưa vào tài liệu của bạn.

Thực tập tự do

Tạo các liên kết anchor từ trang web của bạn để nối đến vài địa chỉ URL mà bạn khám phá được.

8d. Liên kết Các phần của một Trang

Bạn đã biết làm thế nào để liên kết tới những trang web khác là những trang do bạn tạo ra hay là những trang có trên Internet. Nếu bạn muốn kết nối tới một phần dược chỉ định **bên trong** một tài liệu thì sao? BẠN CÓ THỂ LÀM ĐƯỢC ĐIỀU NÀY!

Mục đích

Sau bài này bạn có thể:

- Tạo một dấu hiệu tham khảo ẩn cho một phần riêng biệt bên trong một tập tin HTML.
- Xây dựng một liên kết siêu văn bản để nối đến một phần riêng biệt bên trong tập tin HTML.
- Xây dựng một liên kết siêu văn bản để nối đến một phần riêng biệt bên trong tập tin HTML khác.
- Tạo một bảng mục lục siêu văn bản cho một trang web.

Bài học

Lưu ý: Nếu bạn chưa có tài liệu từ bài học trước, bạn có thể download lại ngay bây giờ.

Anchor được đặt tên

Một **anchor được đặt tên** (named anchor) là một dấu hiệu tham khảo ẩn cho một phần đặc biệt nào đó của tập tin HTML của bạn. Nó được sử dụng để tạo liên kết tới một phần khác của cùng trang khi trang đó dài, hoặc để đánh dấu mục của một trang khác. Lấy ví dụ, trong trang bạn đang xem, tôi có thể tạo ra một dấu ẩn gọi là "check" để đánh dấu tiêu đề "Kiểm tra lại công việc của bạn" bên dưới. Sau đó, tôi viết một liên kết anchor để kết nối tới mục này trong tài liệu này. Một khi bạn click trên một liên kết đến anchor được đặt tên này, web browser của bạn sẽ chuyển đến chỗ để dòng này nằm trên đỉnh của màn hình.

Sau đây là một ví dụ để bạn có thể thử ngay bây giờ. Đi đến Kiểm tra lại công việc của bạn. Khi đến đó, bạn sẽ thấy một liên kết để trở lại mục này.

Dạng HTML cho việc tạo anchor được đặt tên là:

```
<a name="NAME">Text to Link To</a>
```

hay với liên kết bạn vừa thử ở trên:

```
<a name="check">Kiem tra lai cong viec cua ban</a>
```

Chú ý sự khác biệt với liên kết anchor `<a href=...` chúng ta đã xét trong bài 8a.

Viết một liên kết tới một Anchor được đặt tên

Khi bạn muốn tạo một liên kết siêu văn bản (hay một "liên kết anchor", xem bài 8a) tới một anchor được đặt tên, sử dụng dạng HTML sau:

```
<a href="#xxxxx">Text to act as hypertext</a>
```

hay liên kết bạn vừa thử để đến phần dưới của tài liệu này:

```
Di den  <a href="#check">Kiem tra lai cong viec cua ban</a>
```

Kí hiệu "#" chỉ cho web browser của bạn tìm trong tài liệu HTML một anchor được đặt tên là `"xxxxxx"` hay trong ví dụ của chúng ta là `"check"`. Khi bạn chọn hay click trên siêu văn bản, nó mang phần của tài liệu có chứa anchor được đặt tên lên đỉnh của màn hình.

Thêm Anchor được đặt tên vào bài *Volcano Web*

Bây giờ chúng ta sẽ xây dựng một bảng mục lục cho bài của chúng ta để nó sẽ xuất hiện ở đầu trang *Volcano web*. Các phần tử trong bảng mục lục sẽ là các tiêu đề mà chúng ta đã tạo ra và sẽ hoạt động như một liên kết siêu văn bản đến một phần riêng biệt trong bài chúng ta.

Bước đầu tiên là tạo một anchor được đặt tên cho mỗi mục tiêu đề trong bài *Volcano Web*:

1. Mở tập tin `index.htm` trong trình soạn thảo văn bản
2. Tìm tiêu đề **Introduction**. Chuyển đổi nó từ:

   ```
   <h2>Introduction</h2>
   ```

   thành:

   ```
   <h2><a name="intro">Introduction</a></h2>
   ```

Chú ý: Bạn vừa đánh dấu dòng chứa mục tiêu đề "Introduction" với một dấu tham khảo ẩn có anchor được đặt tên là "intro"

3. Theo cách tương tự, thay đổi tất cả các tags <h2> cho các phần khác:
4.
5. <h2><A NAME="term">Volcano Terminology</A></h2>
6.
7. <h2><A NAME="usa">Volcanic Places in the USA</A></h2>
8.
9. <h2><A NAME="mars">Volcanic Places on Mars</A></h2>
10.
11. <h2><A NAME="project">Research Project</A></h2>
12. Nếu bạn **Reload** liền trong web browser của bạn, bạn sẽ không nhìn thấy bất kỳ sự thay đổi nào; các tag anchor được đặt tên thêm vào đều được dấu khi người sử dụng xem trang.

Thêm liên kết tới một Anchor Được đặt tên trong Cùng một Tài liệu

Bây giờ chúng ta sẽ xây dựng một bảng mục lục sẽ xuất hiện trên đầu màn hình. Chúng ta sử dụng một danh sách không có thứ tự (xem lại bài 6 về danh sách) để tạo nó:

1. Nếu chưa mở, hãy mở tập tin `index.htm` trong trình soạn thảo.
2. Bên dưới câu đầu tiên trong tiêu đề **Volcano Web** thêm những dòng sau:
3.
4. <hr>
5. <B>In this Lesson...</B>
6.
7. Introduction
8. Volcano Terminology
9. Volcanic Places in the USA
10. Volcanic Places on Mars
11. Research Project
12.

Chú ý: Bảng mục lục này được đánh dấu trên và dưới bằng tag <hr>. Kiểu chữ *Nghiêng* được dùng cho các mục. Ở đây chúng ta chỉ mới đánh vào tên các mục. Sau này chúng ta sẽ thêm vào để các liên kết hoạt động được.

13. **Lưu** và **Reload** trong web browser của bạn.

Cuối cùng , chúng ta muốn làm cho mỗi mục của danh sách hoạt động như một liên kết siêu văn bản tới một phần khác của tài liệu. Để làm điều này, chúng ta sẽ sử dụng một biến thể của liên kết anchor cơ bản trong bài 8a. Thay vì liên kết tới một tập tin khác, chúng ta liên kết tới một trong những anchor được đặt tên (các dấu ẩn bạn đã tạo ra ở trên) bên trong cùng một tập tin HTML. Chúng ta chỉ định một anchor được đặt tên bằng cách đặt kí hiệu "#" trước tên của dấu tham khảo:

1. Mở tập tin `index.htm` trong trình soạn thảo
2. Đến mục đầu tiên trong phần mục lục. Thay đổi nó từ:
3.
4. Introduction

thành:

<A HREF="#intro">Introduction</A>

5. Bây giờ bạn nên điền các liên kết khác để đoạn văn này trông giống như sau:

6.
7. <hr>
8. <B>In this Lesson...</B>
9.
10. <A HREF="#intro">Introduction</A>
11. <A HREF="#term">Volcano Terminology</A>
12. <A HREF="#usa">Volcanic Places in the USA</A>
13. <A HREF="#mars">Volcanic Places on Mars</A>
14. <A HREF="#project">Research Project</A>
15.
16. **Lưu** và **Reload** vào trong web browser của bạn. Khi bạn click trên một mục của bản mục lục, phần tương ứng sẽ hiện ra trên đỉnh của màn hình.

Thêm Liên kết tới Anchor Được đặt tên trong Tài liệu Khác

Bạn có thể tạo một liên kết nhảy đến một phần được đánh dấu bởi một anchor được đặt tên của một tài liệu HTML khác. HTML xây dựng một liên kết tới một anchor được đặt tên trong một tài liệu HTML cục bộ khác có dạng:

```
<a href="file.htm#NAME">Text to activate link</a>
```

và nếu là tài liệu trên web site khác thì có dạng:

```
<a href="http://www.cheese.org/pub/recipe.htm#colby">
                                Colby Cheese</a>
```

Trong bài 8a chúng ta đã tạo ra một liên kết siêu văn bản từ Mount St. Helens đến `msh.htm` là một tập tin riêng biệt. Bây giờ, chúng ta sẽ thêm một liên kết trong tài liệu thứ hai để trở lại phần nguyên thủy trong trang Volcano chính.

1. Mở tập tin `msh.htm` trong trình soạn thảo
2. Gần cuối trang HTML (nhưng phải trên `</body>`) thêm như sau:
3.
4. <hr>
5. Return to Volcano Web

   Chú ý: Chúng ta có sử dụng kiểu chữ *Nghiêng* với tag ... bên trong văn bản được đánh dấu bởi anchor được đặt tên là "usa".

   Lưu và **Reload** vào web browser của bạn. Khi click vào một vị trí của siêu văn bản ở cuối trang `msh.htm`, bạn sẽ phải thấy qua trở lại mục "Volcanic Plases in the USA" của bài *Volcano Web*.

Trong trường hợp đang xét liên kết chỉ là tên của một tập tin HTML khác, `index.htm`, trong cùng thư mục với tập tin `msh.htm`. Tuy nhiên, bạn có thể sử dụng một URL đầy đủ (xem bài 7) để liên kết tới một anchor được đặt tên trong một tập tin HTML có trên một máy khác ở xa. Ví dụ như, để tạo một liên kết đến phần "Introduction" của tập tin HTML cất trong MCLI WWW server, cách viết nên là:

```
<a href=
  http://www.mcli.dist.maricopa.edu/tut/final/index.htm#intro>
   Introduction to Volcano Web</a>
```

Dấu tham khảo `"#anchor_name"` được gắn thêm vào cuối của URL.

Kiểm tra lại công việc của bạn

So sánh công việc của bạn với ví dụ mẫu để biết tài liệu nên xuất hiện như thế nào. Nếu trang web của bạn không giống như ví dụ mẫu hay các liên kết anchor được đặt tên không hoạt động đúng, hãy kiểm tra lại văn bản bạn đã đánh trong trình soạn thảo.

Ví dụ sử dụng liên kết để trở lại phần mô tả anchor được đặt tên...

Xem lại

1. Làm thế nào nhận dạng một anchor được đặt tên?
2. Những bước nào tạo một liên kết tới một phần khác trong cùng một tài liệu?
3. Làm thế nào hiệu chỉnh một liên kết anchor để nối đến một anchor được đặt tên trong một tài liệu HTML khác?
4. Làm thế nào tạo một bảng mục lục cho một trang web?
5. Sự khác nhau giữa tag `<a href="...">` và tag `<a name="...">` là gì?

Thực tập tự do

Tạo các anchors được đặt tên cho các tiêu đề trong trang web và xây dựng một mục lục để liên kết đến những mục con này.

8e. Liên kết Siêu Hình ảnh

Không phải chỉ có văn bản mới làm được "hyper"... hãy mở rộng khả năng cho các trang web của bạn để các **hình ảnh** hoạt động như một siêu liên kết (hyperlink) (Thử click vào hình hộp!).

Mục đích

Sau bài này bạn có thể:

- Chèn một button hình trong trang web của bạn để nó liên kết tới một tài liệu HTML khác.
- Chèn một hình ảnh nhỏ hoạt động như là một "tem thư" (postage stamp) để liên kết đến việc trình bày một bản sao lớn của hình ảnh đó.

Bài học

Lưu ý: Nếu bạn chưa có tài liệu từ bài học trước, bạn có thể download lại ngay bây giờ.

Button siêu liên kết

Từ những bài học trước bạn đã dễ dàng (hy vọng là vậy) trong việc tạo ra siêu văn bản, tức là đoạn văn bản trong tài liệu của bạn cho phép người xem nối đến những thông tin có liên quan. Bạn cũng có thể làm cho hình ảnh inline (xem bài 7a) hoạt động theo cách giống như "hyper" này. Nếu bạn nhớ lại, trong bài học mới đây chúng ta đã liên kết một đoạn văn bản trong trang *Volcano Web*, `index.htm`, đến trang thứ hai là `msh.htm` để miêu tả Mount St. Helens. Bây giờ trong trang web này chúng ta muốn thêm một button mà khi được click vào sẽ đưa người đọc trở về bài học chính.

Cách để làm được điều này là đặt các tag HTML sao cho các hình ảnh inline *ở bên trong* phần siêu văn bản của tag anchor:

```
<a href="fileX.htm"> <img src="graphic.gif">
  Go to Document X</a>
```

Trong trang web của bạn HTML này sẽ hiển thị một hình ảnh inline và đoạn chữ `Go to Document X`. Cả hai sẽ hoạt động như một siêu liên kết; khi nhấn vào đoạn chữ hay vào hình ảnh sẽ báo cho web browser đi đến tập tin HTML `fileX.htm`. Riêng hình ảnh cũng có thể là một siêu liên kết. Nói chung, trong World Wide Web, một "HyperGraphic" sẽ được bao quanh bởi một hộp màu tương ứng với loại màu sử dụng cho siêu văn bản trong trang web của bạn, nhờ đó bạn biết rằng nó "hoạt động" (active).
Chú ý: Nhiều browser hiện nay có thể thay đổi màu của siêu văn bản và một số bỏ đi việc trình bày đường bao quanh các liên kết HyperGraphic. Nói chung, bạn có thể xác định một vùng siêu liên kết trong một trang web bằng cách xem sự thay đổi của dấu nháy (cursor) khi nó băng qua một vùng "nóng" (hot). Thông thường dấu nháy sẽ chuyển từ dấu mũi tên sang một bàn tay khi nó băng qua một liên kết hoạt động (active link).

Từ quan điểm của một nhà thiết kế, chúng tôi đề nghị khi bạn sử dụng một hình ảnh như là một siêu liên kết thì nên thêm một liên kết bằng văn bản hoặc là sử dụng thêm ALT= attribute trong tag <IMG...> để phục vụ cho trường hợp người xem tắt việc nạp các hình ảnh.

Bây giờ chúng ta sẽ tạo ra một button hình ảnh "hyper". Trước hết, bạn cần chép một bản sao của button mũi tên từ Trung tâm Download Hình ảnh Bài 8e.

Bạn nên có một bản sao của tập tin hình ảnh ở đâu đó trên máy của bạn (Bạn nên cất lại nó vào trong folder/directory `pictures` trong vùng làm việc của bạn).

Tiếp theo, thêm vào HTML để button hoạt động được:

1. Mở tập tin HTML thứ hai `msh.htm` trong trình soạn thảo văn bản.
2. Ở dưới cùng, điều chỉnh dòng cuối thành:
3.
4. <hr>
5. <img src="../pictures/left.gif">
6. Return to Volcano Web

   Lưu ý : Tag hình ảnh `inline` (<img...>) hoàn toàn nằm trong anchor giữa > đánh dấu kết thúc URL và đánh dấu sự kết thúc của siêu văn bản. Ngoài ra cũng chú ý tag được sử dụng trong siêu văn bản để nhấn mạnh tiêu đề của bài. Và cuối cùng, chúng ta đã sử dụng tag <hr> để tạo một dòng thẳng trên hình button (xem tag này trong bài 4).

7. **Lưu trữ** tập tin HTML.
8. Trở về web browser, **Open Local** từ thực đơn **File** để đọc tập tin `"msh.htm"`.
9. Chọn **Reload** từ thực đơn **File** để điều chỉnh các thay đổi.
10. Thử siêu liên kết đến trang Mount St. Helens và button mới để trả bạn lại trang *Volcano Web*.

Các hình ảnh "Tem thư"

Trước đây, chúng tôi có đề nghị không nên sử dụng các hình ảnh inline lớn trong trang web của bạn bởi vì người sử dụng phải mất thời gian chờ đợi để hình ảnh được download vào máy tính của họ. Một cách khác ở đây là tạo một bảng sao bé nhỏ của hình ảnh, hay được gọi là "tem thư" (postage stamps) để trình bày như là hình ảnh inline. Sau đó sử dụng những bước tương tự như trên bạn có thể làm cho "tem thư" hoạt động như một siêu liên kết đến hình với kích thước thật của nó. Theo cách này hình ảnh lớn thật sự **chỉ** được download nếu người xem quyết định muốn coi nó.

Trước hết bạn cần sao chép hai tập tin ảnh từ Trung tâm Download Hình ảnh Bài 8e. (Các tập tin này nên được cất vào folder/directory `pictures` trong vùng làm việc của bạn).

Sau đó, tạo các liên kết tem thư trong tập tin văn bản chính:

1. Mở tập tin `index.htm` trong trình soạn thảo
2. Dưới tiêu đề **Long Valley** đánh vào:
3.
4. This field seismometer measures earthquakes associated
5. with subsurface volcanic forces and may help to predict
6. future events. It sits on a plateau known as the "Volcanic
7. Tableland" formed by a major eruption 600,000 years ago.
8. <p>
9.
10. <img src="../pictures/stamp.gif">
11. -- [View the full size image] --
12. <p>
13. This seismometer measures earthquakes
14. associated with subsurface volcanic forces.

    Hình ảnh inline, `stamp.gif`, hoạt động như một siêu liên kết đến hình lớn hơn, `seismo.jpg`. Khi người sử dụng click trên "tem thư" hay trên đoạn văn "View the full size image --", browser của bạn sẽ trình bày ảnh lớn trong một cửa sổ riêng hay trong một trang trắng.

15. **Lưu** và **Reload** trong web browser của bạn.

Kiểm tra lại công việc của bạn

So sánh trang web của bạn với ví dụ mẫu để biết tài liệu nên xuất hiện như thế nào. Nếu trang web của bạn không giống như ví dụ mẫu, hãy kiểm tra lại văn bản bạn đã đánh trong trình soạn thảo. Lỗi thường gặp là sự khác biệt khi đánh vần tên tập tin và mã HTML cho liên kết anchor, hoặc là không có tập tin hình ảnh tại thư mục chỉ định. Nếu bạn thấy một biểu tượng hình với dấu chấm hỏi:

thì thường có nghĩa là HTML không tìm thấy tập tin có trong tag `<img>` hoặc là ảnh không thuộc dạng GIF hay JPEG.

Xem lại

Những vấn đề ôn lại của bài này là:

1. Làm thế nào tạo một button hình ảnh trong trang web của bạn?
2. Liên kết "tem thư" tiện lợi như thế nào khi thêm hình ảnh vào trang web của bạn?
3. Làm thế nào tạo ra liên kết "tem thư" trong tài liệu của bạn?

Thực tập tự do

Thêm các button để liên kết hai trang web lẫn nhau.

9. Preformatted Text

Làm thế nào bạn có thể hiển thị được những đoạn văn trong đó có xét đến các khoảng trắng và dấu xuống dòng?

display
text
where space and carriage return
characters count?

Mục đích

Sau bài học này bạn có khả năng:

- Tạo một bảng các văn bản đã được định biên
- Điều khiển vị trí của văn bản trên một trang khi tab và khoảng cách được xét đến

---

Bài học

Lưu ý: Nếu bạn không có tài liệu từ những bài học trước, bạn có thể download ngay bây giờ.

Trong những bài học trước chúng ta đã biết là một Web browser sẽ bỏ qua những ký tự khoảng trắng, ký tự tab, và dấu xuống dòng dư thừa trong tập tin HTML của bạn. Tuy nhiên có một vài trường hợp cá biệt mà việc trình bày trang với các tính chất đó trở nên quan trọng. Trong bài này, chúng ta sẽ xét một ví dụ mà ở đó ta sẽ trình bày văn bản thành các cột ngay hàng với nhau.

Tag `preformat` hướng dẫn cho Web browser trình bày văn bản **chính xác** như khi bạn đánh vào trong văn bản HTML, bao gồm cả việc thể hiện các khoảng trắng, các tab, và cả dấu xuống dòng. Một browser tiêu biểu sẽ hiển thị văn bản như vậy theo kiểu:

m o n o s p a c e d

Một ví dụ cho việc sử dụng tag `preformat` là:

```
<pre>
We have indented with 5 space characters.
And used the carriage return to jump
to
a
new
line.

    Here     we
    use
    spaces    to
    create    a
    text     table.
</pre>
```

Không có các tag `<pre>` và `</pre>`, thì đoạn HTML trên sẽ được trình bày như sau:

We have indented with 5 space characters. And used the carriage return to jump to a new line. Here we use spaces to create a text table.

Với tag `preformat`, sẽ dễ hơn khi trình soạn thảo bạn đang sử dụng có thể trình bày trong một phông mono-spaced (như là "Courier" hay "Monoco"); nếu không bạn cần phải đếm kỹ số khoảng trắng để chỉnh cột các đoạn văn (và bạn sẽ khá mệt khi làm chuyện đó).

Với bài Volcano của chúng ta, chúng ta muốn thêm vào một bảng bên dưới phần **Introduction** để liệt kê ra một vài núi lửa nổi tiếng, khi nào nó đã hoạt động lại, và mức độ hoạt động của nó. Để làm điều này:

1. Mở tập tin HTML thứ hai, `index.htm` trong trình soạn thảo văn bản.
2. Dưới phần cuối cùng của **Introduction**, đặt một tiêu đề mức 4 (<h4>) với dòng chữ: **`Volumes of Some Well-Known Volcanic Eruptions`** (Nếu bạn không nhớ cách tạo tiêu đề, hãy xem bài 3).
3. Bên dưới tiêu đề này, nhập **chính xác** văn bản sau (có thể thuận lợi hơn khi sử dụng chức năng cắt và dán từ trang web này!):
4.
5. <pre>
6. Eruption Date Volume in km^3
7. -------- ---- --------------
8. Paricutin, Mexico 1943 1.3
9. Mt. Vesuvius, Italy 79 A.D. 3
10. Mount St. Helen, Washington 1980 4
11. Krakatoa, Indonesia 1883 18
12. Long Valley, California pre-historic 500 - 600
13. Yellowstone, Wyoming pre-historic 2400
14. </pre>

    Trong ví dụ này, chúng ta sử dụng các khoảng trắng để làm cho cột đầu tiên được chỉnh biên trái (left justified) còn hai cột kia được chỉnh biên giữa (center-justified). Các dấu trừ để nhấn mạnh cột tiêu đề.

15. **Lưu trữ** và **Reload** trong web browser của bạn.

Kiểm tra lại công việc của bạn

So sánh trang Web của bạn với ví dụ mẫu để biết tài liệu nên xuất hiện như thế nào. Nếu trang web của bạn không giống như ví dụ mẫu, hãy kiểm tra lại văn bản bạn đã đánh trong trình soạn thảo. Nếu các cột không được chỉnh đúng, có lẽ bạn đánh thiếu hay thừa các khoảng trắng.

Xem lại

Những vấn đề ôn lại của bài này là:

1. Tag `preformat` làm thay đổi cách browser thông dịch HTML như thế nào?
2. Trong những trường hợp nào có thể bạn cần sử dụng tag này?

Thông tin bổ sung

Bạn vẫn có thể sử dụng các tag HTML bên trong văn bản đã được đánh dấu bởi tag `preformat`. Ví dụ như chúng ta có thể thêm một liên kết siêu văn bản và vài tag định kiểu khác **bên trong** cặp tag `<pre>...</pre>`:

HTML	Kết quả

<pre>
This tutorial is copyright
1995, 1996
	This tutorial is copyright
	1995, 1996

```
   <B>Be sure to tell all of
   your friends about it!</B>
      <i>We
         Welcome
           Your
             Feedback</i>
fondly,
<A HREF=
"http://www.mcli.dist.maricopa.edu/">
mcli</A>
</pre>
```

Be sure to tell all of your friends about it!
We
Welcome
Your
Feedback
fondly,
mcli

Chú ý rằng các tag HTML không được xem như là khoảng trắng; chúng được bỏ qua trong vùng preformat.
Một số tác giả của trang web sử dụng tag `<pre>...</pre>` với các ký tự xuống dòng bên trong để thêm những dòng trắng giữa văn bản và hình ảnh trong trang web của họ- nhất là trong trường hợp họ cần nhiều dòng trắng hơn so với tag <p>. Ví dụ như:

HTML	Kết quả

Cheese was long since
abolished from the Orient.
<pre>
 	
</pre>
...until Sir Longhorn arrived with the
great Cheese Crusade of 1167.
	Cheese was long since
	abolished from the Orient.
	
	...until Sir Longhorn
	arrived with the great
	Cheese Crusade of 1167.

Thực tập tự do

Thêm vào một bảng hay một sơ đồ trong tài liêu HTML của bạn sử dụng tag `preformat`.

10. Các kí tự đặc biệt

Làm thế nào để bạn nói...

>>>Ỉ đ ??? ?

Chú ý:Nếu các ký tự trên không thể hiện các dấu nhấn hay các dấu phụ, thì có nghĩa là web browser của bạn không hỗ trợ bộ ký tự ISO. Bạn nên bỏ qua bài này

Mục đích

Sau phần này bạn có thể:

- Sử dụng mã HTML cho những kí tự ISO Latin 1 để hiển thị những dấu nhấn cho những chữ không phải tiếng Anh (non-english letter).
- Viết những kí tự < and > trong HTML khi bạn cần nó trong một tài liệu.

  Thêm những khoảng trắng trong những chữ và từ trong một tài liệu

---

Bài học

Lưu ý: Nếu bạn không có tài liệu từ những bài học trước, bạn có thể download ngay bây giờ.

Các dấu nhấn

Đôi khi có thể bạn cần sử dụng một kí tự đặc biệt trong một tài liệu HTML, một dấu nhấn hay một dấu phụ. Các dấu này đã được biết đến theo ISO. Những kí tự đặc biệt như vậy được đánh dấu trong HTML dưới dạng:

```
&xxxx;
```

Trong đó XXXX là tên mã (code name) cho kí tự đặc biệt đó.

Ví dụ, trong phần "Terminology" của bài Volcano, chúng ta muốn thêm một sự giải thích cho một thuật ngữ kỹ thuật để mô tả một loại riêng biệt của tiến trình bùng nổ của núi lửa. Thuật ngữ đó là **nuee ardente** xuất phát từ một từ tiếng Pháp với nghĩa là "đám mây nóng rực" (glowing cloud) nhưng để phát âm đúng chúng ta cần một dấu nhấn "sắc" (acute), để từ đó xuất hiện là **nu? ardente**. Trong trường hợp này, chúng ta thay thế chữ **e** đầu tiên trong **nuee** bằng mã HTML của chữ "e" có dấu nhấn sắc **é**:

```
nu&eacute;e ardente
```

Để tham khảo, hãy xem danh sách các ký tự ISO.

Bây giờ chúng ta sẽ thêm một câu vào tài liệu HTML của chúng ta mà câu đó có sử dụng một ký tự có dấu nhấn:

1. Mở tập tin HTML là `index.htm` trong trình soạn thảo văn bản.
2. Bên dưới danh sách thuật ngữ của phần Volcanology Terminology thêm vào:
3.
4. ```
   The term <I>nuée ardente,</I> or
   ```
5. ```
   "glowing cloud" was first used by La Croix (1904)
   ```
6. ```
   in his description of the volcanic flows he observed in
   ```
7. ```
   the 1902 eruption of Mt Pel&eacute;e, a historically
   ```
8. ```
   active volcano on the island of Martinique.
   ```

   Chú ý: Chúng ta đã sử dụng dấu nhấn sắc cho hai ký tự "e" trong câu này. Trông có vẻ lạ quá! Hãy chắc chắn rằng bạn đã thay ký tự bằng chuỗi đã viết ở trên để hiển thị được cùng ký tự đó với dấu nhấn.

9. **Lưu** và **Reload** lại tập tin HTML.

Escape Sequences của HTML

HTML cho dấu nhấn là một ví dụ của một lớp các tag tổng quát hơn được gọi là **escape sequences**. Trong quá trình nhập văn bản HTML cho đến giờ, bạn có thể tự hỏi chúng ta sẽ làm gì khi cần sử dụng một dấu < (nhỏ hơn) hoặc một dấu > (lớn hơn)? Hai ký tự này cũng với dấu & (và) có một ý nghĩa đặc biệt trong HTML và không thể sử dụng như một ký tự đánh vào được. Thay vì đánh trực tiếp chúng, hãy sử dụng *escape sequences:*

```
<  thay cho <
>  thay cho >
& thay cho &
```
```

Bây giờ chúng ta hãy áp dụng các ký hiệu này trong bài Volcano của chúng ta. Trong bài trước: chúng ta đã thêm một bảng liệt kê vài núi lửa cùng với khối lượng vật chất đã phun ra từ chúng. Chúng ta nói rằng một trong những số liệu (500-600) cho Long Valley là không được chính xác lắm (thông thường đây là các giá trị gần đúng), và chúng ta muốn sửa đổi trị này thành >450 & <700. Để làm điều này:

1. Mở tập tin HTML `index.htm` trong trình soạn thảo
2. Bên dưới tiêu đề **Volumes of Some Well-Known Volcanic Eruptions**, tìm dòng Long Valley trong bảng:
3.
4. Long Valley, California pre-historic 500 - 600

   và đổi nó thành:

   Long Valley, California pre-historic >450 & <700

   Chú ý: Mặc dầu chúng ta sử dụng escape sequences bên trong một preformatted text, hãy chú ý một web browser vẫn thông dịch đúng và hiển thị các ký tự đặc biệt này. Như vậy, escape sequences có thể sử dụng trong mọi phần của một tài liệu HTML kể cả các tiêu đề và liên kết anchor.

5. **Lưu** và **Reload** tập tin HTML

Những khoảng trống thêm vào

Như bạn đã thấy một web browser sẽ bỏ qua tất cả những khoảng trắng không liên quan trong tập tin HTML của bạn. Tuy nhiên, có những lúc thật sự chúng ta muốn có nhiều hơn một khoảng trắng . Khi nào vậy? Một vài người thích có hai khoảng trắng theo sau một dấu chấm hết câu. Nếu bạn muốn thụt vào câu đầu tiên của mỗi đoạn văn bạn sẽ làm sao? Làm thế nào để có một từ đơn ở vị trí cách xa phần còn lại? Một mã HTML dành cho việc thêm vào một kí tự trắng là kí tự đặc biệt được gọi là "khoảng trắng không bị phá hủy" (non-breaking space):

Sau đây là một vài ví dụ về cách sử dụng khoảng trắng không bị phá hủy:

HTML	Kết quả
Khoảng trắng không bị phá hủy được dùng để trải dài các ký tự trong một từ	
<tt> C H E E S E </tt>	**C H E E S E**
Hai khoảng trắng không bị phá hủy được dùng để thụt các câu đầu của đoạn văn vào trong	
When Sir Longhorn had tragically died, no one was left to carry on his tradition. There was much sadness in the land.	When Sir Longhorn had tragically died, no one was left to carry on his tradition. There was much sadness in the land. And no cheese. But then the young genius Sheila Colby discovered the missing ingredient. And once again, cheese was plentiful.

```
And no cheese.
<p>
    But then
the young genius
Sheila Colby discovered
the missing
ingredient. And once
again, cheese
was plentiful.
```

Một khoảng trắng nữa được thêm vào sau cuối mỗi câu

```
    When
Sir Longhorn
had tragically died, no
one was left to
carry on his tradition.
 
There was much
sadness in
the land.  
And no cheese.
<p>
    But then
the young genius
Sheila Colby discovered
the missing
ingredient.   And
once again, cheese
was plentiful.
```

When Sir Longhorn had tragically died, no one was left to carry on his tradition. There was much sadness in the land. And no cheese.

But then the young genius Sheila Colby discovered the missing ingredient. And once again, cheese was plentiful.

Có thể bạn muốn có kinh nghiệm với các cách sử dụng khác nhau cho khoảng trắng không bị phá hủy. Bây giờ, chúng ta sẽ không sửa các tài liệu HTML của chúng ta, nhưng nếu muốn, bạn có thể thêm những mã để làm thụt đầu dòng các câu mở đầu của tất cả các đoạn văn qua việc sử dụng hai lần mã khoảng trắng không bị phá hủy.

Kiểm tra lại công việc của bạn

So sánh trang Web của bạn với ví dụ mẫu để biết tài liệu nên xuất hiện như thế nào. Nếu trang web của bạn không giống như ví dụ mẫu, hãy kiểm tra lại văn bản bạn đã đánh trong trình soạn thảo. Hãy bảo đảm bạn viết đúng các escape sequence với ký tự & và ;

Thông tin bổ sung

Dưới đây là một vài kí tự đặc biệt có thể hữu ích cho bạn:

Tên	HTML	Kết quả
Copyright	©	?
Trademark	®	?

Xem lại

Những vấn đề nên xem lại trong bài này:

1. Trong HTML, cách nào để hiển thị kí tự tiếng Đức "umlaut" (font>)?

2. Điều gì sẽ xảy ra nếu bạn không sử dụng một escape sequence cho < and >?
3. Tại sao cần một escape sequence cho kí tự &?
4. Làm thế nào có thể thụt các đoạn văn vào trong?

Thực tập tự do

Trong văn bản HTML của riêng bạn, thêm vào một từ nước ngoài mà từ đó cần một dấu nhấn đặt biệt, hay thêm vào một biểu thức toán học có sử dụng dấu < hay dấu >. Hoặc là bạn có thể thêm các khoảng trắng để thụt các đoạn văn vào trong.

11. Danh sách mô tả

Danh sách mô tả (Descriptive List) là một dạng đặc biệt của danh sách:

- **danh sách**
  - **danh sách**
    - **danh sách...**

Mục đích

Sau bài học này bạn có thể:

- Xây dựng danh sách các mục cùng với khối văn bản mô tả nó được thụt vào trong.
- Tạo một danh mục sách bằng danh sách mô tả.

Bài học

Lưu ý: Nếu bạn không có tài liệu từ những bài học trước, bạn có thể download ngay bây giờ.

Trong bài 6 chúng ta đã biết cách để tạo hai loại danh sách: có thứ tự <ol>... và không có thứ tự <ul>.... Bây giờ, chúng tôi sẽ giới thiệu một dạng thứ ba, **danh sách mô tả.** Không giống như những loại danh sách chúng ta đã thấy trước đây, danh sách mô tả đánh dấu các mục của nó không phải là một dấu chấm hoặc một số mà bằng dạng trình bày thụt vào trong của nó.

Dạng của tag danh sách mô tả là :

```
<dl>
<dt> description title1
<dd> description description1
<dt> description title2
<dd> description description2
            :
            :
            :
<dt> description titleN
<dd> description descriptionN
</dl>
```

Tag `<dl> .... </dl>` chứa trong nó nhiều cặp tiêu đề mô tả (description title) **<dt>** và sự mô tả (description) **<dd>** Một Web browser sẽ tạo thành một danh sách với mỗi mô tả được trình bày thụt vào so với tiêu đề của nó.

Khi nhìn trong Web browser ví dụ trên sẽ trông giống như sau:

```
description title1
        description description1
description title2
        description description2
:
:
description titleN
        description descriptionN
```

Một danh sách mô tả có thể được sử dụng như là một tự điển (glossary) các định nghĩa, nhưng trong ví dụ chúng ta, chúng ta sẽ dùng nó để tạo một thư mục sách ngắn cho bài *Volcano Web*:

1. Mở tập tin HTML `index.htm` trong trình soạn thảo.
2. Sau danh sách không có thứ tự của tiêu đề **References** nhập như sau :
3.
4. <h3>Bibliography</h3>
5. Check your library for these books:
6. <dl>
7. <dt>Cas, R.A.F. and Wright, J. V. (1987).
8. <dd><I>Volcanic Successions: Modern and Ancient.</I>
9. London: Allen & Unwin.
10.
11. <dt>La Croix, A. (1904)
12. <dd><I>La Montagna Pelée et ses Éruptions.</I>
13. Paris: Masson
14.
15. <dt>Lipman, P.W. and Mullineaux (eds). (1981)
16. <dd><I>The 1980 Eruptions of Mount St. Helens, Washington.
17. </I>U.S. Geological Survey Professional Paper 1250.
18. </dl>

    Chú ý: Chúng ta đã sử dụng một số Ký tự Đặc biệt cho dấu và ("&") trong tham khảo đầu tiên và dấu nhấn cho tham khảo thứ hai. Nếu bạn chưa quen với các ký tự đặc biệt của HTML, xem lại bài 10

19. **Lưu trữ** và **Reload** trong Web browser của bạn.

Kiểm tra lại công việc của bạn

So sánh tài liệu của bạn với ví dụ mẫu để biết tài liệu nên xuất hiện như thế nào. Nếu tài liệu của bạn không giống như ví dụ mẫu, hãy kiểm tra lại văn bản bạn đã đánh trong trình soạn thảo. Đừng quên các tag <dl>... </dl> để đánh dấu toàn danh sách. Lỗi thường gặp là quên chuyển đổi các tag <dt> và <dd>

Xem lại

Xem lại những chủ đề sau:

1. Danh sách mô tả khác với danh sách có thứ tự và danh sách không có thứ tự ở những điểm nào?
2. Trong trường hợp nào bạn có thể phải sử dụng trong một danh sách mô tả?
3. Sự khác nhau giữa tag `<dt>` và tag `<dd>`?

Thực tập tự do

Sử dụng danh sách mô tả để thêm một bảng thuật ngữ hay một danh mục sách trong trang HTML của riêng bạn.

Một số thông tin thêm

Bạn có thể thêm các danh sách có thứ tự và không có thứ tự vào bên trong một danh sách mô tả. Lấy ví dụ, chúng ta sẽ tạo một một danh sách các nhóm khoáng chất chính. Mỗi nhóm có sự mô tả các tính chất của chúng, cùng với một danh sách con các khoáng chất và chúng được sử dụng trong xã hội như thế nào. Chúng ta muốn nó có dạng như sau (chỉ trình bày một ít):

Oxides

Combinations of metal ions with Oxygen, comprises the major ores extracted in mining operations

- Hematite (iron ore)
- Magnetite (iron ore, magnetic mineral)
- Corundum (gemstone, abrasive)

Sulfates

Metal ions combine with the Sulfate ion (SO4), atomic structure sometimes can allow bonding of water molecules

- Gypsum (plaster)
- Barite (drilling mud)

Dạng HTML cho kết quả trên có dạng:

```
<dl>
<dt><b>Oxides</b>
<dd>Combinations of metal ions with Oxygen, comprises the major
ores extracted in mining operations
   <ul>
   <li>Hematite (iron ore)
   <li>Magnetite (iron ore, magnetic mineral)
   <li>Corundum (gemstone, abrasive)
   </ul>
<dt><b>Sulfates</b>
<dd>Metal ions combines with the Sulfate ion (SO4), atomic
structure sometimes can allow bonding of water molecules
   <ul>
   <li>Gypsum (plaster)
   <li>Barite (drilling mud)
   </ul>
</dl>
```

12. Danh thiếp Địa chỉ và Liên kết E-Mail

Trong bài này bạn sẽ?ạo ra một trang web đúng mốt -- tóm lược bằng một đoạn cuối! Hãy để cho mọi người gởi đến bạn một e-mail ngay từ trong trang Web của bạn!

Mục đích

Sau bài học này bạn có thể:

- Thêm một đoạn cuối (stylized footer) vào cuối trang Web.

- Tạo một liên kết siêu văn bản để gởi e-mail

Bài học

Lưu ý: Nếu bạn không có tài liệu từ những bài học trước bạn có thể download ngay bây giờ.

Một đặc điểm đáng giá của những trang Web là "footer" ở cuối của một trang để cung cấp thông tin về tác giả và tài liệu, có thể là ngày được cập nhật cuối cùng, cũng như là để gởi tới tác giả một thông báo bằng e-mail.

Đây chính là nơi để đặt tag `address` mà trong đó khối văn bản được in trong dòng mới dưới kiểu chữ nghiêng.

Thật là một ý tưởng hay để trong footer có một mô tả ngắn gọn, thông tin và quan hệ giữa những trang Web của bạn. Môt vài thông tin hữu ích là:

- Tiêu đề hay chủ đề của trang hiện thời.
- Ngày nó được tạo ra và cập nhật.
- Chủ quyền thích đáng.
- Tên và e-mail cho tác giả của trang Web.
- Tên và liên kết siêu văn bản tới nguồn gốc/tổ chức.

Như là một ví dụ, bạn hãy nhìn footer ở cuối mỗi trang Web của tài liệu hướng dẫn này. Trong quá trình sáng tạo ra những kiểu riêng của bạn, hãy nhìn thử những trang Web khác để thu thập thêm những ý kiến. ***Bắt chước vẫn là cách tốt nhất để tiến bộ!***

Dạng HTML của tag `address` trông giống như sau :

```
<address>
  text text text text text text text text text
  text text text text text text text text text
</address>
```

Lưu ý rằng tất cả những HTML khác bên trong tag `address` đều hợp lệ, vì thế chúng ta có thể sửa đổi nó với những tag in đậm, ngắt dòng và một tag liên kết siêu văn bản:

HTML	Kết quả
<address> Page Title Last Updated February 31, 1999 Web Page by Alan Levine (levine@maricopa.edu) Maricopa Community Colleges </address>	***Page Title*** *Last Updated February 31, 1999* *Web Page by Alan Levine (levine@maricopa.edu)* *Maricopa Community Colleges*

Bây giờ, giả sử một người nào đó đang đọc trang Web của bạn và muốn gởi cho bạn một lời phê bình về trang Web của bạn. Họ sẽ phải viết ra địa chỉ e-mail, nạp một chương trình khác, và gởi cho bạn một thông báo. Có thể gởi e-mail từ trong web browser của bạn được không? Được chứ, hầu hết các web browser hiện nay đều có thể làm điều này!

Cách để làm điều này là tạo một liên kết siêu văn bản với loại là `mailto` trong URL (hãy xem lại bài 8b). Tạo một liên kết siêu văn bản như sau:

```
<a href="mailto:levine@maricopa.edu">send an e-mail to alan</a>
```

Khi nhấn vào dòng chữ **send an e-mail to alan**, Web browser sẽ mở ra một màn hình để bạn có thể soạn một thông báo và gởi nó đến cho tôi. Hơn nữa, bạn còn có thể thêm một dòng chủ đề mặc nhiên (subject) cho thông báo e-mail:

```
<a href="mailto:levine@maricopa.edu? subject=hi from lesson 12">
    send an e-mail to alan</a>
```

Hãy thử đi! Hãy gởi cho tôi một thông báo! send an e-mail to alan

Lưu ý rằng bạn có thể làm cho bất kỳ một dòng chữ (hay hình ảnh) nào hoạt động như là một liên kết siêu văn bản. Vì vậy trong ví dụ vừa rồi, chúng ta sẽ sửa đổi HTML để có một địa chỉ internet tạo liên kết cho việc gởi e-mail.

HTML	Kết quả
<address> Page Title Last Updated February 31, 1999 Web Page by Alan Levine <A HREF="mailto:levine@maricopa.edu">(levine@maricopa.edu)</A> Maricopa Community Colleges </address>	***Page Title*** *Last Updated February 31, 1999* *Web Page by Alan Levine* *(levine@maricopa.edu)* *Maricopa Community Colleges*

Và cuối cùng chúng tôi đề nghị rằng bạn cũng nên đặt URL của trang vào trong footer. Tại sao vậy? Khi một ai đó in ra trang Web của bạn nhưng chưa viết lại hoặc lưu lại URL trong bookmark, có URL trên trang giấy cho phép một sự tham khảo bằng tay. Chỉ việc điều chỉnh ví dụ trên một chút (hãy chú ý đoạn HTML này ở **dưới** tag `<address>...</address>`):

<address> Page Title Last Updated February 31, 1999 Web Page by Alan Levine <A HREF="mailto:levine@maricopa.edu">(levine@maricopa.edu)</A> Maricopa Community Colleges </address> <p> <tt> URL: http://www.mcli.dist.maricopa.edu/tut/ </tt>	***Page Title*** *Last Updated February 31, 1999* *Web Page by Alan Levine* *(levine@maricopa.edu)* *Maricopa Community Colleges* URL: http://www.mcli.dist.maricopa.edu/tut/

Bây giờ đã đến lúc thêm một footer vào trang file HTML của bạn. Đối với ví dụ này, chúng tôi giả sử rằng bạn là "Lorrie Lava" một Nhà nghiên cứu núi lửa tại Big University (hoàn toàn tự do nếu bạn muốn đưa thông tin của bạn vào vị trí dưới đây):

1. Mở tập tin HTML `index.htm` trong trình soạn thảo của bạn.
2. Tại cuối trang Web (nhưng trên những tag </body></html>), thêm:
3.
4. <HR>
5. <address><B>Volcano Web</B>
6. created by Lorrie Lava,

```
7.  <A HREF="mailto:lava@pele.bigu.edu">
8.  lava@pele.bigu.edu</A><br>
9.  Volcanic Studies,
10. <A HREF="http://www.bigu.edu/">Big University</A><p>
11. <TT>last modified: April 1, 1995</TT>
12. </address>
13. <p>
14. <tt>URL: http://www.bigu.edu/web/index.htm</tt>
```

Lưu ý: Chúng ta đã sử dụng một vài tag HTML mà đã được xét đến trong các bài trước. Cũng nên lưu ý đến tag <p> ở cuối trang; điều này chắc chắn rằng dòng cuối của văn bản luôn luôn được thấy rõ.

15. **Lưu trữ** và **Reload** tập tin HTML.

Kiểm tra lại công việc của bạn

So sánh tài liệu của bạn với ví dụ mẫu. Nếu tài liệu của bạn trông khác với mẫu, hãy xem lại văn bản bạn đã nhập trong trình soạn thảo.

Xem lại

Xem lại những chủ đề sau:

1. Tag `address` dùng để làm gì?
2. Tag `address` phải ở cuối một trang phải không?
3. Làm thế nào bạn tạo một tag để gởi e-mail đến bạn? hoặc đến một người khác? hoặc cùng với dòng chủ đề?

Thực tập tự do

Thêm một danh thiếp địa chỉ cùng với liên kết e-mail vào tài liệu HTML của riêng bạn.

13. Blockquote !

Vẫn còn một tag HTML đơn giản khác để sắp xếp lại văn bản của bạn:
Tag `<blockquote>` hỏi các tag <p> và `<br>` **"Chúng ta làm gì đây?"**. Các tag này trả lời "Không rõ lắm, nhưng anh rất khác biệt so với chúng tôi!"

Mục đích

Sau bài học này bạn có khả năng:

- Chèn một khối văn bản được trình bày thụt vào trong so với phần thân văn bản.
- Áp dụng những tag định kiểu bên trong một văn bản được blockquote.

Bài học

Lưu ý: Nếu bạn chưa có tài liệu từ những bài học trước, hãy download ngay bây giờ.

Theo cách viết truyền thống, một đoạn trích dẫn của ba hoặc nhiều câu được làm nổi bật so với đoạn văn bản chính bằng cách trình bày như một khối văn bản được thụt vào bên trong

(indented block of text). HTML cũng có khả năng này bằng cách sử dụng tag `<blockquote>....</<blockquote>`:

```
<blockquote>
"This is a long quotation from a very famous person. Since it is
so long and interesting, it should really be set off from the
rest of the text. This indicates clearly that the quote is from
someone other than the writer."
</blockquote>
```

Được nhìn thấy như sau :

"This is a long quotation from a very famous person. Since it is so long and interesting, it should really be set off from the rest of the text. This indicates clearly that the quote is from someone other than the writer."

Bây giờ chúng ta sẽ áp dụng tất cả những gì của HTML chúng ta đã học trước đây **vào bên trong** tag `<blockquote>`, chẳng hạn như trong ví dụ sau đây:

```
<blockquote>
<H2>A Manifesto</H2>
This is a <B>long</B> quotation from a
<A HREF="http://www.mcli.dist.maricopa.edu/alan/">
very famous person</A>.
Since it is so long and interesting, it should really be
<pre>     set off</pre>
from the rest of the text.
<p>
<I>This indicates</I>:
<ul>
<li>clearly that
<li>the quote is from
<li>someone other than the writer.
</ul>
</blockquote>
```

cho kết quả như sau:

A Manifesto

This is a **long** quotation from a very famous person. Since it is so long and interesting, it should really be
    set off
from the rest of the text.

This indicates:

- clearly that
- the quote is from
- someone other than the writer.

Chúng ta hãy thêm một đoạn trích dẫn vào trang Volcano Web. Chúng ta sẽ sử dụng blockquote trong phần của nhà tự nhiên học người La mã Pliny, người đã chứng kiến sự hoạt động lại của núi lửa Vesuvius vào năm 79 sau công nguyên.

1. Mở tập tin HTML `index.htm` trong trình soạn thảo.
2. Bên dưới đề mục `<h1>Volcano Web</h1>`, thêm những dòng sau :
3.
4. <BLOCKQUOTE>
5. <B><I>
6. "Nature raves savagely, threatening the lands"
7. </I></B>
8. -- <A HREF="http://magic.geol.ucsb.edu/~fisher/pliny.htm">
9. Pliny the Elder</A>, who died of asphyxiation after
10. observing the destruction of Pompeii by the
11. 79 A.D. eruption of Mount Vesuvius.
12. </BLOCKQUOTE>

**Chú ý: hãy xem cách chúng ta sử dụng kết hợp cả in đậm và in nghiêng (xem bài 5) để làm sáng lên đoạn trích dẫn. Cách để sang dòng mới với tag
 (xem bài 4). Chúng ta cũng tạo siêu liên kết cho "Pliny the Elder" đến một web site khác có nhiều thông tin hơn về Pliny và các quan sát của ông ta.**

Cũng nên chú ý cách tag `<blockquote>` thêm dấu ngắt đoạn trước và sau khối văn bản.

Và cuối cùng, Các CHÚ Ý (giống như bạn đang xem) mà chúng tôi đã sử dụng trong các bài đều dùng đến tag `<blockquote>`!

13. **Lưu trữ** và **Reload** tập tin HTML.

Kiểm tra lại công việc của bạn

So sánh trang của bạn với ví du mẫu. Nếu tài liệu của bạn trông khác với mẫu, hãy xem lại văn bản bạn đã nhập trong trình soạn thảo.

Xem lại

Xem lại những vấn đề sau :

1. Tag `<blockquote>` dùng để làm gì?
2. Tại sao không cần tag <p> trước một blockquote?

Thực tập tự do

Hãy tạo kinh nghiệm với việc sử dụng tag `<blockquote>` trong trang Web của bạn. Đừng có nghĩ rằng tag này chỉ sử dụng cho việc trích dẫn. Nó có thể khá hiệu quả để làm cho sự trình bày trang web của bạn thêm đa dạng. Tag `<blockquote>` là một trong các cách để làm cho nhiều trang web không phải là những đoạn văn dài dòng vô vị.

Một vài tác giả sử dụng hai (hoặc nhiều hơn) tag `<blockquote>` lồng vào nhau để tạo ra hiệu ứng chừa lề. Ví dụ như:

```
<BLOCKQUOTE>
 <BLOCKQUOTE>
  <BLOCKQUOTE>"Oh, for more tags," he lamented.
   <p>Replied she, "Remember, dear, that while HTML may
   be limited in terms of control
   over page layout, there is a great
   deal of potential in creatively use
```

```
    of the number of tags."
   </BLOCKQUOTE>
  /BLOCKQUOTE>
</BLOCKQUOTE>
```

Được hiển thị như sau:

"Oh, for more tags," he lamented.

Replied she, "Remember, dear, that while HTML may be limited in terms of control over page layout, there is a great deal of potential in creatively use of the number of tags."

14. Gộp lại (Lumping) hay Phân mảnh (Splitting)

Bạn đã làm được một home page! Một việc lớn! (Nhưng này bạn! Đây mới chỉ là công việc giống như Hercule thôi!) BẤY GIỜ, những người bạn của tôi ơi, đây là lúc để chuyển một "trang" văn bản dài thành những đoạn thông tin của "web" được nối với nhau.

Mục đích

Sau bài này, bạn có thể:

- Chuyển một trang web đơn thành một chuỗi các trang được liên kết với nhau
- Tạo ra một trang mẫu (template) cho nhiều trang web
- Xây dựng các đặc tính hoa tiêu (navigational feature) để nối nhiều trang web lại với nhau

---

Bài học

Lưu ý: Nếu bạn chưa có tài liệu từ những bài học trước, hãy download ngay bây giờ.

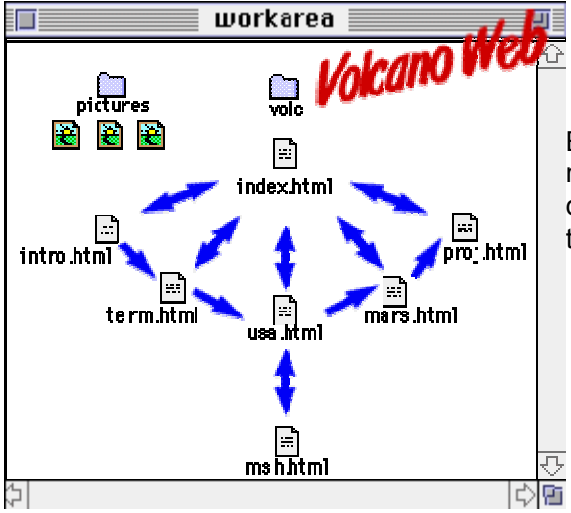
Bạn làm một **người gộp lại** hay là một **người chia mảnh**? Không phải? Hay cả hai?

Để tổ chức thông tin, đôi lúc tốt hơn hết là "gộp" chúng lại với nhau; đôi lúc tốt hơn hết lại là "chia" chúng thành các phần riêng biệt. Kéo lên xuống trong các trang web dài thường tạo ra sự chán nản. Khi dài, một trang web cần tốn nhiều thời gian để nạp từ mạng xuống nếu so sánh với việc chia nó thành những trang nhỏ hơn.

Trong nhiều trường hợp, bạn có thể xác định những điểm luận lý để "chia" thông tin thành nhiều trang web. Tuy nhiên, điều này không có một công thức chung, và ý kiến cũng không giống nhau. Bạn nên tạo sự cân bằng giữa các trang bằng cách chia nội dung thành các phần đều nhau để tránh cho người đọc phải click quá nhiều khi muốn đến một nội dung nào đó. Cũng là một điều quan trọng khi xây dựng các liên kết siêu văn bản hay các manh mối dễ nhìn thấy bên trong trang web giúp người đọc tìm được thông tin cần thiết.

Trước đây chúng ta vừa tạo được một trang web với một liên kết đến một trang ngắn hơn. Trong bài 8d chúng ta đã tạo ra một danh sách các liên kết như là một mục lục để nối chúng đến các anchor được đặt tên cho các phần khác nhau của bài *Volcano Web*. Sự phân đoạn theo cách như vậy có thể được xem là các điểm ngắt để phân mảnh một trang đơn dài thành các trang con.

Cho đến giờ chúng ta đã tạo ra một thư mục gọi là `volc` trong đó có chứa hai tập tin HTML (trang bài học `index.htm` và trang thứ hai `msh.htm`). Chúng ta cũng đã tạo ra một thư mục thứ hai là `pictures` để giữ các tập tin hình ảnh.



Bây giờ chúng ta sẽ phân chia tập tin Volcano Web đơn thành một chuỗi các trang web được liên kết với nhau như trong sơ đồ này. Điểm vào của nó là trang mục lục/bìa tại `index.htm` trong đó có liên kết đến tất cả các phần khác của bài học:

- **Introduction** `[intro.htm]`
- **Volcano Terminology** `[term.htm]`
- **Volcanic Places in the USA** `[usa.htm]`
- **Volcanic Places on Mars** `[mars.htm]`
- **Research Project** `[proj.htm]`

Mỗi phần của bài học sẽ liên kết ngược lại phần mục lục và trang trước của nó. Cũng nên chú ý đến liên kết hai chiều giữa `usa.htm` và `msh.htm`

Chú ý: Để hoàn thành bài học này, chúng ta cần tạo thêm một số tập tin mới và cũng phải thực hiện nhiều lần thao tác chép/dán từ tập tin bạn đang làm việc. Hãy chắc chắn rằng bạn biết rõ cách chuyển qua lại giữa các chương trình và cửa sổ tài liệu khác nhau trên máy của bạn.

Điều đầu tiên chúng ta cần làm là tạo một tập tin `index.htm` mới, được sử dụng làm trang "bìa" của bài *Volcano Web*:

1. Trước hết tạo một bản sao của tập tin `index.htm` và đặt tên là `old.htm` hay là một tên nào tương tự như vậy.
2. Bây giờ mở lại tập tin `index.htm` nguyên gốc trong trình soạn thảo. Từ điểm này trở đi, đây sẽ là bài mới của chúng ta.
3. Chúng ta sẽ sử dụng lại hình ảnh, lời mở đầu, và bảng mục lục trong trang đầu tiên này. Để làm điều đó, chúng ta sẽ xóa đi các phần cần "cắt" ra khỏi trang web.

   Xóa bỏ đoạn từ **Introduction** đến **References**, tức là mọi thứ trong:

   ```
    <hr>
   <h2><A NAME="intro">Introduction</A></h2>
   A <b>volcano</b> is a location where magma, or hot
   melted rock from within a planet, reaches the surface.
    :
   ```

   và

   ```
     :
   <dt>Lipman, P.W. and Mullineaux (eds). (1981)
   <dd><I>The 1980 Eruptions of Mount St. Helens, Washington.
   </I>U.S. Geological Survey Professional Paper 1250.
   </dl>
   ```

   Bạn có thể so sánh tập tin HTML của bạn với ví dụ mẫu để biết trang nên như thế nào tại điểm này.

4. Bây giờ tìm đoạn văn có "In this Lesson..." Trước đây, chúng ta đã dùng các liên kết siêu văn bản đến anchor được đặt tên (tức là `<a name="term">..</a>`) trong cùng một tài liệu (xem bài 8a). Bây giờ chúng ta sẽ điều chỉnh các liên kết anchor này để liên kết đến một trang web khác (mà chúng ta sẽ tạo ra dưới đây).

Tìm đoạn có dạng:

```
<B>In this Lesson...</B>
 <ul><i>
 <li><A HREF="#intro">Introduction</A>
 <li><A HREF="#term">Volcano Terminology</A>
 <li><A HREF="#usa">Volcanic Places in the USA</A>
 <li><A HREF="#mars">Volcanic Places on Mars</A>
 <li><A HREF="#project">Research Project</A></i>
 </ul>
```

và điều chỉnh thành:

```
<B>In this Lesson...</B>
 <ul><i>
 <li><A HREF="intro.htm">Introduction</A>
 <li><A HREF="term.htm">Volcano Terminology</A>
 <li><A HREF="usa.htm">Volcanic Places in the USA</A>
 <li><A HREF="mars.htm">Volcanic Places on Mars</A>
 <li><A HREF="proj.htm">Research Project</A></i>
 </ul>
```

Bạn có thể so sánh tập tin HTML của bạn với ví dụ mẫu để biết trang nên như thế nào tại điểm này.

Chú ý: Bạn cần hiểu rõ sự khác nhau giữa một liên kết có dạng:

```
<a href="#quest">go to questions</a>
```

với một liên kết khác có dạng:

```
<a href="quest.htm">go to questions</a>
```

Việc kế tiếp cần làm là tạo các tập tin riêng cho các phần trong bài của chúng ta. Sẽ dễ dàng hơn khi trước hết chúng ta tạo ra một tập tin mẫu (template file) rồi sau đó hiệu chỉnh cho các trang web khác nhau.

1. Trong trình soạn thảo, tạo một tập tin mới gọi là `temp.htm`
2. Trong tập tin này, đánh vào đoạn HTML sau (nếu muốn, bạn có thể chép lại một tập tin ví dụ mẫu):

HTML	Ghi chú
<html> <head> <title>XXXXXXXX</title> </head> <body>	**HEAD:** Trong phần đầu của mỗi tài liệu, XXXXXXXX là tên của phần đó
<H5>Volcano Web / <A HREF="index.htm">Index</A> / <A HREF="xxxx.htm">back</A> / <A HREF="xxxx.htm">next</A></H5>	**NAVIGATION:** Tại đỉnh mỗi trang chúng ta sử dụng một tiêu đề nhỏ (h=5) để tạo ra các liên kết hoa tiêu (navigation link). Điểm `Index` để trở về trang bìa chính. Liên kết `next` và `back` để đến trang kế tiếp và trang trước. Bạn sẽ phải điền lại tên tập tin thích hợp cho `xxxx.htm`. Hãy chú ý cách này tạo ra một hướng dẫn nhỏ cho mỗi trang web.

XXXXXXXX : : :	**HEADER:** Sử dụng tiêu đề mức 2 cho các mục của trang.
<hr> <ADDRESS> <A HREF="index.htm"> Writing HTML</A> : XXXXXXXX <p> created by Lorrie Lava, <A HREF="mailto:lava@pele.bigu.edu"> lava@pele.bigu.edu</A> Volcanic Studies, <A HREF="http://www.bigu.edu/"> Big University</A><p> <TT>last modified: April 1, 1995</TT> </ADDRESS> <p>	**ADDRESS FOOTER:** Chú ý cách footer chỉ ra tên của trang chính (với một liên kết ngược về nó) cùng với đoạn văn chỉ ra tên của phần hiện tại XXXXXXXX. Đặt tên trang ở đây tạo ra một hướng dẫn nhỏ cho thấy vị trí của nó trong cấu trúc web mà chúng ta đã tạo ra.
<tt>URL: http://www.bigu.edu/web/xxxxxxxx.htm </tt> <p <body> </html>	**URL:** Cần điều chỉnh dòng này để chỉ ra URL của tài liệu tương ứng với tên tập tin `xxxxxxxx.htm` của nó

3. Đến đây, bạn cần tạo ra 5 bản sao của tập tin mẫu và thực hiện các điều chỉnh thích hợp trong từng mẫu:

Tên Tập tin	Mục	Ghi chú
intro.htm	**Introduction**	Vì đây là mục đầu tiên, xóa đi dòng <A HREF="xxxx.htm">back</A> trong phần hoa tiêu
term.htm	**Volcano Terminology**	
usa.htm	**Volcanic Places in the USA**	
mars.htm	**Volcanic Places on Mars**	
proj.htm	**Research Project**	Vì đây là mục cuối cùng, xóa đi dòng <A HREF="xxxx.htm">next</A> trong phần hoa tiêu

4. Bây giờ, hãy mở tập tin `index.htm` cũ (mà chúng ta đổi tên là `old.htm`) trong trình soạn thảo. Với mỗi tập tin mới tạo, bạn cần chép lại (copy) đoạn HTML phía dưới đề mục <h2>...</h2> và dán (paste) vào tập tin mới tương ứng. Chú ý rằng trong **Volcanic Places in the US** và **Research Projects** cả hai đều có các mục con sử dụng tiêu đề <h3>...</h3>.

5. Cuối cùng, bạn cần điều chỉnh liên kết trong tập tin `msh.htm`. Trước đây, nó quay về một anchor được đặt tên trong bài chính (mục **Volcanic Places in the US**) còn bây giờ nó cần liên kết đến tập tin `usa.htm`. Mở tập tin `msh.htm` trong trình soạn thảo và điều chỉnh:

```
    <a href="usa.htm">
    <img src="../pictures/left.gif" alt="** ">
    Return to
  <i>Volcano Web</i></a>
```

Cũng để cho tương đồng, bạn nên điều chỉnh lại phần footer như sau:

```
<HR>
<ADDRESS>
<B><A HREF="index.htm">
Volcano Web</A> : <A HREF="usa.htm">
Volcanic Places in the USA</A> :
Mount St. Helens</B> <p>

created by Lorrie Lava,
<A HREF="mailto:lava@pele.bigu.edu">
lava@pele.bigu.edu</A><br>
Volcanic Studies,
<A HREF="http://www.bigu.edu/">
Big University</A><p>
<TT>last modified: April 1, 1995</TT>
</ADDRESS>
<p>
<tt>URL: http://www.bigu.edu/web/msh.htm</tt>

<body>
</html>
```

Kiểm tra lại công việc của bạn

So sánh trang của bạn với ví dụ mẫu. Nếu tài liệu của bạn trông khác với mẫu hay các liên kết siêu văn bản hoạt động không đúng, hãy xem lại văn bản bạn đã nhập trong trình soạn thảo. Trong bài này chúng ta đã tạo ra nhiều tập tin do vậy rất dễ có lỗi do đánh sai.

Xem lại

Các vấn đề nên xem lại với bài này:

1. Cho biết vài tiện lợi của các trang web ngắn so với một trang web dài?
2. Điều gì xảy ra nếu chúng ta không sửa lại liên kết siêu văn bản trong tập tin `msh.htm`?
3. Các đặc tính hoa tiêu mà chúng ta đã đưa vào bài là gì?

Thông tin bổ sung

Về nguyên tắc, trang web của bạn sẽ dễ đọc hơn khi các liên kết siêu văn bản đồng nhất vào nội dung của văn bản quanh nó. Điều này trở nên quan trọng hơn khi bạn tạo ra nhiều trang web có các siêu văn bản liên kết lẫn nhau. Hãy so sánh:
In the spring of 1980, most people living in the vicinity of Mount St. Helens took heed to the scientists warning about an impending volcanic eruption. However, several were insistent on staying in their homes and sadly perished in the May 18 event. In that same year, measured increases in seismic recording devices caused scientists to warn of a possible event in Long Valley, California, and order a large evacuation of the Mammoth resort area. However, no

such event occurred, and residents were angrily resentful for what they perceived as a false warning that caused great economic loss.
với
In the spring of 1980, most people living in the vicinity of Mount St. Helens took heed to the scientists warning about an impending volcanic eruption. (Click here to see a picture of Mount St. Helens) However, several were insistent on staying in their homes and sadly perished in the May 18 event. In that same year, measured increases in seismic recording devices caused scientists to warn of a possible event in Long Valley, California, and order a large evacuation of the Mammoth resort area. (Click here to see a seismometer) However, no such event occurred, and residents were angrily resentful for what they perceived as a false warning that caused great economic loss.
Các dòng `"Click here..."` không làm phá hủy ý nghĩa của dòng văn bản, nhưng đoạn liên kết `"here"` không có quan hệ gì với mục liên quan. Như là một đề nghị, tránh viết các đoạn tương tự như `"click here to return to the home page"`. Thay vì vậy chỉ viết rõ một liên kết, tức là `<a href="home.htm">Home Page</a>` - bản thân việc click đã được hiểu ngầm khi sử dụng web browser.

Thực tập tự do

Hãy xem lại trang web bạn đang thực hiện. Nó có quá dài không? Có thể có những đoạn mà bạn có thể "tách" trang không? Hãy tạo ra trang bìa/mục lục và các liên kết thích hợp cho phần liên kết hoa tiêu giữa chúng. Sau đó tạo mẫu cho các "trang con".

Bây giờ hãy yêu cầu các bạn hay đồng nghiệp của bạn xem các trang web của bạn. Họ có thích cách "phân" trang hay "ghép" nhóm của bạn không? Họ có dễ dàng biết được cách để xem thông tin của bạn không?

15. HTML "tiêu chuẩn" và "nâng cao"

Hãy xét đến một thực tế của HTML
--nhiều điều bạn có thể thực hiện hơn nhưng chúng cũng làm cho HTML trở thành không thỏa "tiêu chuẩn" hơn.

Mục đích

Đây là một bài học ngắn và không cần thực tập gì hết! Sau bài này bạn có khả năng:

- Quyết định khi nào dùng những đặc điểm của HTML mà có thể không hoạt động được cho tất cả web browser.

Bài học

Trong bài 0, chúng ta đã giới thiệu về HTML "tiêu chuẩn". Trong những bài học tiếp theo trong tài liệu này sẽ chỉ cho bạn cách sử dụng một số tag HTML khác mà có thể không thấy được trên tất cả các Web browser. Ở đây chúng ta chỉ xem qua một vài ý tưởng trước khi bạn bước vào những bài sau.

Trước hết chúng ta hãy xem sơ qua về lịch sử của nó.

Cách đây khá lâu ("trong một thiên hà rất xa?")... vào khoảng năm 1990, World Wide Web là hệ thống được xây dựng trên văn bản có nền tảng dựa trên ngôn ngữ HyperText Markup Language. Các tag và sự thông dịch của nó dựa trên những chuẩn (HTML 1.0) đặt bởi một tổ chức quốc tế (international committee). Đây chính là chìa khóa để cho "Web" trở nên "phổ biến" (world wide) bởi vì theo những tiêu chuẩn này, thông tin hoàn toàn độc lập với hệ thống máy tính dùng để xem nó.

Ngay cả khi NCSA Mosaic xuất hiện một cách đột ngột vào năm 1993 như là một web browser đồ thị đầu tiên, những tiêu chuẩn vẫn được dựa trên ký tự và được cập nhật thành HTML 2.0.

Web trở nên phổ biến,

phổ biến rộng rãi,

phổ biến một cách điên rồ.

Những nhà lập trình bắt đầu xây dựng những Web browser có cung cấp những chức năng giống như Mosaic (bởi vì họ phải hỗ trợ tất cả các tính chất của HTML có trong tiêu chuẩn quốc tế). Một nhóm người trong đó có các nhà phát triển Mosaic đã thành lập một công ty mới có biểu tượng là "Mozilla" ("Mosaic" + "Godzilla"?) và tạo ra một web browser mới tên là NetScape.

Netscape đã nhanh hơn so với NCSA Mosaic. Tuy nhiên, nó phát triển trong đại chúng vì nó chứa đựng tất cả những tính chất của HTML 2.0 CỘNG với nhiều tag thêm vào cho những thứ mà bạn không thể thực hiện được trong HTML 2.0. Phần "mở rộng" (extensions) hay "nâng cao" này (enhancements) đã là nguyên nhân (và vẫn là một nguyên nhân) của sự tranh cãi lớn giữa những người làm ra chuẩn HTML và những người thích các đặc điểm mà Netscape thêm vào.

Mozilla Netscape đã hết sức phổ biến và nhanh chóng chiếm lĩnh 3/4 thị trường Web browser. Bây giờ, trong HTML, bạn có thể đặt màu nền cho trang của bạn, tạo bảng, đặt văn bản quanh hình ảnh, và còn nhiều nữa. Bạn cũng đã từng nhìn thấy những trang web có câu **"This page optimized for NetScape"**. Nhiều Web browser khác đã bắt đầu hỗ trợ những đặc tính của Netscape "HTML 2.0+". Cùng với các dịch vụ giúp đỡ trực tuyến được mở ra trên Web, thị trường browser ngày càng trở nên đông đúc (và hỗn độn).

Ủy ban quốc tế đã đối đầu với một tình thế lưỡng nan, cùng với việc thị trường trở nên rộng lớn là việc đòi hỏi những tag "không chuẩn" trở nên một phần của HTML. Khi những qui tắc cho HTML 3.0 được phát triển, họ bắt đầu đưa vào hầu hết (nhưng không phải là tất cả) những tag đã được Netscape giới thiệu. Tuy vậy, quá trình chuẩn hoá xem ra là quá chậm đối với nhiều người.

Và cuộc chiến trở nên lớn hơn khi vào năm 1996 Microsoft đã giới thiệu những tag HTML riêng biệt của họ. Như vậy HTML có trở thành một mớ hỗn độn không? Để biết thêm thông tin về sự cạnh tranh này, chúng tôi đề nghị bạn tham khảo đến HTML 3.0 and NetScape. Những hoạt động gần đây nhất được đề nghị là của HTML 3.2, mà trong đó chứa đựng hầu hết các đặc điểm được hỗ trợ bởi những nhà sản xuất lớn trên thị trường hiện nay, đó là Netscape và Microsoft. Các phát biểu gần đây nhất là sự thảo luận về các chuẩn nên được hỗ trợ trong tất cả các browser.

Nhưng những điều đó thì có ý nghĩa gì với bạn? Đó là điều quan trọng nhất, khi bạn là một nhà thiết kế trang Web, bạn cần phải biết loại browser nào mà người đọc sẽ phải sử dụng. Có lẽ bạn là một thầy giáo trong một trường học hoặc bạn đang làm việc trong một công ty mà tất cả mọi người đang sử dụng cùng một loại browser nào đó. Vậy thì bạn có chắc chắn rằng các tag đặc biệt bạn đang sử dụng là không có vấn đề không.

Tuy nhiên, thông thường bạn vẫn "xuất bản" (publishing) các trang Web vào một Internet server mà không có ý kiến gì về loại browser sẽ được sử dụng. Bạn có thể thêm những khuyến cáo đặc biệt trong trang Web của bạn. Bạn có thể tuân theo một cách chặt chẽ với những chuẩn được hỗ trợ rộng rãi bởi tất cả các Web browser. Ngay cả khi bạn sử dụng những tag đặc biệt, cũng có vài cách thông dụng để không gây ra sự tàn phá nào cho những người sử dụng các browser khác.

Điều quan trọng nhất ... không phải là việc quan tâm đến trang đó trông như thế nào trên hệ thống của bạn! Người đọc có thể có những Web browser khác nhau, có những phông chữ khác nhau, sử dụng những tham khảo màu khác nhau, kích cở màn hình khác nhau.. tất cả những thứ đó có thể gây ra sự hiển thị khác nhau về kích thước, về dạng trình bày so với máy của bạn. Nếu bạn có thể thử trang Web trên các máy tính khác nhau, hãy phóng to và thu nhỏ cửa sổ của browser, chuyển đổi phông chuẩn.

Thật là may mắn, thiết kế nguyên thủy của HTML rất thoáng và dễ tha thứ cho một số các luật - nếu một browser bắt gặp một tag mà nó không biết xử lý hay trình bày như thế nào, nó dễ dàng bỏ qua các tag này.

Ví dụ như, xem như là browser của tôi có hổ trợ tag `<drip>...</drip>`. Nghĩa của nó là đoạn văn bên trong sẽ xuất hiện bình thường rồi sau đó từ từ "rơi" xuống đáy của trang **(người biên tập - TÔI THÌ LÀM CHO NÓ ĐI LÊN!)**, một hiệu ứng lạ lùng cho trang web của tôi:

```
<drip><H2 align=center>Welcome</H2></drip>
to my sloppy home page!. Look out
for the puddles!
```

hoạt động của nó tạo ra một sự duyên dáng trên browser tự tạo của **tôi**. Trên browser của bạn không có hổ trợ đặc tính này, bạn sẽ thấy:

Welcome

to my sloppy home page!. Look out for the puddles!

Khi browser của bạn không hổ trợ tag này nó chỉ hoàn toàn băng qua tag mà không tạo ra sự phá hủy hay thông báo sai nào.

Đó là một ý tưởng thông minh, đúng không?

Khi chúng ta tiếp tục các bài học nâng cao này, các hướng dẫn hơi dài và phức tạp hơn. Nhưng bạn sẽ tiếp tục, đồng ý chứ!

Xem lại

1. Tại sao bạn phải quan tâm đến các tiêu chuẩn của HTML ?
2. Trường hợp nào bạn cảm thấy thoải mái khi sử dụng HTML không chuẩn?
3. Điều gì xảy ra nếu một web browser bắt gặp một tag mà nó không hiểu ?

16. Màu sắc và Cấu trúc cho Nền

Đừng bao giờ để cho trang Web của bạn có một màu xám xịt! Hãy đặt m à u S ắ c hoặc bố trí một cấu trúc nền nào đó "phía sau" phần văn bản.

Mục đích

Sau bài học này bạn có khả năng:

- Tạo một màu nền cố định cho trang Web.
- Tính được mã dưới hệ thập lục phân cho một giá trị màu.
- Thay đổi màu của văn bản và những mục liên kết siêu văn bản.
- Tạo một cấu trúc nền từ một tập tin hình ảnh.

Bài học

Lưu ý: Nếu bạn không có tài liệu của bạn từ bài học trước, hãy download ngay bây giờ.

Nền (background) của trang Web - chính là phần nền. Khi chúng ta thêm những màu sắc khác nhau hoặc ngay cả những mẫu nền, bạn nên nhớ để nó không ảnh hưởng đến việc đọc của phần văn bản. Với các trang trong tài liệu hướng dẫn này, chúng tôi sử dụng một màu trắng cố định cho nền để tạo ra sự rõ ràng và phông nền không gây ra ảnh hưởng đến nội dung.

Với một vài bổ sung cho tag <body> (được giới thiệu trong bài 1), bạn có thể tạo ra màu nền cho trang Web của bạn. Nhưng trước khi chúng tôi chỉ cho bạn cách tạo ra màu sắc thích thú như vậy, chúng ta phải nói sơ qua các giá trị màu RGB (red green blue) và sự thể hiện của chúng dưới dạng số "thập lục phân".

"Hex-Dec" và cơ bản về màu sắc

Trong một web browser, bạn được tùy ý sử dụng 256 màu của hệ thống để tạo màu cho văn bản và nền. Mỗi một màu được xác định dựa trên các giá trị **Red**- **Green**- **Blue** (RGB) của nó, ba số này có giá trị từ 0 đến 255, mỗi một số thể hiện cường độ của thành phần màu **Red**, **Green**, hay **Blue** cho màu đang xét. Giá trị lớn nhất của cả ba (R=255, G=255, B=255) cho ra màu **trắng** và giá trị nhỏ nhất (R=0, G=0, B=0) cho ra màu **đen**. Tất cả những màu khác được biểu diễn bởi các bộ ba RGB khác nhau.

Tiếp theo đây là một phần phức tạp. Thay vì xác định một màu theo dạng tương tự như "102,153,255" thì mỗi số được chuyển từ dạng biểu diễn theo cơ số hệ mười (là những số đếm hàng ngày dùng các ký số 0, 1, 2, 3, 4, 5, 6, 7, 8, 9) sang dạng biểu diễn theo cơ số hệ thập lục phân (dùng các ký số 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F). Tại sao vậy? Vì máy tính dễ sử dụng và dễ hiểu số hệ thập lục phân hơn. Vì vậy đối với màu cho ví dụ trên, chúng ta sẽ viết trong dạng số thập lục phân là: "6699FF". Trong ví dụ này, "66" là trị của màu **Red**, "99" là trị của màu **Green**, và "FF" là trị của màu **Blue**.

Sau đây là một vài ví dụ trị hệ thập lục cho các màu khác nhau:

Màu mẫu	Mã 16
xx oo xx FFCCCC	xx oo xx 3300FF
xx oo xx 33FF66	xx oo xx AA0000
xx oo xx 663300	xx oo xx 9900FF
xx oo xx 000077	xx oo xx FFFF00
xx oo xx EEEEEE	xx oo xx 888888
xx oo xx 444444	xx oo xx 000000

Nhưng, đừng lo sợ về việc chuyển đổi hệ thống số đếm! Có rất nhiều công cụ trong đó bạn chỉ cần click vào một bảng màu và chúng sẽ cho biết cách biểu diễn ở hệ thập lục phân. Rất nhiều công cụ màu như vậy có thể tìm thấy trong Yahoo.

Nhưng tốt hơn thế nữa, bây giờ rất nhiều browser hỗ trợ nhanh chóng cho 16 màu sau (chúng là các màu của VGA trên Windows):

xx oo xx aqua	xx oo xx black
xx oo xx blue	xx oo xx fuchsia
xx oo xx gray	xx oo xx green
xx oo xx lime	xx oo xx maroon
xx oo xx navy	xx oo xx olive
xx oo xx purple	xx oo xx red
xx oo xx silver	xx oo xx teal
white	xx oo xx yellow

Màu nền cố định

Chú ý: Trước hết bạn có thể thử kiểm tra để biết được browser của bạn có hỗ trợ những màu nền cố định không (solid color background).
Với Volcano Web của chúng ta, điều đầu tiên chúng ta sẽ làm là thêm một màu nền vào tập tin index.htm. Trong HTML để thêm một màu nền cố định cần điều chỉnh tag <body> như sau:

```
<body bgcolor=#XXXXXX>
```

trong đó XXXXXX là dạng biểu diễn thập lục phân (được chỉ định bởi dấu # ở phía trước) của màu được chỉ định.

Nếu bạn nhớ lại, hình chúng ta sử dụng trong phần mở đầu là hình núi lửa trên một nền đen - vì vậy nếu chúng ta cũng sử dụng màu đen cho màu nền của trang Web, hình ảnh sẽ rất hòa hợp với trang Web của chúng ta:

1. Mở tập tin index.htm trong trình soạn thảo.
2. Tìm tag <body> và thay đổi nó thành :

   ```
   <body bgcolor=#000000>
   ```

3. **Lưu trữ** và **Nạp** tập tin HTML trong Web browser của bạn

Nếu bạn thực hiện chính xác những điều trên, browser của bạn sẽ đổi màu nền thành một màu đen cố định. **Nhưng bạn có thể nhận thấy là bạn không thể nhìn thấy được văn bản của bạn**. Tại sao vậy? À, bởi vì màu mặc định cho chữ cũng là màu đen, do đó bạn biết rằng bạn có những chữ màu đen trên một nền đen! May mắn thay, chúng ta có thêm những khả năng khác trong tag body để thay đổi màu của chữ và những siêu văn bản:

```
<BODY BGCOLOR=#XXXXXX TEXT=#XXXXXX LINK=#XXXXXX VLINK=#XXXXXX>
```

trong đó các giá trị XXXXXX sẽ xác định:

- BGCOLOR = màu sắc của nền (mặc định là màu **xám** - grey)
- TEXT = màu của chữ trong thân văn bản (mặc định là **đen** - black)
- LINK = Màu sắc của một mục liên kết siêu văn bản (mặc định là **xanh** - blue)
- VLINK = Màu sắc của một mục siêu văn bản đã xem (mặc định là **tím** - purple)

Bây giờ bạn có thể thêm các giá trị màu này để thay đổi tag thành:

```
<BODY BGCOLOR=#000000 TEXT=#EEEEBB LINK=#33CCFF VLINK=#CC0000>
```

Tag này sẽ cho nền màu **đen**, chữ màu **vàng**, siêu văn bản màu **xanh dương sáng**, và siêu văn bản đã xem màu **đỏ** .

Chú ý : thứ tự của các mục trong tag <BODY> không quan trọng.
Bây giờ bạn nên sửa đổi các tag <BODY> trong tất cả các tập tin HTML của bạn (cách nhanh và dễ dàng là chép và dán ví dụ trên cho mỗi tag <BODY>).

Cấu trúc nền

Lưu ý: Trước hết có lẽ bạn hãy thử kiểm tra xem web browser của bạn có hỗ trợ cấu trúc nền không.
Màu sắc cố định tạo thêm vài biến đổi khác nhau cho trang Web - nhưng bạn có thể đi xa hơn nữa bằng cách thêm cấu trúc nền (textured background). Bạn sử dụng một tập tin hình ảnh nhỏ (dạng GIF hay JPEG) và browser sẽ "lợp" vào trang web các bản sao liên tục nhau của hình ảnh này. Vài điều bạn cần phải nhớ là:

- **Kích cở tập tin:** Việc thêm một cấu trúc nền đòi hỏi một hình ảnh nữa phải được download. Chúng tôi đề nghị tập tin hình ảnh nên có kích thước nhỏ hơn 10k.
- **Khả năng đọc:** Hãy chọn lựa! Rất nhiều tập tin sắp đặt nền làm cho người đọc cảm thấy khó nhìn thấy chữ. Cố gắng sử dụng những màu rất sáng (với chữ rất tối) hoặc những màu thật tối (với chữ rất sáng) - có nghĩa là hãy chọn độ tương phản cao giữa nền và chữ.
- **Hiệu quả:** trong các web browser đầu tiên có sử dụng màu nền, trang sẽ chưa xuất hiện cho tới khi nào màu nền đã được nạp xuống - do sự kết nối chậm, người đọc có thể phải chờ hình ảnh nền một thời gian dài trước khi nhìn thấy được một thông tin nào đó trên màn hình! Tuy nhiên, các web browser sau này download nền **cuối cùng** do đó trước tiên trang sẽ có màu xám, tiếp theo là văn bản và hình ảnh xuất hiện, và cuối cùng là nền. HÃY GHI NHỚ! Thời gian nạp trang sẽ chậm hơn nữa (nếu xét với các loại modem thế hệ cũ) khi trang của bạn được đọc từ server.

Trong phần này của bài học, chúng tôi sẽ tạo cơ hội để bạn có kinh nghiệm với ba hình nền khác nhau. Dạng HTML cho việc thêm một tập tin hình nền là :
```
<body background="bgfile.gif">
```
trong đó `bgfile.gif` là tên của tập tin hình ảnh (có thể sử dụng URL đầy đủ hay tương đối - xem bài 8a).

Dưới đây chúng ta sẽ liệt kê tên của ba tập tin nền. Bạn có thể download từng cái (nếu bạn không biết cách download hình ảnh từ một trang web, xin tham khảo bảng trợ giúp Download Hình ảnh) của chúng tôi. Bạn nên cất các tập tin hình ảnh trong directory/folder `pictures` trong vùng làm việc của bạn:

Blue Tile `[bg.gif]`
: Một mẫu hình vuông được lặp lại:
HTML: `<body background="../pictures/bg.gif">`
Tập tin ví dụ có nền Blue Tile

Volcano Text `[vtext.gif]`
: Văn bản lớn có màu xám sáng:
HTML: `<body background="../pictures/vtext.gif">`
Tập tin ví dụ với nền là Volcano Text

Legal Paper `[paper.gif]`
: Giấy có kẻ hàng
HTML: `<body background="../pictures/paper.gif">`
Tập tin ví dụ với nền là Legal Paper

Bạn cũng có thể sửa đổi màu của văn bản trong trang Web như chúng tôi đã làm trong ví dụ trên. Lấy ví dụ, nếu bạn muốn chữ màu **ĐỎ** cho nền Giấy có kẻ hàng, chúng ta có thể viết lại HTML sau:
```
<body background="../pictures/paper.gif" text=##AA0000>
```
sẽ cho chúng ta chữ đỏ trên giấy vàng.
Chú ý: Có nhiều web browser có khả năng thay đổi màu mặc định - đôi khi một người sử dụng có thể có sự ưa thích một tập các màu mà những màu này sẽ làm ảnh hưởng đến tập màu bạn đã chọn. Vì vậy, chúng tôi đề nghị khi sử dụng một tag nền bất kỳ

nào (màu cố định hay tập cấu trúc nền) thì bạn cũng nên bao gồm các màu "bình thường" - màu đen cho phần chữ văn bản, màu xanh dương cho liên kết siêu văn bản và màu tím cho các liên kết đã xem :

<BODY TEXT=#000000 VLINK=#660099 LINK=#0000EE>

Nếu bạn muốn tìm vài ví dụ về các tập tin cấu trúc nền, hãy xem Kai's Power Tips Background Archives

Kiểm tra lại công việc của bạn

So sánh trang của bạn với ví dụ mẫu. Nếu trang của bạn trông khác với mẫu hay các liên kết siêu văn bản hoạt động không đúng, hãy xem lại văn bản bạn đã nhập trong trình soạn thảo. Chúng ta sẽ giữ lại các tập tin mẫu với màu đen cố định mà chúng ta đã thêm vào ở phần đầu của bài này.

Xem lại

Xem lại những vấn đề sau:

1. Làm thế nào để thêm màu nền đồng nhất cho trang Web?
2. Tại sao mã màu lại được viết trong dạng mã khó hiểu như #EE66CC?
3. Làm thế nào để thêm màu cho văn bản trong trang Web?
4. Giữa

   ```
   <body bgcolor=#FFFFFF>
   ```

   và

   ```
   <body background="tiles.gif">
   ```

   khác nhau như thế nào?

Thực tập tự do

Thêm một màu nền cố định hay một tập tin cấu trúc nền cho (các) trang web của bạn. Hỏi vài người khác xem họ có đọc được văn bản với màu nền bạn đã chọn không.

17. Đừng làm cho chữ nhấp nháy!

Bạn muốn tạo một sự chú ý có ấn tượng trong trang Web của bạn phải không? Hãy dựa vào **nội dung** của nó hơn là tạo sự chú ý bằng cách làm cho nó trở nên nhấp nháy một cách rẻ tiền...

Mục đích

Sau bài học này bạn sẽ:

- Không bao giờ sử dụng tag nhấp nháy (blink tag).

Bài học

Lần đầu tiên khi NetScape đưa ra web browser của họ, họ thêm vào đó một tag với mục đích là để nhấn mạnh những từ hoặc những cụm từ quan trọng. Tag này làm cho chúng nhấp nháy trên trang Web.

Cấu trúc của tag nhấp nháy khủng khiếp này như sau :

<blink>Wow</blink>
là một dấu hiệu mà những người có kinh nghiệm xem trên web sẽ nói **"tránh xa trang này đi - người viết ra trang web này chỉ mới học HTML từ miếng giấy bọc kẹo cao su."**

Không nói nhiều đến các lý do rất đần độn về việc sử dụng nó, hãy tránh tất cả những gì giống như sau:

Đây *là* một vùng ***NÓNG*** TRÊN Web

Trừ phi bạn muốn xây dựng một dòng **nhếch nhác** cho các trang web... Vâng, đây là quan điểm soạn thảo của chúng tôi. Cứ tiếp tục và Hãy tranh cãi.

18. Trang trí cho Văn bản

Bạn không chỉ có thể thêm **màu sắc** cho nền mà còn có thể thêm **màu sắc**, đổi KÍCH THƯỚC, và ngay cả phông chữ cho các phần xác định của văn bản! Bạn phải trả hết bao nhiêu đây?

Nhưng HÃY CHỜ MỘT CHÚT! Bây giờ bạn có thể viết chữ ở trên và ở dưới như với những công thức hóa học và toán như sau:

$CO_2 + SO_4^{-2}$
$3x^2 - 2y^{-1/2} = 4z$

Mục đích

Sau bài học này bạn có thể:

- Thay đổi kích cở của một phần văn bản được chỉ định trong một trang Web.
- Thay đổi màu sắc của một phần văn bản được chỉ định trong một trang Web.
- Tạo chữ viết trên và dưới cho văn bản trong một trang Web.
- Chỉ định phông cho những phần văn bản trong một trang Web.

Bài học

Lưu ý: Nếu bạn chưa có tài liệu từ những bài học trước, bạn có thể download ngay bây giờ.

Với HTML do NetScape và HTML 3.0 giới thiệu, bạn có thêm một vài khả năng định dạng nữa cho văn bản của bạn. Đặc biệt nhất là, bạn có thể định những kích cở chữ cũng như màu sắc khác nhau cho văn bản. Khi sử dụng một cách đúng đắn, sự định dạng văn bản có thể làm nổi bật cách trình bày trang Web. Khi sử dụng một cách tự do, chúng có thể tạo ra các trang web trông rất hỗn độn.

Bạn cũng có thể tạo ra chữ viết trên và dưới được sử dụng trong biểu thức toán học, công thức hóa học, hoặc đánh dấu các ghi chú.

Trong bài này chúng tôi sẽ giới thiệu với bạn những tính chất này cùng với vài ví dụ mà bạn sẽ sử dụng để sửa đổi trang *Volcano Web*. Trước hết có thể bạn muốn tham khảo đến trang kiểm tra phông chữ để biết được web browser của bạn có hỗ trợ những tag được sử dụng trong bài học này không.

Kích cở phông chữ

Tag `<font size=N>...</font>` do NetScape giới thiệu có thể được sử dụng để đặt lại kích cở của phông chữ từ N là 1 (nhỏ nhất) tới 7 (lớn nhất) với cở 3 là kích cở bình thường:

Cở phông = 1
Cở phông = 2
Cở phông = 3 -- cở chữ thường
Cở phông = 4
Cở phông = 5
Cở phông = 6
Cở phông = 7
Cần nhớ rằng kích cở thật của chữ còn phụ thuộc vào phông chữ trên máy tính do người sử dụng đã chọn cho web browser của họ -- bạn đang điều chỉnh lại kích cở tương đối so với phông mặc định mà họ đã chọn.

Chú ý: Nếu một web browser không hổ trợ cho tag <font>, bạn có thể thử những tag của HTML 3.0:

<BIG>...</BIG>
<SMALL>...</SMALL>
sẽ cho bạn ít loại kích cở hơn nhưng vẫn có thể hữu ích cho những browser đó.

Dạng HTML cho tag chỉnh cở phông là:

```
<font size=N>blah blah blah</font>
```

trong đó N=1 tới 7. Tag phông có thể được sử dụng chung với những tag định kiểu chữ khác (bài 5) hay bên trong các tag tiêu đề (bài 3).

Một cách sử dụng khác của tag `font` là tạo ra một sự thay đổi kích cở **tương đối**:

```
<font size=+1>blah blah blah</font>
<font size=-2>blah blah blah</font>
```

tức là, các giá trị `+N` hoặc `-N` cho biết **độ dịch chuyển** (offset) so với kích cở phông hiện tại. Chúng thường được sử dụng cùng với một tag khác:

```
<basefont size=5>
```

để thay đổi **cở phông cơ bản** (base font size) từ giá trị mặc định của nó là 3 thành một giá trị khác. Có thể bạn phải làm điều này trong một trang Web mà có hầu hết các văn bản với cở lớn hơn hoặc nhỏ hơn cở phông chuẩn. Với cách này, nếu bạn cần điều chỉnh một phần nhỏ của trang, bạn có thể sử dụng tag chỉnh cở phông tương đối như trên.

Điểm mạnh của việc sử dụng tag chỉnh cở phông tương đối (ví dụ như `size=+2`) so với tag chỉnh cở phông tuyệt đối (ví dụ như `size=5`) là chúng ta có thể dễ dàng thay đổi cở của TẤT CẢ văn bản trong trang đó bằng một tag `<basefont>`.

Chú ý: tag `<basefont>` không có tag đóng lại - nó vẫn là cở phông cơ sở cho đến khi một tag `<basefont>` khác xuất hiện.

Trước hết chúng ta sẽ sử dụng tag chỉnh cở phông để sửa đổi tiêu đề của trang mở đầu:

1. Mở tập tin `index.html` trong trình soạn thảo của bạn.
2. Trước đây chúng ta đã sử dụng một tag tiêu đề `<h1>..</h1>` để định dạng cho tiêu đề cho trang này. Bây giờ chúng ta sẽ sử dụng vài tag chỉnh cở phông thay thế tạo một tiêu đề có cở hỗn hợp.

   Thay đổi dòng:

```
<h1>Volcano Web</h1>
```

thành:

```
<p>
<B><font size=+4>V</font><font size=+3>OLCANO
WEB</font></B>
```

Hãy nhìn cẩn thận chúng ta đã làm gì - V bây giờ đã tăng cở lên 4 đơn vị so với giá trị cơ bản, còn những ký tự khác (bây giờ dưới dạng chữ hoa) tăng lên 3 đơn vị so với giá trị cơ bản. Điều này tạo thành một cách trình bày của CHỮ HOA NHỎ (SMALL CAPS). Cũng lưu ý rằng chúng ta đã thêm tag `<b>...</b>` để làm tiêu đề nổi bật lên. Và cuối cùng, bởi vì chúng ta không sử dụng tag tiêu đề để tag này tự tạo ra sự ngắt dòng như mặc định, chúng ta phải thêm tag <p> trong tiêu đề ở trên để ép buộc nó xuống dòng mới (sau này bạn sẽ thấy không cần vì dòng HTML kế tiếp là một <BLOCKQUOTE> và nó tự thêm sự ngắt dòng của nó - xem bài 13).

3. Tiếp theo, chúng ta muốn làm lời chú thích của Pliny nổi bật hơn một chút nữa, vì thế chúng ta sẽ tăng nó lên thêm một cở nữa:
4.
```
<B><I><font size=+1>"Nature raves savagely,
threatening the lands"</font></I></B><br>
```

5. **Lưu trữ** và **Nạp** lại trong Web browser của bạn.

Trước khi tiếp tục, chúng ta sẽ sử dụng tag chỉnh cở phông để sửa đổi hai trang Web khác trong bài của chúng ta. Đó là trang **Volcanic Places in the USA** (tập tin `usa.html`) và **Research Project** (tập tin `proj.html`) đang sử dụng tag `<h2>...</h2>` cho tiêu đề chính và tag `<h3>...</h3>` cho các phần đề mục con. Đến trình soạn thảo và thay đổi mỗi sự xuất hiện của tag:

```
<h3>blah blah blah</h3>
```

thành :

```
<p>
<font size=+1><B>blah blah blah</B></font><br>
```

Lưu ý là tag <p> luôn luôn thêm một dòng mới cho văn bản theo sau và tag <br> tạo một ngắt dòng (không có dòng trắng) cho văn bản theo sau.

Bạn có nhìn thấy được sự khác nhau giữa việc sử dụng tag `<font>` và tag `<hN>` cho phần tiêu đề không? Sự định dạng khác nhau này rất tinh tế, nhưng nó cung cấp cho bạn - tác giả của trang Web - một cách nhìn mới trong việc thiết kế trang Web. Nhưng nhớ rằng nếu browser của người đọc không hỗ trợ tag chỉnh cở phông, **tất cả** chúng sẽ được nhìn thấy ở cở chữ bình thường.

Hai Phương pháp để tạo Tiêu đề

** ... **	**<p> ... **
...before the next great innovation that rocked the world of rotten milk. **History of Longhorn Cheese** Longhorn cheese was first discovered by Alister Longhorn in 1754 when he...	...before the next great innovation that rocked the world of rotten milk. **History of Longhorn Cheese** Longhorn cheese was first discovered by Alister Longhorn in 1754 when he...

Màu của font

Trong bài 16, chúng tôi đã giới thiệu những tag định màu sắc cho văn bản trong toàn bộ trang Web cùng với việc biểu diễn giá trị màu trong HTML. Bạn cũng có thể thêm những thuộc tính vào tag để định màu một đoạn văn bản xác định bằng cách sử dụng mã hệ thập lục phân hoặc tên của 16 màu:

```
<font color=red>...</font>
<font color=#993459>...</font>
<font color=lime>...</font>
<font color=#002200>...</font>
<font color=navy>...</font>
<font color=#193467>...</font>
```

Chúng ta sẽ không đi sâu vào những rắc rối của tag màu, nhưng như là một ví dụ, chúng ta sẽ đổi màu của đoạn VOLCANO WEB mà chúng ta đã xét trong trang bìa. Nếu bạn vẫn còn nhớ, chúng ta đã sử dụng màu cho văn bản của trang này là màu vàng và chúng ta sẽ thay màu khác cho phần tiêu đề:

6. Mở tập tin `index.htm` trong trình soạn thảo.
7. Điều chỉnh dòng chứa chữ VOLCANO WEB thành như sau:
8. <B>V
   OLCANO WEB</B>

9. **Lưu trữ** và **Reload** trong Web browser.

Chú ý: Các thuộc tính `size` và `color` có thể ở trong cùng một tag `<font>`. Ở đây kết quả làm cho chữ "V" có màu viôlét sáng và tăng cở hơn một đơn vị so với các chữ khác.

Chữ trên và chữ dưới (Superscript and Subscript)

HTML 3.0 cho phép bạn viết các biểu thức toán học, công thức hóa học, hoặc những biểu thức khác với superscript và/hay subscript. Những tag mới này nâng cao/hạ xuống đoạn văn bản "script" một nửa dòng và giảm bớt cở của nó một đơn vị.

Dạng HTML cho những tag này là:

Superscripts / Subscripts	
HTML	**Kết quả**
^{...}	super^{script}
_{...}	sub$_{script}$

Bây giờ chúng ta sẽ sử dụng các tag này cho trang **Introduction** của chúng ta:

10. Mở tập tin `intro.htm` trong trình soạn thảo.
11. Trước hết chúng ta sẽ sử dụng subscript để viết một vài công thức hóa học. Sau câu cuối cùng của đoạn 2 ("Compare the history of human beings..."), thêm câu sử dụng subscript sau:
12. Volcanoes were important contributors to
13. the early earth atmosphere by releasing
14. gases such as nitrogen (N₂),
15. carbon dioxide (CO₂), and
    ammonia (NH₄).

16. Bây giờ chúng ta sẽ sử dụng superscripts để biểu thị một trị lập phương. Bên dưới bảng được tạo ra bằng tag `<pre>...</pre>`, thêm câu sau:
17. Note that volcanic eruptions that occurred

```
18.       before historic times were several orders
19.       of magnitude larger (more than 1000 km<sup>3</sup>
20.       in erupted volume) than ones observed by
      humans.
```

21. Nếu bạn chú ý cột thứ ba của bảng, trước đây chúng ta sử dụng "km^3" để chỉ ra "km^3". Mặc dầu đoạn văn này ở **bên trong** tag preformat, chúng ta vẫn có thể sử dụng tag superscript. Đổi:

    ```
    Volume in km^3
    ```

    thành

    ```
    Volume in km<sup>3</sup>
    ```

Dạng phông (font face)

Trong tiêu chuẩn HTML 3.2 hiện nay có thêm vài thông số cho tag để chỉ ra dạng phông dùng cho việc hiển thị. Chức năng này có thể không làm việc trên nhiều browser, vì thế bạn có thể thử kiểm tra xem browser của bạn có hỗ trợ đổi dạng phông không.

HTML cho việc chỉ định một dạng phông là:

```
<font FACE="font1,font2">some text
```

Nếu browser của người đọc hỗ trợ đặc tính `font FACE` và họ có một danh sách các phông đã được cài đặt sẵn trên máy tính của họ, thì văn bản sẽ được hiển thị theo dạng phông được chỉ định. Ngược lại, browser sẽ sử dụng cùng một phông với phần còn lại của trang web.

Nếu bạn quyết định sử dụng một dạng phông, bạn nên sử dụng các dạng phông chuẩn hoặc là chắc chắn rằng các máy tính của người dùng đã cài đặt thêm dạng phông đó.

Bây giờ chúng ta sẽ điều chỉnh tag cho trang tiêu đề để từ "Volcano Web" xuất hiện ở một dạng font khác:

22. Mở tập tin `index.htm` trong trình soạn thảo.
23. Sửa đổi dòng chứa đoạn văn VOLCANO WEB thành:

```
24.       <B><font face="Arial,Helvetica"
25.         size=+4 color=#FF66FF>V</font>
26.       <font face="Arial,Helvetica"
      size=+3 color=#DD0055>OLCANO WEB</font></B>
```

27. **Lưu trữ** và **Reload** trong Web browser .

Chú ý: Chúng ta chỉ định cho browser chọn phông Arial với Windows, Helvetica cho Macintosh và/hay các máy tính chưa được cài đặt phông Arial.

Sử dụng tính chất này của HTML với sự thận trọng! Nghệ thuật sử dụng dạng phông khác hoàn toàn với sự lạm dụng chúng!

Kiểm tra lại công việc của bạn

So sánh trang web của bạn với ví dụ mẫu để biết dạng hiển thị của nó. Nếu trang của bạn khác với mẫu hay các liên kết siêu văn bản hoạt động không đúng, hãy xem lại văn bản bạn đã đánh trong trình soạn thảo.

Xem lại

Xem lại những chủ đề sau:

28. Làm thế nào thay đổi cở và màu sắc của một phần văn bản trong một trang Web?
29. Bạn có thể sử dụng dạng HTML nào nếu browser không hỗ trợ tag `<font>...</font>`?
30. Làm thế nào bạn tạo được subscript trong HTML?
31. Làm thế nào tạo được một khối văn bản để được trình bày dưới một dạng phông xác định?

Thông tin bổ sung

Dưới đây là hai kiểu nữa được cho phép trong HTML 3.2 và có thể đang được chấp nhận trong web browser của bạn.

Các tag Định kiểu	
HTML	**Kết quả**
<u>This is Underline...</u>	This is Underline
<strike>This is Strike-through...</strike>	~~This is Strike-through~~

Và cuối cùng, có một điểm tế nhị về màu sắc của văn bản mà một lúc nào đó bạn có thể thấy sự hữu ích. Trong bài 16, chúng ta đã học cách sử dụng tag `<body>` để tạo màu cho nền, văn bản, và các liên kết. Nếu chúng ta tô màu một khối văn bản với tag `<font>` , nó sẽ chỉ tác động lên phần thân văn bản, mà không tác động lên các liên kết siêu văn bản - có nghĩa là chúng vẫn được giữ màu xanh mặc định hoặc là màu được xác định trong tag `<body>`.

Màu Liên kết Mặc định	
HTML	**Kết quả**
 It was a long time after the sad death of <A HREF="http://www.longhorn.org/sir/"> Sir Longhorn</A> that someone was able to recreate his formula.	It was a long time after the sad death of Sir Longhorn that someone was able to recreate his formula.

Chúng ta có thể thay đổi màu của liên kết siêu văn bản bằng cách đặt các tag *bên trong* một liên kết anchor. Lưu ý rằng điều này chỉ có hiệu quả cho các liên kết chưa được xem; một khi bạn đã thấy trên màn hình trang tương ứng của liên kết, nó sẽ được tô bởi màu liên kết đã xem, tức là màu tím mặc định:

Màu Liên kết được Sửa đổi	
 It was a long time after the sad death of <A HREF="http://www.longhorn.org/sir/"> Sir Longhorn</A> that someone was able to recreate his formula.	It was a long time after the sad death of Sir Longhorn that someone was able to recreate his formula.

Thực tập tự do

Thử tìm kinh nghiệm với các tag `<font>...</font>` trong trang web riêng của bạn. Cũng nên xem thử nó hoạt động như thế nào bên trong các tag `<hN>...</hN>`. Tìm kinh nghiệm với việc sử dụng các phông khác nhau, ngay cả với các phông lung tung!

Tìm vài chỗ mà bạn nghĩ rằng bạn có thể cần superscript hay subscript. Một ví dụ có thể là các chú thích - bạn có thể đánh mã số hay ký tự cho chúng, sau đó làm cho mỗi chú thích thành một liên kết siêu văn bản đến một vùng chú thích (hay đến một trang riêng có các ghi chú chung):

- ... and after Linberger and Gordon's 1963 study[12] on the effects of temperature on cheese maturation, Gange and Walters (1964)[13] as well as Colby (1969)[14] reached the same conclusion.
- blah blah blah

<hr size=8> :

<hr size=20> :

Chiều rộng của đường

Thông số định dạng khác cho tag <hr> là điều chỉnh độ rộng của đường - không cần thiết phải trải nó ngang ra toàn bộ trang. Bạn có thể làm điều này bằng cách sử dụng dạng sau:

```
<hr width=X>
<hr width=Z%>
```

Trong đó X là số điểm của chiều rộng còn Z là tỷ lệ phần trăm của trang hiện hành. Thông thường, chúng tôi đề nghị sử dụng dạng tỷ lệ phần trăm vì nó sẽ tự điều chỉnh theo độ rộng cửa sổ browser đang được người đọc sử dụng (vài người sử dụng màn hình có độ phân giải cao trong khi các người khác chỉ có màn ảnh 13").

Sau đây là một vài ví dụ (hãy xem cách chúng ta thêm luôn tag size):

<hr width=80 size=10> :

<hr width=80% size=10> :

Chú ý: Thử co vào và/hay dãn ra độ rộng cửa sổ web browser của bạn để thấy sự khác nhau giữa việc chỉ định chiều rộng theo số điểm tuyệt đối (`width=80`) so với chỉ định theo tỷ lệ phần trăm của trang web (`width=80%`)

<hr width=40% size=10> :

<hr width=5% size=10> :

Không có bóng

Cuối cùng, có thể có trường hợp mà bạn không muốn bóng ba chiều trên tag <hr>. Nó chỉ đơn giản như sau:

<hr width=80% size=6 noshade> :

Xem lại

Xem lại những chủ đề sau :

1. Làm thế nào thay đổi độ dày của một hard rule?
2. Làm thế nào thay đổi chiều rộng của một hard rule?
3. Thuộc tính noshade bên trong tag <hr> tạo ra hiệu ứng gì?

Thực tập tự do

Tạo kinh nghiệm bằng một vài thuộc tính cho <hr> trong các trang web của bạn.

20. Xét thêm về Sự Chỉnh lề

Tại sao bạn chỉ ở bên **trái**?

... khi nào bạn qua bên **phải**?

hay là để vào giữa?

Mục đích

Sau bài học này bạn có thể:

- Tạo văn bản được canh vào giữa trang.
- Sắp xếp chung hình ảnh và văn bản.
- Tạo văn bản được chỉnh lề bên phải.

Bài học

Chú ý: Nếu bạn chưa có tài liệu từ những bài học trước, bạn có thể download bây giờ. Bạn cũng có thể xem qua trang kiểm tra để biết web browser của bạn có hỗ trợ những tag được sử dụng trong bài này hay không.

Sự chỉnh lề văn bản

Khi mà Web ngày càng phát triển, ngày càng xuất hiện thêm các yêu cầu cho việc điều khiển cách trình bày của trang. Một trong những ý muốn đó là định dạng văn bản sao cho nó được sắp xếp ngay giữa trang, hơn là chỉ được canh lề bên trái.

Đây là một trong những đặc tính khác hẳn giữa NetScape so với HTML "tiêu chuẩn". NetScape đã giới thiệu tag mở rộng ... - tag này sắp xếp mọi thứ bên trong nó ở ngay giữa trang web. Dường như nó gây ra ấn tượng hơn? Vậy thì, bạn hãy nói với một nhà làm chuẩn rằng sự chỉnh lề hàng là một **đặc tính**. Dạng HTML tổng quát cho sự sắp xếp vào giữa là một sự biến đổi của tag :

```
<p>
blah blah blah
<img src="file.gif"><br>
blah blah
</p>
```

Lưu ý rằng ở đây có tag đóng . Mỗi lần bạn muốn canh giữa một vài đoạn văn bản, bạn đánh dấu chúng bằng ...

Hiện nay, hầu hết các loại browser và chuẩn HTML 3.2 đều hỗ trợ tag ... của NetScape. Nhớ rằng nếu browser không hỗ trợ một tag, nó chỉ việc bỏ qua.

Để thấy được hiệu quả của việc canh giữa, hãy so sánh trong ví dụ dưới đây:

Không canh Giữa

Cheese In History

Cheese was been there for many major events

When Magellan sailed through the narrow Straits, he was eating cheese. When the Allied troops landed in Europe, cheese was there. When Neil Armstrong stepped onto the moon, he had just had finished a cheese snack.

return to the cheese home page

Được canh Giữa

Cheese In History

Cheese was been there for many major events

When Magellan sailed through the narrow Straits, he was eating cheese. When the Allied troops landed in Europe, cheese was there. When Neil Armstrong stepped onto the moon, he had just had finished a cheese snack.

return to the cheese home page

Giờ đây chúng ta sẽ sửa đổi trang bìa của chúng ta (`index.htm`) bằng cách sử dụng tag canh giữa trên danh sách các phần của bài học. Chúng ta cũng sẽ tạo một vài thay đổi để cải thiện sự trình bày của nó.

1. Mở tập tin `index.htm` trong trình soạn thảo.
2. Tìm phần sau:

```
<B>In this Lesson...</B>
<ul>
<i>
<li><A HREF="intro.htm">Introduction</A>
<li><A HREF="term.htm">Volcano Terminology</A>
<li><A HREF="usa.htm">Volcanic Places in the USA</A>
<li><A HREF="mars.htm">Volcanic Places on Mars</A>
<li><A HREF="proj.htm">Research Project</A>
</i>
</ul>
```

và thay thế nó bằng đoạn HTML sau:

```
<p align=center>
<font size=+1>
<i>
<A HREF="intro.htm">Introduction</A><br>
<A HREF="term.htm">Volcano Terminology</A><br>
<A HREF="usa.htm">Volcanic Places in the USA</A><br>
<A HREF="mars.htm">Volcanic Places on Mars</A><br>
<A HREF="proj.htm">Research Project</A><br>
</i>
```

```
</font>
</p>
```

12. **Lưu trữ** và **Nạp** lại trong Web browser của bạn.

Chú ý: Hãy nhìn một cách cẩn thận những thay đổi mà chúng ta đã thực hiện. Toàn bộ phần này được bao quanh trong tag `<p align=center>..</p>` của sự chỉnh lề. Danh sách không có thứ tự mà chúng ta đã xây dựng trong bài 6 được canh vào giữa một cách lổ chổ - các dấu bullet sẽ bị lộn xộn. Vì thế chúng ta sẽ bỏ đi cấu trúc `<ul> <li> ... <li> ... </ul>`. Các tag `<br>` ở cuối mỗi dòng sẽ tạo một ngắt dòng. Và cuối cùng, chúng ta thêm một tag `<font>...</font>` để tăng cỡ văn bản.

Nếu muốn bạn có thể sử dụng tag `<center>...</center>` thay cho tag `<p align=center>...</p>`

Bạn có thể muốn so sánh tập tin HTML của bạn đối với ví dụ mẫu để biết được nó nên xuất hiện như thế nào.

Mọi tag tiêu đề (header) `<hN>...</hN>` đều có thể canh giữa bằng cách thêm một thuộc tính như sau:

```
<hN align=center>blah blah blah</hN>
```

Bây giờ chúng ta sẽ sử dụng thuộc tính này để canh giữa tựa đề của mỗi trang Web:

1. Mở tất cả file HTML của bạn trong trình soạn thảo.
2. Với mỗi một trang, sửa chữa lại văn bản xuất hiện ở gần đỉnh trong tag `<h1>` hay tag `<h2>` theo như ví dụ dưới đây cho tập tin `intro.htm`, thay đổi từ:

   ```
   <h2>Introduction</h2>
   ```

   thành

   ```
   <h2 align=center>Introduction</h2>
   ```

3. **Lưu trữ** và **Reload** trong Web browser của bạn.

Chú ý: Cũng có một thuộc tính chỉnh lề cho tag `<hr>` phù hợp với các thông số khác mà chúng ta đã xem trong bài 19 "Easy Hard Rules". Khi bạn chỉ định một chiều rộng ngắn hơn cho một hard rule, bạn cũng có thể chỉ định canh lề cho nó về bên phải hay bên trái (theo mặc định các hard rule được canh vào giữa trang):

```
<hr size=8 width=60% align=right>
```

sẽ tạo ra:

trong khi sử dụng canh lề trái

```
<hr size=8 width=60% align=left>
```

sẽ cho kết quả là:

Sự sắp xếp giữa Hình ảnh và Văn bản

Trong bài 7a chúng ta đã xét cách đặt hình ảnh vào trong trang Web của chúng ta và đã thấy rằng có thể có một dòng văn bản được sắp xếp ở trên hoặc cuối một hình ảnh. Tuy nhiên, cho đến tận bây giờ, chúng ta chưa thể có một khối văn bản và một hình ảnh inline nằm cạnh nhau.

Với việc thêm thuộc tính `align` vào tag `<img>` giờ đây bạn có thể có chỉ định hình ảnh tự sắp về bên trái hay bên phải của trang. Nhưng hơn thế nữa chúng ta có thể có các văn bản HTML khác "lấp đầy" những khoảng trống quanh hình. HTML để thực hiện điều này là:

```
<img src="filename.gif" align=right>
<img src="filename.gif" align=left>
```

Hãy so sánh hai ví dụ sau đây:

không có thuộc tính align
`align=bottom`

Cheese In History

Cheese was been there for many major events
When Magellan sailed through the narrow Straits, he was eating cheese. When the Allied troops landed in Europe, cheese was there. When Neil Armstrong stepped onto the moon, he had just had finished a cheese snack.

có văn bản đi theo
`align=left`

Cheese In History

Cheese was been there for many major events
When Magellan sailed through the narrow Straits, he was eating cheese. When the Allied troops landed in Europe, cheese was there. When Neil Armstrong stepped onto the moon, he had just had finished a cheese snack.

Hãy co dãn cửa sổ browser của bạn để thấy các hiệu ứng trong trang của bạn với việc sắp xếp hình ảnh/văn bản.

Thêm một điều lưu ý nữa. Có khi văn bản hoặc những mục khác được sắp cùng với hình ảnh khá ngắn và bạn muốn đẩy văn bản kế tiếp xuống bên dưới hình ảnh. Có một thuộc tính cho tag `<br>` để xoá đi sự sắp xếp, ví dụ như:

```
<br clear=left>
<br clear=right>
<br clear=all>
```

sẽ "xóa" bất kỳ sự sắp xếp nào đã được cài đặt trong tag `<img>` trước đó. Chúng tôi đề nghị bạn luôn luôn sử dụng những tag này bởi vì sự sắp xếp sẽ phụ thuộc rất nhiều vào phông chữ trên browser của người đọc và độ rộng cửa sổ browser của họ.

Bây giờ chúng ta sẽ trở lại trang `index.htm` để định dạng lại phần mở đầu. Hình trông có vẻ đẹp, nhưng nó lại chiếm quá nhiều hàng trước khi các văn bản được trình bày. Chúng ta sẽ sử dụng sự sắp xếp hình ảnh/văn bản để đặt văn bản kế cận bên hình ảnh.

1. Mở tập tin `index.htm` trong trình soạn thảo.
2. Tìm phần sau:

```
<img alt="A Lesson on:" src="../pictures/lava.gif">
<p>
<B><font size=+4 color=#FF66FF>V</font>
```

```
6.      <font size=+3 color=#DD0055>OLCANO WEB</font></B>
7.      <BLOCKQUOTE>
8.      <B><I><font size=+1>"Nature raves savagely,
9.      threatening the lands"</font></I></B><br>
10.     -- <A HREF="http://magic.geol.ucsb.edu/~fisher/pliny.htm">
11.     Pliny the Elder</A>, who died of asphyxiation after
12.     observing the destruction of Pompeii by the
13.     79 A.D. eruption of Mount Vesuvius.
14.     </BLOCKQUOTE>
15.
16.     In this lesson you will use the Internet
17.     to research information on volcanoes and then
18.     write a report on your
        results.
```

và thay nó bằng:

```
<img alt="A Lesson on:"
  src="../pictures/lava.gif" align=left>
<B><I><font size=+1>
"Nature raves savagely, threatening the lands"
</font></I></B><br>
-- <A HREF="http://magic.geol.ucsb.edu/~fisher/pliny.htm">
Pliny the Elder</A>, who died of asphyxiation after
observing the destruction of Pompeii by the
79 A.D. eruption of Mount Vesuvius.
<p>
<B><font size=+4 color=#FF66FF>V</font>
<font size=+3 color=#DD0055>OLCANO WEB</font></B>
<p>
In this lesson you will use the Internet
to research information on volcanoes and then write
a report on your results.
<br clear=left>
```

19. **Lưu trữ** và **Nạp** lại trong Web browser của bạn.

Chú ý: Mọi thứ ở giữa tag `<img... align=left>` và tag `<br clear=left>` sẽ được sắp chung với hình ảnh - hình ở bên trái và phần HTML trong các vị trí trống còn lại.

Chúng ta cũng để câu trích dẫn lên đầu để gây sự chú ý. Với dạng trình bày này, tag `<blockquote>` không còn có hiệu quả nữa, do vậy nó được xóa bỏ.

Nếu muốn, bạn có thể so sánh tập tin HTML của bạn với ví du mẫu để biết tại đây nên trình bày như thế nào.

Chỉnh lề (Justification)/Sắp xếp (Alignment) văn bản

Chúng ta sẽ thêm một tag sắp xếp nữa là tag "divisions" `<div>...</div>`, được giới thiệu trong HTML 3.0. Tất cả văn bản bên trong tag này được chỉnh lề dựa vào thuộc tính `align`:

```
<div align=left>...</div>
<div align=right>...</div>
<div align=center>...</div>
```

Lưu ý rằng thuộc tính center có hiệu quả giống như tag `<center>...</center>` của NetScape.

Bây giờ chúng ta sẽ sử dụng tag này để làm cho đoạn văn bản trong phần ghi chú mở đầu của trang bìa được chỉnh lề ở bên phải của trang:

1. Mở tập tin `index.htm` trong trình soạn thảo.
2. Thêm tag `<div>` and `</div>` như dưới đây:

```
<div align=right>
<B><I><font size=+1>
"Nature raves savagely, threatening the lands"
</font></I></B><br>
-- <A HREF="http://magic.geol.ucsb.edu/~fisher/pliny.htm">
Pliny the Elder</A>, who died of asphyxiation after
observing the destruction of Pompeii by the
79 A.D. eruption of Mount Vesuvius.
<p>
<B><font size=+4 color=#FF66FF>V</font>
<font size=+3 color=#DD0055>OLCANO WEB</font></B>
</div>
```

14. **Lưu trữ** và **Reload** trong Web browser của bạn.

Kiểm tra lại công việc của bạn

So sánh trang Web của bạn với ví dụ mẫu để xem nó nên xuất hiện như thế nào. Nếu trang của bạn khác với mẫu hay các liên kết siêu văn bản không hoạt động đúng, kiểm tra lại văn bản bạn đánh vào trong trình soạn thảo.

Xem lại

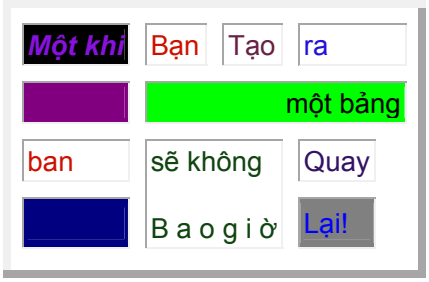
Xem lại những chủ đề sau :

1. Làm thế nào để cho một phần nào đó trong trang web của bạn được canh vào giữa?
2. Cách nào đúng nhất để làm cho một hình ảnh được sắp xếp bên lề phải của trang Web?
3. Làm cách nào bạn tạo ra một văn bản được chỉnh lề phải?

21. Bảng (Table)

Hãy tạo một bảng...

...và hoàn tất
cuộc cách mạng hóa trên
những trang Web thông thường



Mục đích

Sau bài học này bạn sẽ có thể:

- Thiết kế một bảng với những hàng và cột văn bản trình bày theo dạng lưới trên trang Web.
- Viết HTML tạo sự trộn lẫn phức tạp giữa hình ảnh và văn bản.
- Viết HTML cho một bảng ẩn để trình bày các cột dữ kiện.
- Tạo một bảng với những ô có màu khác nhau.

Bài học

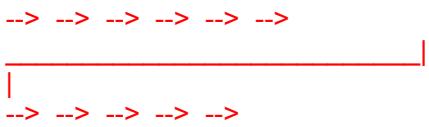
Chú ý: Nếu bạn chưa có tài liệu từ những bài học trước, bạn có thể download ngay bây giờ. Bạn cũng có thể xem trang kiểm tra để biết web browser đang sử dụng có hỗ trợ cho những tag trong bài này không.

Bảng được giới thiệu trong HTML 3.0 và được NetScape phát triển thêm để tạo ra một hướng thiết kế khác cho trang Web. Chúng cung cấp một cấu trúc để tổ chức những phần tử HTML trong trang thành dạng "lưới" (grid). Một trong những ứng dụng rõ ràng nhất của bảng là khi bạn phải định dạng... một bảng theo từng cột! Nhưng bảng cũng mở ra một chân trời mới cho nhiều khả năng trình bày trang khác nhau.

Dạng HTML cho bảng trông có thể rất phức tạp - nhưng chúng ta sẽ bắt đầu một cách đơn giản với việc xây dựng một vài bảng đơn giản trong bài *Volcano Web*.

Đối với những người mới bắt đầu, hãy nhớ kỹ khái niệm sau:

Các bảng được xây dựng bắt đầu từ phần tử cao nhất bên trái, tiếp theo xây dựng các phần tử của hàng đầu tiên, xong chuyển đến hàng thứ hai, xây dựng các phần tử của hàng thứ hai....



Chúng ta gọi mỗi ô lưới trong một bảng là một *phần tử* (cell)

Những tag cơ bản của Bảng

HTML cho cấu trúc cơ bản của một bảng được trình bày như dưới đây:

HTML

```
<table border=1>
 <tr>
 <td>Row 1 col 1</td>
 <td>Row 1 col 2</td>
 <td>Row 1 col 3</td>
 </tr>

 <tr>
 <td>Row 2 col 1</td>
 <td>Row 2 col 2</td>
 <td>Row 2 col 3</td>
 </tr>

 <tr>
 <td>Row 3 col 1</td>
 <td>Row 3 col 2</td>
 <td>Row 3 col 3</td>
 </tr>

</table>
```

Kết quả

Row 1 col 1	Row 1 col 2	Row 1 col 3
Row 2 col 1	Row 2 col 2	Row 2 col 3
Row 3 col 1	Row 3 col 2	Row 3 col 3

Thuộc tính `border=1` trong tag `<table>` chỉ thị cho browser vẽ một đường viền quanh bảng với độ dày là một điểm. Lưu ý rằng mỗi một hàng được xác định bởi tag **T**able **R**ow là `<tr>...</tr>`, rồi sau đó mỗi phần tử trong hàng được định nghĩa bởi tag **T**able **D**ata là `<td>...</td>`. Trong mỗi tag `<td>...</td>` có thể có bất kỳ loại tag nào của HTML mà chúng ta đã sử dụng trong tài liệu này - tiêu đề, định kiểu, liên kết siêu văn bản, hình ảnh inline, v.v.... Trong tag này bạn còn có thể sử dụng một vài loại thuộc tính để điều khiển sự chỉnh lề của các thông tin trong phần tử đó:

Chỉnh lề theo chiều ngang

- **`<td align=left>`** Sắp xếp tất cả các thành phần về bên trái của cell (đây là thuộc tính mặc định)
- **`<td align=right>`** Sắp xếp tất cả các thành phần về bên phải của cell
- **`<td align=center>`** Sắp xếp tất cả các thành phần vào giữa cell

Chỉnh lề theo chiều dọc

- **`<td valign=top>`** Sắp xếp các thành phần từ đỉnh cell xuống dưới
- **`<td valign=bottom>`** Sắp xếp các thành phần từ dưới cell lên trên
- **`<td valign=middle>`** Sắp xếp tất cả phần tử ở giữa của cell (đây là thuộc tính mặc định)

Bạn có thể kết hợp các thuộc tính này lại với nhau:

```
<td align=left valign=bottom>
```

Dạng HTML này sẽ hiển thị một phần tử với các thành phần dọc theo bên trái và từ dưới lên.

Các Hàng và Cột

Bảng ở ví dụ trên trông rất đơn giản và vuông vức - ba hàng và ba cột. Bây giờ hãy xem chúng ta làm được gì khi sử dụng các tag thuộc tính `colspan` và `rowspan` trong tag `<td>...</td>`.

HTML	Kết quả

```
<table border>
  <tr>
  <td>Row 1 col 1</td>
  <td align=center colspan=2>
    Row 1 col 2-3</td>
  </tr>

  <tr>
  <td>Row 2 col 1</td>
  <td>Row 2 col 2</td>
  <td>Row 2 col 3</td>
  </tr>

  <tr>
  <td>Row 3 col 1</td>
  <td>Row 3 col 2</td>
  <td>Row 3 col 3</td>
  </tr>
</table>
```

** Lưu ý đến thuộc tính cho phần tử thứ hai của hàng đầu tiên - nó **mở rộng** ra thành 2 cột. Chúng ta cũng chỉnh văn bản vào chính giữa của phần tử này này.

Row 1 col 1	Row 1 col 2-3	
Row 2 col 1	Row 2 col 2	Row 2 col 3
Row 3 col 1	Row 3 col 2	Row 3 col 3

Được rồi, chúng ta đã biết một phần tử mở rộng trong hai cột - giờ đây, chúng ta sẽ tạo một phần tử mở rộng trong hai hàng. Hãy nhớ rằng HTML xây dựng bảng từ bên trái, băng qua hàng, rồi xuống dưới, rồi băng qua...

HTML	Kết quả

```
<table border=1>
  <tr>
  <td>Row 1 col
1</td>
  <td align=center
colspan=2>
   Row 1 col 2-
3</td>
  </tr>

  <tr>
  <td>Row 2 col
1</td>
  <td valign=top
rowspan=2>
  Row 2 col 2</td>
  <td>Row 2 col
3</td>
  </tr>

  <tr>
  <td>Row 3 col
1</td>
  <td>Row 3 col
3</td>
  </tr>
</table>
```

Row 1 col 1	Row 1 col 2-3	
Row 2 col 1	Row 2 col 2	Row 2 col 3
Row 3 col 1		Row 3 col 3

Vẫn còn một vài điều phải xét đến, nhưng chúng ta tạm ngưng các ví dụ minh họa ở đây để thực hiện công việc trên trang Web của chúng ta.

Một bảng dữ liệu

Trong trang Introduction của chúng ta có một bảng ("Volumes of Some Well Known Eruptions") mà chúng ta đã tạo ra lần đầu tiên trong bài 9 bằng cách sử dụng tag preformat `<pre>...</pre>`. Giờ đây chúng ta sẽ cải thiện nó bằng cách sử dụng tag `table`.

1. Mở tập tin `intro.htm` trong trình soạn thảo.
2. Xoá bỏ mọi thứ bên trong tag `<pre>...</pre>` và cả chính nó.
3. Thay vào chính chỗ đó bằng:

```
<table border>
<tr>
<th>Eruption</th>
<th>Date</th>
<th>Volume in km<sup>3</sup></th>
</tr>

<tr>
<td>Paricutin, Mexico</td>
<td align=center>1943</td>
<td align=center>1.3</td>
</tr>

<tr>
<td>Mt. Vesuvius, Italy</td>
<td align=center>79 A.D.</td>
<td align=center>3</td>
```

```
22.  </tr>
23.
24.  <tr>
25.  <td>Mount St. Helens,<br>Washington</td>
26.  <td align=center>1980</td>
27.  <td align=center>4</td>
28.  </tr>
29.
30.  <tr>
31.  <td>Krakatoa, Indonesia</td>
32.  <td align=center>1883</td>
33.  <td align=center>18</td>
34.  </tr>
35.
36.  <tr>
37.  <td>Long Valley, California</td>
38.  <td align=center>pre-historic</td>
39.  <td align=center>>450 & <700</td>
40.  </tr>
41.
42.  <tr>
43.  <td>Yellowstone, Wyoming</td>
44.  <td align=center>pre-historic</td>
45.  <td align=center>400</td>
46.  </tr>
47.  </table>
```

Chú ý: Hãy nhìn dòng HTML đầu tiên. Tag Table Header là `<th>...</th>` có kết quả giống y như tag `<td>...</td>` ngoại trừ là mọi văn bản trong nó đều được in đậm và tất cả các thành phần của nó đều tự động được canh giữa.

48. **Lưu trữ** và **Nạp** lại trong Web browser. Bạn có thể so sánh với bản mẫu để xem nó nên xuất hiện như thế nào tại lúc này.

Chú ý: Bảng có thể xem không được rõ khi sử dụng nền màu đen.

49. Bây giờ hãy thêm một vài thứ vào trong bảng của chúng ta.
50. Một vài browser cho phép bạn chỉ định một vài thông số cho các đường tạo ra bảng. Chúng là các thuộc tính của tag `<table>`:

```
<table border=X cellpadding=Y cellspacing=Z>
```

trong đó X là chiều rộng (tính bằng điểm) đường viền bên ngoài của bảng. Thuộc tính `cellpadding` cho biết có bao nhiêu "khoảng trắng" giữa khối bên trong phần tử và vách ngăn - trị Y càng lớn càng làm cho các phần tử (cell) lớn hơn do "độn thêm" (padding). Thuộc tính `cellspacing` cho biết độ rộng (tính bằng điểm) của những đường bên trong bảng để phân chia các phần tử.

Hãy sửa tag `<table>` của bạn thành:

```
<table border=3 cellpadding=4 cellspacing=8>
```

51. Tag `<caption>` đặt một chuỗi văn bản (được chỉnh vào giữa theo chiều rộng) như là tiêu đề cho bảng. Hãy sửa đổi những dòng trong bảng của bạn thành:

```
52.    <table border=3 cellpadding=4 cellspacing=8>
53.    <caption><b><font size=+1>
54.    Volumes of Some Well-Known
Volcanic Eruptions</font></b></caption>
```

Bạn có thể lồng vào trong tag <caption> các HTML khác; nhưng phải nhớ là tag này phải **nằm trong** tag <table>...</table>.

55. Và để tăng sự thích thú, hãy tô màu đoạn văn trong tag <th>...</th> thành màu cam (để biết thêm về tô màu cho văn bản, xem bài 19). Điều chỉnh phần HTML cho hàng đầu tiên thành:

```
56.    <tr>
57.   <th><font color=#EE8844>Eruption</font></th>
58.   <th><font color=#EE8844>Date</font></th>
59.   <th><font color=#EE8844>Volume in km<sup>3</sup></font></th>
</tr>
```

60. Và hãy di chuyển bảng này vào giữa trang. Nếu browser của bạn có hỗ trợ tag ... , thì chỉ cần bao quanh bảng bằng tag này. Để biết thêm về chỉnh lề cho văn bản, xem lesson 20.
61. **Lưu trữ** và **Nạp lại** trong web browser của bạn. Bạn có thể so sánh với ví dụ mẫu để biết bảng này nên như thế nào. Vẫn tốt chứ, phải không?
62. Cuối cùng, chúng ta sẽ thêm một cột về phía bên trái - chúng ta muốn ghép nhóm các núi lửa đã phun lại trong lịch sử (bốn hàng đầu tiên) và các núi lửa thời tiền sử (hai dòng cuối). Bây giờ chúng ta sẽ thêm một phần tử rỗng bằng cách thêm <th></th> vào hàng đầu tiên - lý do của việc làm này sẽ được làm rõ hơn trong những bước sắp tới khi chúng ta xây dựng cột mới này .

```
63.   <tr>
64.   <th></th>
65.   <th><font color=#EE8844>Eruption</font></th>
66.   <th><font color=#EE8844>Date</font></th>
67.   <th><font color=#EE8844>Volume in km<sup>3</sup></font></th>
</tr>
```

68. Chúng ta đến dòng đầu tiên và thêm một phần tử ở đầu tiên mà phần tử này sẽ mở rộng ra 4 hàng:

```
69.
70.   <tr>
71.   <td rowspan=4>
72.   <font color=#EE8844>
73.   <i>eruptions<br>
74.   observed<br>
75.   by humans</i>
76.   </font></td>
77.   <td>Paricutin, Mexico</td>
78.   <td align=center>1943</td>
79.   <td align=center>1.3</td>
80.   </tr>
```

**Chú ý: Chúng ta đã thêm một vài tag
 để cột đầu tiên này không trở nên quá rộng. Bạn có thể tự đánh giá khi bỏ qua những tag này.**

81. Và trong hàng thứ năm, chúng ta thêm một phần tử mà nó được mở rộng trong hai hàng:

```
82.
83.   <tr>
84.   <td rowspan=2>
85.   <font color=#EE8844>
86.   <i>inferred<br>
87.   by study<br>
88.   of deposits</i>
89.   </font></td>
90.   <td>Long Valley, California</td>
```

```
91.   <td align=center>pre-historic</td>
92.   <td align=center>>450 & <700</td>
      </tr>
```

93. **Lưu trữ** và **Nạp lại** một lần nữa trong browser của bạn. Bạn có thể so sánh với bài mẫu để biết kết quả lúc này nên như thế nào. Đây là tất cả những gì chúng ta sẽ thêm vào bảng này.

Bảng không nhìn thấy hay Bảng Ma

Một bảng cùng với những đường viền thì rất tiện lợi nhưng đôi khi chúng ta muốn tạo nên một sự trình bày ở dạng lưới và không muốn có các đường viền. Chúng ta gọi đó là các bảng "ma" (phantom) vì người đọc không nhìn thấy rõ ràng đó là một bảng.

Một bảng ma được xây dựng theo cách giống như bảng mà chúng ta đã thực hiện ở trên ngoại trừ tag **<table>** có dạng như sau:

```
 <table>
```

hoặc

```
 <table border=0>
```

Trên một vài browser, khi không có thuộc tính border sẽ làm cho nó hiển thị một bảng không có bất kỳ đường viền nào. Trên vài browser khác, bạn cần phải cài đặt border là 0 điểm để tạo ra hiệu quả này.

Nếu bạn xem lại phần gần đầu của trang này, thật sự nó là một bảng ma mà trong đó có một phần tử của nó lại là một bảng có đường viền! Tuy vậy, chúng ta hãy cứ tiếp tục. Để làm ví dụ, chúng ta sẽ định dạng lại tập tin **usa.htm** (Volcanoes in the USA) thành dạng có hai cột. Thay vì tạo các đoạn văn được sắp đặt thẳng đứng dựa theo độ rộng của trang, chúng ta sẽ làm chúng giống như bản phác họa sau:

Lưu ý rằng cột bên phải cũng có một hình nằm kề bên, và đó chính là một siêu liên kết đến một ảnh lớn trong một cửa sổ riêng (xem bài 8e).

1. Mở tập tin `usa.htm` trong trình soạn thảo.
2. Tìm phần sau đây:

```
3.
4.    <font size=+1><B>Mount St Helens</B></font><br>
5.    On May 18, 1980, after a long period of rest,
6.    this quiet mountain in Washington provided
7.    <a href="msh.htm">detailed observations</a> on
8.    the mechanics of highly explosive eruptions.
9.
10.   <p>
11.   <font size=+1><B>Long Valley</B></font><br>
12.   This field seismometer measures earthquakes
13.   associated with subsurface volcanic forces and
```

```
may help to predict future events. It sits on a
plateau known as the "Volcanic Tableland" formed
by a major eruption  600,000 years ago.<p>
<a href="../pictures/seismo.jpg">
<img src="../pictures/stamp.gif">
-- [full size image] --</a>
```

và thay thế nó bằng đoạn HTML sau:

```
<table cellpadding=6>
<tr>
<td><font size=+1><B>Mount St Helens</B></font></td>
<td colspan=2><font size=+1><B>Long Valley</B></font></td>
</tr>

<tr>
<td valign=top>On May 18, 1980, after a
long period of rest, this quiet mountain
in Washington provided <a href="msh.htm">
detailed observations</a> on
the mechanics of highly explosive eruptions.
</td>

<td valign=top>
This field seismometer measures earthquakes
associated with subsurface volcanic forces
and may help to predict future events. It
sits on a plateau known as the
"Volcanic Tableland" formed by a major
eruption  600,000 years ago.
</td>
<td valign=top><a href="../pictures/seismo.jpg">
<img src="../pictures/stamp.gif"></a>
</td>
</tr>

<tr>
<td colspan=3 align=right>
<a href="../pictures/seismo.jpg">
-- [full size image] --</td>
</tr>
</table>
```

Chú ý: Hãy xem cẩn thận dạng HTML ở đây. Thật sự chúng ta đã sử dụng một bảng có 3 cột - đoạn văn đầu tiên (Mount St Helens) ở trong cột bên trái. Còn cột bên phải bao gồm một cột văn bản và một cột khác có một ảnh nhỏ. Hàng dưới cùng được chỉnh lề phải và mở rộng trong 3 cột được dùng để nối liên kết siêu văn bản đến một hình ảnh giống như hình thu nhỏ này nhưng lớn hơn.

19. **Lưu trữ** và **Nạp lại** một lần nữa trong web browser.

Chia đoạn một danh sách dài

Một cách sử dụng khác nữa của bảng là đổi dạng cho một danh sách dài (tạo bởi tag `list` , xem bài 6). Danh sách được kéo dài theo trang, và rất là có giá trị nếu chúng ta có thể sử dụng thêm được phân bên phải của trang.

Hiệu quả là sự chuyển dạng của danh sách:

Danh sách Dài		**Cột 1**	**Cột 2**
    xxxxxx xxxx xxxx xxx x xxxx xxx xxxxx xx x xxxxx xxx xx xxxx x xxx x xxx 	**thành**	 xxxxxx xxxx xxxx xxx x xxxx xxx xxxxx 	 xxx xx xxxx x xxx x xxx

Bây giờ chúng ta sẽ chia danh sách các tài nguyên trong trang Resource Projects của chúng ta (tập tin `proj.htm`).

1. Mở tập tin `proj.htm` trong trình soạn thảo.
2. Tìm danh sách `<ul>...</ul>` bên dưới tiêu đề **References** và nhập như sau :
3.
4. <table>
5. <tr>
6. <td valign=top>
7.
8. <A HREF="http://www.avo.alaska.edu/">
9. Alaska Volcano Observatory</A>
10. <A HREF="http://vulcan.wr.usgs.gov/home.html">
11. Cascades Volcano Observatory</A>
12. <A HREF=
13. "http://www.dartmouth.edu/pages/rox/volcanoes/elecvolc.html">
14. The Electronic Volcano</A>
15. <A HREF="http://www.geo.mtu.edu/volcanoes/">
16. Michigan Tech Volcanoes Page
17. <A HREF="http://www.geo.mtu.edu/eos/">
18. NASA Earth Observing System (EOS) IDS Volcanology Team</A>
19.
20. </td>
21. <td valign=top>
22.
23. <A HREF=
24. "http://spso2.gsfc.nasa.gov/NASA_FACTS/volcanoes/volcano.html">
25. NASA Facts: Volcanoes and Global Climate Change</A>
26. <A HREF=
27. "http://www.ngdc.noaa.gov/seg/hazard/hazards.html">
28. NGDC Natural Hazards Data
29. <a href=
30. "gopher://hoshi.cic.sfu.ca:5555/11/epix/topics/volcano">
31. Volcano Listserv
32.
33. Volcano Watch Newsletter
34. <a href=
35. "http://seawifs.gsfc.nasa.gov/JASON/HTML/EXPEDITIONS_JASON_6_home.html">
36. JASON Project VI: Island Earth
37. <A HREF="http://volcano.und.nodak.edu/">
38. VolcanoWorld</A>
39.
40. </td>
41. </tr>

```
</table>
```

Chú ý: chúng ta chỉ lấy danh sách đi `<ul>...</ul>` và chia nó thành hai danh sách. Mỗi một danh sách là một phần tử của một bảng ma có một hàng và hai cột.

42. **Lưu trữ** và **Reload** trong Web browser. Bạn có thể so sánh với mẫu để biết bảng nên có dạng như thế nào tại đây.

Màu sắc của bảng

Một vài browser hiện nay hỗ trợ để tô màu bảng, hàng và những cell của phantom table. Thật ra, chúng ta đã sử dụng nó trong suốt tài liệu này - trong trang Giới thiệu , trong Mục Lục, và trong các bài khi trình bày các ví dụ HTML.

Bạn có thể xem trang kiểm tra cuối cùng để biết browser của bạn có hỗ trợ việc tô màu cho bảng không.

Chúng ta sẽ sử dụng mã màu hệ thập lục phân đã biết trong bài 16 để tô màu nền của những trang Web và trong bài 18 để tô màu chữ.

Dưới đây là một ví dụ đơn giản về việc chèn màu `bgcolor=#FF0000` trong các tag `<table>`:

Phần của Bảng	Dạng HTML
bảng Tô màu nền cho toàn bảng	<table bgcolor=#880000>
hàng tô màu nền cho một hàng đơn	<tr bgcolor=#880000>
phần tử tô màu nền cho một phần tử	<td bgcolor=#880000>

Bây giờ chúng ta sẽ thêm một vài màu cho bảng dữ liệu bạn đã tạo trong trang "Introduction". Trong ví dụ của chúng ta, đơn giản chúng ta chỉ thêm dạng HTML để làm cho các phần tử trong các hàng và cột tiêu đề có màu xám sáng hơn một chút so với nền đen.

1. Mở tập tin `intro.htm` trong trình soạn thảo.
2. Trước hết tìm tất cả tag `<th>` để giữ các tiêu đề của cột:
3.
4. `<th><font color=#EE8844>Eruption</font></th>`
5. `<th><font color=#EE8844>Date</font></th>`
6. `<th><font color=#EE8844>Volume in km<sup>3</sup></font></th>`

và thêm thuộc tính để tô màu các phần tử này thành màu xám (#222222):

```
<th bgcolor=#222222><font color=#EE8844>Eruption</font></th>
<th bgcolor=#222222><font color=#EE8844>Date</font></th>
<th bgcolor=#222222><font color=#EE8844>Volume in
                    km<sup>3</sup></font></th>
```

7. Bây giờ, tìm hai tag đặt tên cho các hàng và tô với cùng màu sắc vào như trên:
8.
9. `<td  bgcolor=#222222 rowspan=4>`
10. `<font color=#EE8844>`
11. `<i>eruptions<br>`

```
observed<br>
by humans</i>
</font></td>

  :
  :
  :
<td  bgcolor=#222222 rowspan=2>
<font color=#EE8844>
<i>inferred<br>
by study<br>
of deposits</i>
</font></td>
```

26. Lưu trữ và Nạp lại trong Web browser của bạn. Bạn có thể so sánh với mẫu để biết bảng nên xuất hiện như thế nào tại đây.

Kiểm tra lại công việc của bạn

So sánh trang của bạn với mẫu để biết chúng nên như thế nào. Nếu trang của bạn khác với mẫu hoặc các liên kết siêu văn bản hoạt động không đúng, kiểm tra lại văn bản bạn đã đánh trong trình soạn thảo. Các bảng trở nên phức tạp rất nhanh, vì vậy nên kiểm tra lại một lần nữa xem các hàng và cột có hoàn toàn đúng không.

Xem lại

Xem lại những chủ đề sau cho bài này:

1. Các phần tử của tag bảng được một browser diễn dịch theo thứ tự nào?
2. Sự khác nhau giữa tag `<td>...</td>` và tag `<th>...</th>` là gì?
3. Thuộc tính `colspan=X` để làm gì? Cách hoạt động của nó?
4. Làm thế nào tạo một bảng có các cạnh không trông thấy được?
5. Làm thế nào tô màu một dòng của bảng?

Thực tập tự do

Xem lại trang web của bạn và tìm các vị trí mà bạn thấy bảng sẽ làm cho cách trình bày tốt hơn. Hay thêm một sơ đồ vào trang web của bạn và sử dụng bảng để định dạng cho nó. Thử tạo bảng với các phần tử được tô màu, hoặc giống như chúng tôi đã làm trong tài liệu này, sử dụng màu trong các bảng không nhìn thấy được.

22. Xét thêm về Hình ảnh và Danh sách

Hãy làm bốc hơi thoát ra khỏi những hộp dễ gây tức giận
bao quanh các button hình ảnh và

- biểu diễn
  - những
- bullet

A. cộng thêm với
b. việc thay đổi
III. **kiểu số**
iv. và

109. **giá trị** các
110. phần tử trong danh sách

Mục đích

Sau bài học này bạn có thể:

- Tạo một hình ảnh siêu liên kết không có đường viền xung quanh.
- Viết HTML cho một danh sách không có thứ tự sử dụng các dấu bullet khác nhau.
- Tạo một danh sách có thứ tự được đánh số bằng chữ hoa hay chữ thường, hoặc là những số La mã lớn hoặc nhỏ.
- Sửa đổi một danh sách có thứ tự để bắt đầu đánh số từ một trị dương bất kỳ.

Bài học

Lưu ý: Nếu bạn không có tài liệu từ những bài học trước, bạn có thể download ngay bây giờ. Bạn cũng có thể xem qua trang kiểm tra để biết browser của bạn có hỗ trợ các tag được sử dụng trong bài này không.

Không có Đường viền trên Hình ảnh Siêu liên kết

Trong bài 8e chúng ta đã biết làm thế nào để một hình ảnh hoạt động như một siêu liên kết đến những trang Web khác hay đến một bản sao lớn hơn của hình ảnh. Chúng ta nhận thấy là browser đặt một hộp đường viền quanh hình ảnh, để cho biết rằng đó là một "hyper" như các mục siêu văn bản khác.

hoạt động như bất kỳ một liên kết siêu văn bản nào khác.

Tuy nhiên, cái hộp đôi lúc tạo ra sự cản trở, nhất là trong trường hợp chúng ta có một hình có sẵn đường viền không vuông vức. Người sử dụng về nguyên tắc vẫn xác định được một hình ảnh là "hyper" nếu dạng dấu nháy thay đổi khi di chuyển chuột băng qua hình ảnh (nó thường chuyển thành hình "bàn tay" khi ở trên một liên kết hoạt động được).

Bạn có thể "bỏ đi" hộp bao quanh bằng cách thêm thuộc tính vào tag `<img...>` như sau:

```
<a href="bigpict.gif"><img src="lilpict.gif" border=0>
```

Trong ví dụ này, hình ảnh `lilpict.gif` hoạt động như một siêu liên kết đến một hình ảnh lớn hơn là `bigpict.gif`. Thuộc tính `border=0` không có nghĩa gì cả nếu như tag `<img..>` không ở bên trong một tag `<a href=...</a>` khác.

Chúng ta đã có hai chỗ trong bài *Volcano Web* của chúng ta có hình ảnh inline siêu liên kết - Bạn có nhớ ở đâu không?

Cái đầu tiên trong tập tin `usa.htm`, trong đó một hình ảnh nhỏ của máy đo địa chấn liên kết đến một hình ảnh lớn hơn. Cái thứ hai là dấu mũi tên trong trang `msh.htm` để quay lại phần bài học.

1. Mở cả hai tập tin `usa.htm` và `msh.htm` trong trình soạn thảo.
2. Tìm ra nơi mà chúng ta đã đặt hình ảnh nhỏ có liên kết đến thứ khác.
3. Sửa đổi mỗi tag `<img src=....>` để nó là một siêu liên kết và có chứa một thuộc tính `border=0` . Lấy ví dụ, trong tập tin `msh.htm` nó sẽ trông giống như sau:
4.
5. <img src="../pictures/left.gif" alt="** " border=0>
   Return to Volcano Web

6. **Lưu trữ** và **Nạp** lại các tập tin `usa.htm` và `msh.htm` trong browser của bạn.

7. Nếu các hình nhỏ không còn đường viền và vẫn nối đến đúng nơi cần thiết khi bạn click vào, công việc của bạn coi như đã xong.

Các bullet cho Danh sách Không có Thứ tự

Trong bài 6 chúng ta đã tạo các danh sách không có thứ tự `<ul>...</ul>`. Web browser tự động đặt một dấu bullet phía trước mỗi mục của danh sách - và dấu bullet sẽ tự thay đổi nếu chúng ta tạo một danh sách nữa bên trong một danh sách. Với vài loại browser bạn có thể chỉ định trong HTML một trong ba loại bullet sẽ sử dụng bằng cách thêm thuộc tính vào tag `<ul>`:

```
<ul type=xxxx>
```

trong đó `xxxx` có thể là:

- type=circle
- type=square
- `type=disc` [bullet mặc định cho danh sách cấp ngoài cùng]

Hơn thế nữa! Bạn có thể thay đổi loại **bên trong** một danh sách bằng cách đặt một thuộc tính `type` trong tag `<li>`:

HTML	Nó nhìn thấy như thế nào

`<ul type=square>`
`<li>this is item 1`
`<li>this is item 2`
`<li>this is item 3`
`<li type=circle>`
`Hey! how about these bullets?`
`<li>this is another item`
`<li>and one more`
`<li type=disc>`
`Hey! how about these bullets?`
`</ul>`
	▪ this is item 1
	▪ this is item 2
	▪ this is item 3
	o Hey! how about these bullets?
	o this is another item
	o and one more
	• Hey! how about these bullets?

Lưu ý rằng loại được chỉ định trong tag `<li type=xxxx>` vẫn được sử dụng cho tất cả các tag `<li>` tiếp theo cho đến khi nào một loại bullet khác được chọn.

Bây giờ, chúng ta sẽ thay đổi loại bullet trong trang Research Projects của chúng ta (tập tin `proj.htm`).

1. Mở tập tin `proj.htm` trong trình soạn thảo.
2. Dòng đầu tiên trong tập tin này là một danh sách có thứ tự `<ol>...</ol>` nhưng bây giờ chúng ta đổi nó thành một danh sách không có thứ tự với bullet `circle`. Sửa đổi lại HTML cho danh sách đầu tiên thành:

```
<ul type=circle>
<li>Type of volcano
<li>Geographic location
<li>Name, distance, and population of nearest major city
<li>Date of most recent eruption and date of most
   destructive eruption
<li>Other events associated with the last eruption
   (earthquakes, floods, mudslides, etc)
</ul>
```

12. **Lưu trữ** và **Nạp lại** tập tin của bạn trong browser. Bạn có thể so sánh trang Web của bạn với ví dụ mẫu để biết nó nên xuất hiện như thế nào tại đây.

Các kiểu và Giá trị cho Danh sách Có thứ tự

Khi chúng ta tạo ra một danh sách có thứ tự `<ol>...</ol>` lần đầu tiên trong bài 6, chúng ta đã thấy cách browser tự động đánh số thứ tự cho các mục do tag `<li>` chỉ định. Chúng ta sẽ làm gì khi chúng ta không muốn sử dụng các số Ả rập (1,2,3...)? Tốt thôi, đây là câu trả lời, sử dụng thuộc tính `type=x` bên trong các tag `<ol>` và `<li>`:

Số Ả Rập	Chữ Hoa	Chữ Thường	Số La Mã Lớn	Số La Mã Nhỏ
<ol type=1>	**<ol type=A>**	**<ol type=a>**	**<ol type=I>**	**<ol type=i>**
1. Tôi đầu tiên!	A. Tôi đầu tiên!	a. Tôi đầu tiên!	I. Tôi đầu tiên!	i. Tôi đầu tiên!
2. Tôi thứ hai!	B. Tôi thứ hai!	b. Tôi thứ hai!	II. Tôi thứ hai!	ii. Tôi thứ hai!
3. Tôi thứ ba!	C. Tôi thứ ba!	c. Tôi thứ ba!	III. Tôi thứ ba!	iii. Tôi thứ ba!
4. Tôi thứ tư!	D. Tôi thứ tư!	d. Tôi thứ tư!	IV. Tôi thứ tư!	iv. Tôi thứ tư!
5. Tôi thứ năm!	E. Tôi thứ năm!	e. Tôi thứ năm!	V. Tôi thứ năm!	v. Tôi thứ năm!

Chúng ta đã thấy một ví dụ của việc sử dụng danh sách có thứ tự bên trong một danh sách có thứ tự khác khi trình bày trang - với thuộc tính `type` chúng ta có thể có những trang với dạng trình bày chuẩn như:

I. Cheese in Pre-Historic time
   A. Africa
      1. Afar Triangle
      2. East Coast
   B. Asia
   C. Europe
      1. France
         a. Cave paintings depicting cheese harvesting
         b. Burial rituals inferred from dried cheese remnants
      2. British Isles
   D. North America
II. Cheese in the Middle Ages
   A. King Arthur's Longhorn
   B. Sharp Cheddar for the Crusades
III. Cheese in the Space Age

Một thứ khác chúng ta có thể làm với danh sách có thứ tự là để nó bắt đầu đếm từ một giá trị không phải là 1. Để làm điều này chúng ta thêm thuộc tính `start=y` vào tag `<ol>`. Lưu ý rằng ngay cả chúng ta sử dụng các thuộc tính `type=x` khác, chúng ta vẫn chỉ định được giá trị bắt đầu y theo dạng như "2, 3, 10, 100, ...". Hãy xem vài ví dụ sau:

Số Ả Rập	Chữ Hoa	Chữ Thường	Số La Mả Lớn	Số La Mả Nhỏ
<ol type=1 start=11>	**<ol type=A start=11>**	**<ol type=a start=11>**	**<ol type=I start=11>**	**<ol type=i start=11>**
11. Tôi thứ 11!	K. Tôi thứ 11!	k. Tôi thứ 11!	XI. Tôi thứ 11!	xi. Tôi thứ 11!
12. Tôi thứ 12!	L. Tôi thứ 12!	l. Tôi thứ 12!	XII. Tôi thứ 12!	xii. Tôi thứ 12!
13. Tôi thứ 13!	M. Tôi thứ 13!	m. Tôi thứ 13!	XIII. Tôi thứ 13!	xiii. Tôi thứ 13!
14. Tôi thứ 14!	N. Tôi thứ 14!	n. Tôi thứ 14!	XIV. Tôi thứ 14!	xiv. Tôi thứ 14!
15. Tôi thứ	O. Tôi thứ	o. Tôi thứ	XV. Tôi thứ	xv. Tôi thứ

15! 15! 15! 15! 15!

Và cuối cùng bạn có thể thay đổi thứ tự số **bên trong** một danh sách bằng cách thêm thuộc tính `value=z` vào tag `<li>`. Hãy xem ví dụ sau:

HTML	Nó xem Như thế nào
<ol type=A start=5> Important Cheese Laws Cheese Curing Act of 1905 Milk Content Ruling of 1923 <p> <li value=12>Cheese Import Tariff of 1942 Cheese Freshness Codes of 1942 <p> <li value=1>Cheese Values Act of 1789	*Important Cheese Laws* E. Cheese Curing Act of 1905 F. Milk Content Ruling of 1923 L. Cheese Import Tarif of 1942 M. Cheese Freshness Codes of 1942 A. Cheese Values Act of 1789

Không thể nói rõ lắm (nhất là trong ví dụ này) khi nào bạn có thể cần phải sử dụng các tag này - chỉ cần nhớ rằng bạn có khả năng làm điều đó khi viết trang web của bạn. Chúng ta sẽ minh họa thêm một lần nữa trong trang Research Projects (tập tin `proj.htm`). Nếu bạn nhớ lại trong bài về bảng chúng ta đã chia danh sách không có thứ tự của những site cần tham khảo thành hai cột. Hãy đổi nó thành một danh sách có thứ tự và sử dụng thuộc tính `type` để liệt kê chúng, sử dụng chữ thường để đánh số thứ tự. Vì thật sự chúng ta có đến **hai** danh sách, hãy xem cách tại sao chúng ta quyết định sử dụng thuộc tính `start`.

1. Mở tập tin `proj.htm` trong trình soạn thảo.
2. Hãy tìm bảng chúng ta đã tạo dưới phần **References**. Thay đổi cả hai tag `<ul>` và `</ul>` thành tag `<ol type=a>` và `</ol>`
3. Bây giờ chúng ta đã có 5 mục trong danh sách thứ nhất, vì thế chúng ta muốn danh sách thứ 2 bắt đầu đánh số từ 6. Sửa đổi tag `<ol>` thành:

   <ol type=a start=6>

4. **Lưu trữ** và **Nạp lại** trong browser của bạn. So sánh trang của bạn với trang mẫu để xem danh sách của bạn nên xuất hiện như thế nào. Cột các mục đầu tiên được đánh thứ tự từ "a" đến "e" còn cột thứ hai từ "f" đến "k"

Kiểm tra lại công việc của bạn

So sánh trang của bạn với ví dụ mẫu để biết nó nên xuất hiện như thế nào. Nếu trang của bạn khác với mẫu, kiểm tra lại các văn bản bạn đã đánh trong trình soạn thảo.

Xem lại

Xem lại những chủ đề sau :

1. Làm thế nào bỏ được hộp liên kết bao quanh một hình ảnh khi hình ảnh này là một siêu liên kết?
2. Làm thế nào thay đổi dấu bullet cho một danh sách không có thứ tự ?
3. Browser có sử dụng cùng một dấu bullet cho tất cả các danh sách không? Nếu không, làm thế nào để đổi dấu bullet cho các danh sách con?
4. Bạn có thể tạo một danh sách mà mỗi mục được đánh thứ tự bằng một số La mã không?

Thực tập tự do

Hãy tạo kinh nghiệm với các kiểu dấu bullet và đánh số thứ tự khác nhau cho danh sách trong trang web riêng của bạn. Bạn có nghĩ ra vài cách cố định cho các tính chất này không? Bạn có thể tạo ra cách trình bày phức tạp như dạng trình bày "chuẩn" không? Mọi điều về tag
...
bây giờ trở nên khá phức tạp!

23. Bản đồ Hình ảnh Liên kết

Hãy làm cho **các phần** khác nhau trong một hình ảnh inline hoạt động như một liên kết dựa trên bản đồ hình ảnh của client ...

Mục đích

Sau bài học này bạn có thể:

- Hiểu được sự khác nhau giữa tiến trình về phía server (server-side process) và tiến trình về phía client (client-side process).
- Tạo được một hình ảnh inline có những phần khác nhau làm siêu liên kết tới những trang Web, hình ảnh, và những site khác trên Internet.

Bài học

Lưu ý : Nếu bạn chưa có tài liệu từ những bài trước, hãy download ngay bây giờ. Bạn cũng có thể xem trước trang kiểm tra để biết browser của bạn có hỗ trợ những tag được sử dụng trong bài này không.

Trong bài 7a chúng ta đã biết làm thế nào để viết HTML cho việc đặt một hình ảnh inline trong trang Web, và trong bài 8e chúng ta cũng đã biết rằng có thể làm cho toàn bộ hình ảnh hoạt động như là một siêu liên kết đến một mục có liên quan khác. Từ những ngày đầu tiên của Web, đã có một cách để những phần khác nhau trong một hình ảnh inline trở thành siêu liên kết -- được biết dưới tên gọi là "Bản đồ hình ảnh liên kết" (clickable image map).

Nhưng mãi cho đến thời gian gần đây, bản đồ hình ảnh liên kết vẫn đòi hỏi một vài thứ được xử lý từ trên web server. Đây là cách hoạt động của nó:

1. Người đọc xem trang có bản đồ hình ảnh liên kết và... nhấn vào phần thấp nhất bên phải của hình ảnh.
2. Web browser nói, "À! Tôi đã nhận được một sự nhấn chuột trên hình này! Tôi gởi một thông báo đến Web server - một người nào đó đã click vào tọa độ này của hình."
3. Web server nói, "Hmmm. Tôi đã nhận được các tọa độ của hình này - Các tọa độ này được dò tìm trong một tập tin do tập tin HTML qui định... Tập tin này nói, Được rồi, các tọa độ này được tìm thấy trong một vùng chữ nhật và hãy gởi đến người xem URL này của tôi" Rồi server gởi thông tin này lại cho web browser.
4. Web browser nói, "Đồng ý! server đã nói hãy đến URL này - Chúng ta đi đến đó!"

Đây là một tiến trình **về phía server**; một tiến trình khá hiệu quả và server có thể xử lý nó trong vài phần giây. Nhưng nó cũng có nghĩa là sử dụng một bản đồ hình ảnh liên kết luôn luôn phải truy xuất vào server.

Spyglass là browser đầu tiên có khả năng xử lý việc tính toán và truyền chính xác URL dựa trên các tọa độ mà việc này được thực hiện ngay bên trong phần HTML của trang. Đây là một tiến trình **về phía client**. Tất cả việc làm chúng ta vừa nói ở trên bây giờ chỉ được giải quyết nhờ vào browser và phần HTML của trang! Để biết thêm chi tiết về bản đồ hình ảnh, hãy xem Imagemap Help Page (IHiP).

Dạng HTML cho một bản đồ hình ảnh liên kết về phía client (client-side clickable image map) là:

```
<img src="image.gif" usemap="#map_name">
```

trong đó `image.gif` là tên tập tin hình ảnh và `map_name` là một liên kết anchor (xem bài 8d) ở đâu đó trong cùng tập tin HTML và có dạng:

```
<map name="map_name">
<area shape="rect" coords="x1,y1,x2,y2" HREF=URL1>
<area shape="rect" coords="x1,y1,x2,y2" HREF=URL2>
    :
    :
</map>
```

Mỗi dòng trong tag `<map>...</map>` xác định một vùng "nóng" khác nhau dựa trên các toạ độ được chỉ định bằng `coords=`. Trị `x1,y1` cho biết tọa độ tính theo chiều ngang và chiều cao của điểm ở góc trên bên trái (tính tương đối so với góc trên bên trái của toàn hình ảnh), còn `x2,y2` cho biết tọa độ tính theo chiều ngang và chiều cao của điểm ở góc dưới bên phải. `URL1, URL2, ...` là các URL (hoặc tên tập tin cục bộ) cho vùng tương ứng khi được click.

Chú ý: Để chỉ định các tọa độ cho vùng nóng, bạn cần sử dụng một vài loại chương trình đồ họa. Yahoo có vài trình tiện ích khác giúp bạn làm việc này dễ dàng hơn. Với bài này, chúng tôi sẽ cung cấp sẵn các tọa độ chính xác cho bạn.

Trong bài này chúng ta sẽ thêm vào trang Volcano Terminology (tập tin `term.htm`) một hình ảnh chỉ ra các các núi lửa khác nhau đã hoạt động lại. Mỗi vị trí trong hình sẽ tạo siêu liên kết đến một web site bên ngoài. Ngoài ra chúng ta sẽ tạo thêm hai liên kết nữa đến các tập tin riêng (cục bộ).

1. Trước tiên, bạn cần chép lại một bản sao của tập tin ảnh dùng Trung tâm Download Hình ảnh Bài 23. Cất tập tin này thành `volc.jpg` và chắc chắn rằng bạn đã lưu nó vào trong subdirectory/folder `pictures`.
2. Mở tập tin `term.htm` trong trình soạn thảo.
3. Phía dưới đoạn cuối cùng ("...a historically active volcano on the island of Martinique. <p>"), thêm đoạn HTML sau:
4.
5. There are many different types of volcanic eruptions and
6. landforms. They can be classified according to the degree
7. of "explosiveness" and the height of the eruption column:
8. <p>
9.
10.
11. Investigate each type by clicking on a picture
12.
13. <img src="../pictures/volc.jpg" usemap="#volcmap" border=0>
14. <p>
15.

    Chú ý: Chúng ta đã thêm vào một hình ảnh inline cùng với việc tham khảo một tập các tọa độ kết hợp với tên "volcmap". Tag `<center>...</center>` đưa hình ảnh vào giữa trang (xem bài 20). Thuộc tính `border=0` bên trong tag `<img>` hủy bỏ việc thể hiện hộp bao quanh hình ảnh siêu liên kết.

16. Kế tiếp chúng ta sẽ thêm đoạn HTML cho các tọa độ của bản đồ liên kết. Đoạn này có thể để bất kỳ chỗ nào trong tài liệu HTML - nó là phần HTML **không** được hiển thị

trong web browser. Chúng tôi đề nghị đặt nó ngay dưới phần HTML bạn đã thêm vào trong bước vừa rồi:

```
<map name="volcmap">
<area shape="rect" href="
http://volcano.und.nodak.edu/vwdocs/frequent_questions/grp7/europe/question308.html"
   coords="48,46,204,153">
<area shape="rect" href="explode.html"
   coords="0,66,26,227">
<area shape="rect" href="height.html"
   coords="95,283,378,309">
<area shape="rect" href="http://www.geo.mtu.edu/volcanoes/pinatubo/"
   coords="321,154,468,261">
<area shape="rect" href="http://www.geo.mtu.edu/~boris/STROMBOLI_main.html"
   coords="172,155,318,274">
<area shape="rect" href="http://www.soest.hawaii.edu/hvo/"
   coords="36,155,168,276">
<area shape="rect" href="http://www.geo.mtu.edu/volcanoes/santamaria/"
   coords="90,3,343,123">
</map>
```

Chú ý: Bạn nên biết rằng 5 trong 7 vùng được định nghĩa kết nối đến các site internet ở xa. Hai cái còn lại đến các tài liệu cục bộ mà chúng ta sẽ tạo ra trong bước kế tiếp.

33. Cất lại tập tin `term.htm`.
34. Bây giờ chúng ta sẽ tạo ra hai tập tin HTML nữa được liên kết từ image map. Tạo ra một tài liệu mới trong trình soạn thảo và đặt vào đó:

```
<html>
<head>
<title>Explosiveness</title>
</head>
<BODY BGCOLOR=#000000 TEXT=#EEEEBB
LINK=#44DDFF VLINK=#00FF88>
<H5>Volcano Web /
<A HREF="index.htm">Index</A> /
<A HREF="term.htm">Volcano Terminology</A> /
</H5>

<h1 align=center>Explosiveness</h1>
The <b>explosiveness</b> of an observed volcanic
eruption is estimated by the calculated force of
the eruption. Experiments have shown that when water
comes in contact with hot magma, the eruption is
much stronger- this is known as <b>hydro-volcanism</b>.
<p>
For pre-historic eruptions, the explosiveness is estimated
by the degree of fragmentation of small volcanic particles.
A more explosive eruption will "shatter" volcanic tephra
much more then a less explosive eruption.
<p>
<a href="term.htm">
<img src="../pictures/left.gif" alt="** " border=0>
Return to <i>Volcano Web</i></a>

<HR>
<ADDRESS>
<B><A HREF="index.htm">Volcano Web</A> :
```

```
<A HREF="term.htm">Volcano Terminology</A> :
Explosiveness</B> <p>

created by Lorrie Lava,
<A HREF="mailto:lava@pele.bigu.edu">
lava@pele.bigu.edu</A><br>
Volcanic Studies, <A HREF="http://www.bigu.edu/">
Big University</A><p>
<TT>last modified: April 1, 1995</TT>
</ADDRESS>
<p>
<tt>URL: http://www.bigu.edu/web/explode.htm</tt>
<p>
</body>
</html>
```

Cất lại tài liệu này trong cùng thư mục với các tập tin HTML khác và gọi nó là `explode.htm`

Bây giờ tạo tài liệu thứ hai và đưa vào đó:

```
<html>
<head>
<title>Height of Eruption Column</title>
</head>

<BODY BGCOLOR=#000000 TEXT=#EEEEBB
LINK=#44DDFF VLINK=#00FF88>
<H5>Volcano Web /
<A HREF="index.htm">Index</A> /
<A HREF="term.htm">Volcano Terminology</A> /
</H5>

<h1 align=center>Height of Eruption Column</h1>
The <b>height</b> of a volcanic eruption cloud
(in kilometers) is taken from direct observation
or from estimates based upon historic accounts.
<p>
For pre-historic eruptions, this scale is calculated
based upon the <b>dispersal</b> of the volcanic products-
i.e how far and wide they are scattered. Volcanic
eruptions that have very tall columns will spread
tephra far and wide.
<p>
<a href="term.htm">
<img src="../pictures/left.gif" alt="** " border=0>
Return to <i>Volcano Web</i></a>

<HR>
<ADDRESS>
<B><A HREF="index.htm">Volcano Web</A> :
<A HREF="term.htm">Volcano Terminology</A> :
Height of Eruption Column</B> <p>

created by Lorrie Lava,
<A HREF="mailto:lava@pele.bigu.edu">
lava@pele.bigu.edu</A><br>
Volcanic Studies,
<A HREF="http://www.bigu.edu/">
```

```
121. Big University</A><p>
122. <TT>last modified: April 1, 1995</TT>
123. </ADDRESS>
124. <p>
125. <tt>URL: http://www.bigu.edu/web/height.htm</tt>
126. <p>
127. </body>
128. </html>
```

Cất lại tài liệu này trong cùng thư mục với các tập tin HTML và gọi nó là `height.htm`

129. Bây giờ mở tập tin `term.htm` trong browser của bạn. Hình các núi lửa khác nhau sẽ xuất hiện và khi bạn di chuyển chuột qua các vùng khác nhau của hình ảnh hay vài chỗ cuối trang, dấu nhắc sẽ phải chuyển thành hình "bàn tay", tại đó URL của vùng "nóng" cũng phải được hiển thị.

Bạn nên kiểm tra trang web của bạn với ví dụ mẫu để biết trang bản đồ hình ảnh liên kết nên được thấy như thế nào.

Điều cần xem lại

Cần nhớ rằng đây không phải là sự phát triển mới của HTML, bạn nên nghĩ đến việc có những người xem sử dụng một web browser không hỗ trợ client-side image maps. Tài liệu của NetScape cho vài ví dụ để xử lý trường hợp này. Nếu bạn sử dụng bản đồ hình ảnh liên kết về phía server, bạn có thể dùng nó cho các browser không hỗ trợ việc thông dịch về phía client, khi đó nó phải tham khảo đến server. Chúng tôi đã sử dụng cách này cho bản đồ hình ảnh liên kết của chúng tôi tại Maricopa Community Colleges.

Với trường hợp của chúng ta, chúng ta có thể tạo một trang đặc biệt để thông báo cho các web browser không có hỗ trợ bản đồ hình ảnh liên kết về phía client:

1. Trước hết chúng ta cần tạo một tập tin HTML mới gọi là `nomap.htm`. Mở tập tin này trong trình soạn thảo và đưa vào đó:

```
2.
3.    <html>
4.    <head>
5.    <title>No Image Map Available</title>
6.    </head>
7.    <BODY BGCOLOR = #000000 TEXT=#EEEEBB
8.    LINK=#44DDFF VLINK=#00FF88>
9.    <H5>Volcano Web /
10.   <A HREF="index.htm">Index</A> /
11.   <A HREF="term.htm">Volcano Terminology</A> /
12.   </H5>
13.
14.   <h1 align=center>Sorry!</h1>
15.   Apparently your web browser does not support
16.   client-side image maps. You can still reach the
17.   i
18.   nformation by following these links:
19.   <ul>
20.   <li><A HREF="explode.htm">Explosiveness</A>
21.   <li><A HREF="height.htm">Height of Eruption Column</A>
22.   <p>
23.   <li><A HREF="
24.   http://volcano.und.nodak.edu/vwdocs/frequent_questions/grp7/europe/question308.
      html"
```

```
25.     >Surtseyan</A>
26.    <li><A HREF="http://www.geo.mtu.edu/volcanoes/santamaria/">Phreato-
    Plinian</A>
27.    <li><A HREF="http://www.soest.hawaii.edu/hvo/">Hawaiian</A>
28.    <li><A
    HREF="http://www.geo.mtu.edu/~boris/STROMBOLI_main.html">Strombolian</A>
29.    <li><A HREF="http://www.geo.mtu.edu/volcanoes/pinatubo/">Plinian</A>
30.    </ul>
31.    <p>
32.    <a href="term.htm">
33.    <img src="../pictures/left.gif" alt="** " border=0>
34.    Return to<i>Volcano Web</i></a>
35.
36.    <HR>
37.    <ADDRESS>
38.    <B><A HREF="index.htm">Volcano Web</A> :
39.    <A HREF="term.htm">Volcano Terminology</A> :
40.    No Image Map</B> <p>
41.
42.    created by Lorrie Lava,
43.    <A HREF="mailto:lava@pele.bigu.edu">
44.    lava@pele.bigu.edu</A><br>
45.    Volcanic Studies,
46.    <A HREF="http://www.bigu.edu/">Big University</A><p>
47.    <TT>last modified: April 1, 1995</TT>
48.    </ADDRESS>
49.    <p>
50.    <tt>URL: http://www.bigu.edu/web/nomap.htm</tt>
51.    <p>
52.    </body>
</html>
```

Cất lại tập tin này với tên `nomap.htm`

Chú ý: Hãy xem cách chúng ta cung cấp cho người xem cùng một loại thông tin như trong bản đồ hình ảnh liên kết. Một trang web tốt không giới hạn bớt thông tin đối với một người nào cả vì họ có quyền sử dụng các loại browser khác nhau.

53. Bây giờ mở tập tin `term.htm` trong trình soạn thảo.
54. Tìm dòng sau:

```
<img src="../pictures/volc.jpg" usemap="#volcmap" border=0>
```

và thay nó thành

```
<a href="nomap.htm">
<img src="../pictures/volc.jpg" usemap="#volcmap" border=0>
</a>
```

Chú ý: Bạn có thể phân tích thêm về HTML này - nếu web browser có thể hiểu bản đồ hình ảnh liên kết về phía client, nó sẽ thực hiện điều đó. Ngược lại, toàn bộ hình ảnh là một siêu liên kết đến trang `nomap.htm` page.

55. **Cất lại** tập tin này và **Nạp lại** nó vào web browser của bạn.

Kiểm tra lại Công việc của Bạn

So sánh các trang web của bạn với ví dụ mẫu để biết nó nên xuất hiện như thế nào. Nếu các trang của bạn khác với trang mẫu hay các liên kết siêu văn bản hoạt động không đúng, hãy xem lại các văn bản bạn đã đánh vào trong trình soạn thảo.

Xem lại

Các vấn đề nên xem lại cho bài này:

1. Sự khác nhau giữa bản đồ hình ảnh liên kết về phía client và bản đồ hình ảnh liên kết về phía server?
2. Tag `<map>...</map>` dùng làm gì?
3. Bạn giải quyết như thế nào khi một web browser không hiểu bản đồ hình ảnh liên kết về phía client?

Thông tin bổ sung

Chú ý rằng sự tham khảo các thông tin tọa độ trong tag `<map>...</map>` không nhất thiết phải ở trong cùng một tài liệu HTML. Thuộc tính `usemap=` có thể liên kết đến một tập tin HTML khác - ví dụ như:

```
<img src="../pictures/buttonbar.gif" usemap="maps.htm#bmap3">
```

trong đó maps.htm có thể là một tập tin chứa các tag `<map>...</map>` cho một vài tập tin. Bạn có thể làm được điều này không?

Hãy xét trường hợp bạn tạo ra vài (hay nhiều) trang mà ở trên đỉnh của trang có một hình với vài button liên kết đến các trang khác. Nếu bạn sử dụng cách này, bạn có thể tham khảo đến chỉ một hình và một tập tin có các tọa độ. Khi bạn phải thay đổi thanh hoa tiêu của bạn (một hình mới? một sự thay đổi URL cho một button?) **bạn chỉ cần điều chỉnh một tập tin!** (so sánh với việc thay đổi các tag `<map>...</map>` nếu chúng được thêm vào mỗi trang web).

Thực tập tự do

Tìm chỗ trong trang web của bạn có thể thêm vào một clickable image map trong bài hay trong phần hoa tiêu. Đừng bao giờ vứt bỏ ngay bản cũ! Bạn phải làm thử vài lần để xác định được các vùng "nóng" (bạn có thể phỏng đoán nếu bạn thật sự liều mạng). Theo các ví dụ trong bài này để viết HTML cho bản đồ hình ảnh liên kết.

LẬP TRÌNH WEB VỚI PHP

Trung tâm đào tạo Lập trình viên quốc tế Hue-Aptech

Địa chỉ: 06 Lê Lợi – TP Huế

Điện thoại: 054. 3823077

Email: hue-aptech.@huesoft.com.vn

Website: www.huesoft.com.vn

Nội dung

- Lập trình php và giới thiệu các framework phổ biến.
  - Giới thiệu về php
  - Ngôn ngữ lập trình php.
  - Giới thiệu về Mysql.
  - Các framework phổ biến.
- Thực hành

Giới thiệu về php

- Định nghĩa
  - PHP: Hypertext Preprocessor
  - PHP là ngôn ngữ lập trình được thực hiện trên máy chủ.
  - Kết quả xử lý được trình diễn theo dạng HTML.
- Lịch sử ra đời
  - PHP được giới thiệu năm 1994. Tác giả là R.Lerdoft.
  - PHP có thể kết hợp với nhiều
  - Đến nay, php đã được công bố với 4 phiên bản.

APTECH COMPUTER EDUCATION Huesoft

Ngôn ngữ lập trình php

- Cú pháp
  - Cú pháp căn bản
  - Chú thích
- Biến
- Biến dạng chuỗi
- Toán tử
  - Số học
  - Gán
  - So sánh
  - Logic

Ngôn ngữ lập trình php(TT)

- Các lệnh điều khiển
  - If…else
  - Switch
- Mảng
  - Mảng kiểu số
  - Mảng kết hợp
  - Mảng đa chiều
- Điều khiển lặp
  - While, Do…while, For, Foreach

APTECH COMPUTER EDUCATION Huesoft

Ngôn ngữ lập trình php(TT)

- Hàm trong php
  - Hàm cơ bản
  - Hàm chứa tham số
  - Hàm trả về giá trị
- Xử lý form trong php
  - GET
  - POST
  - REQUEST
- Hàm include file
  - Include()
  - Require()

Giới thiệu về Mysql

- Mysql là hệ quản trị CSDL.
- Mysql có cùng cách truy xuất và mã lệnh tương tự SQL

APTECH COMPUTER EDUCATION Huesoft

Các framework phổ biến

- Giới thiệu vê framework.
  - Cung cấp cấu trúc cơ bản để xây dựng website.
  - Giúp xây dựng các ứng dụng ổn định và chính xác.
- Các framework phổ biến.
  - Zend framework
  - CakePhp.
  - Symfony.

Session và Cookie

- Session
  - Bắt đầu session
  - Khởi tạo session
  - Giá trị cho session
  - Hủy session
- Cookie
  - Khởi tạo cookie
  - Giá trị của cookie
  - Hủy cookie.

Thực hành

- Sử dụng session login hệ thống.
- Hiển thị dữ liệu từ CSDL ra trình diễn.

www.huesoft.com.vn

LẬP TRÌNH PHP

(CĂN BẢN)

1. Giới thiệu

- PHP (viết tắt hồi quy "PHP: Hypertext Preprocessor") là một ngôn ngữ lập trình kịch bản hay một loại mã lệnh chủ yếu được dùng để phát triển các ứng dụng viết cho máy chủ Webserver.
- Do PHP được phát triển chỉ để dành cho ứng dụng web nên rất dễ dàng nhúng mã vào trang HTML. Tốc độ nhanh, nhỏ gọn, cú pháp giống C, dễ học và thời gian xây dựng sản phẩm tương đối ngắn hơn so với các ngôn ngữ khác nên PHP đã nhanh chóng trở thành một ngôn ngữ lập trình web rất phổ biến.
- Ngôn ngữ, các thư viện, tài liệu gốc của PHP được xây dựng bởi cộng đồng nhưng cũng có sự đóng góp rất lớn của Zend Inc., công ty do các nhà phát triển cốt lõi của PHP lập nên.
- Nói đơn giản: PHP là ngôn ngữ dạng script thực thi trên webserver nhằm tạo ra trang web động
- Thẻ <? và thẻ ?> đánh dấu bắt đầu và kết thúc của phần mã PHP, qua đó máy chủ biết để xử lý và dịch mã cho đúng.
- Mỗi dòng kết thúc bằng dấu ;
- Trong trang html, các lệnh php nằm trong tag <? … ?> hoặc <?php …?>
- Các khối lệnh nằm trong { … }
- Tên mở rộng của file là php
- Chú thích

```
// : 1 dòng
```

```
/ *
…      Khối lệnh
*/
```

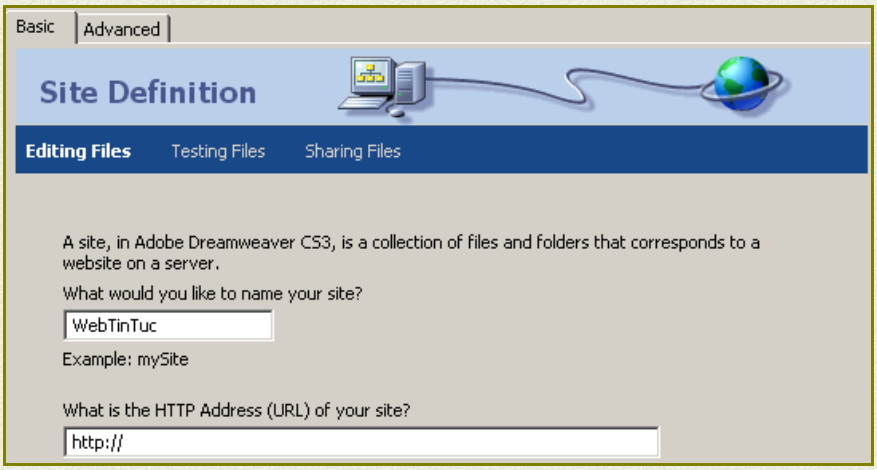
- File PHP chỉ chạy khi nó được chuyển giao từ webserver. Vì vậy, muốn thực thi file php, bạn phải request nó thông qua webserver (nghĩa là mở browser, gõ http://localhost/trangdong.php. Do đó không thể nhúp đúp vào file php để xem nó ở localcal như trang html được. Để chạy file PHP :

  + Nếu máy của bạn cài IIS : tạo file trong folder C:\Inetpub\wwwroot

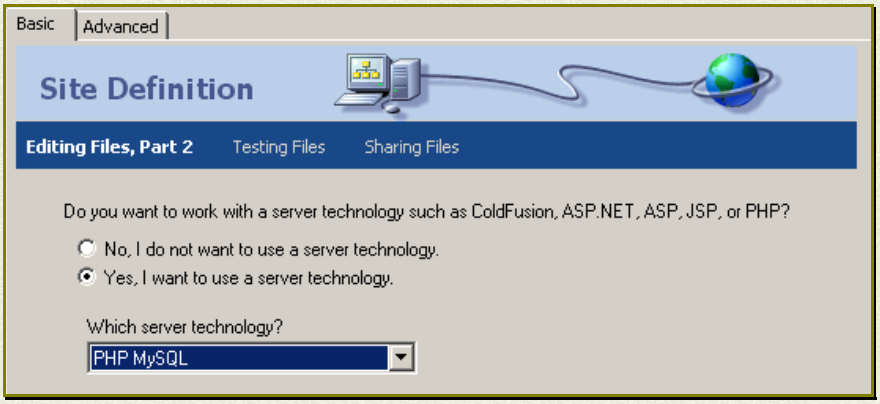
  + Nếu máy của bạn cài AppserV : tạo file trong folder C:\AppserV\www

2. Tạo website để lập trình PHP

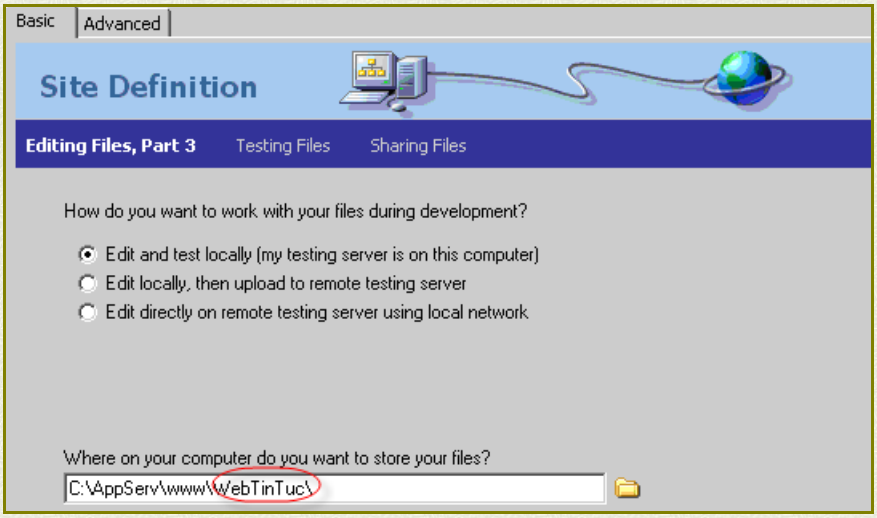
- Yêu cầu: folder chứa site của bạn phải nằm trong folder gốc của webserver
- Giả sử folder site của bạn là WebTinTuc, đặt trong C:\AppSerV\www. Dưới đây là hướng dẫn cách tạo site PHP
- B1: Trong Dreamweaver, nhắp menu Site ➔ New Site
- B2: Gõ tên Site (WebTinTuc) rồi nhắp Next



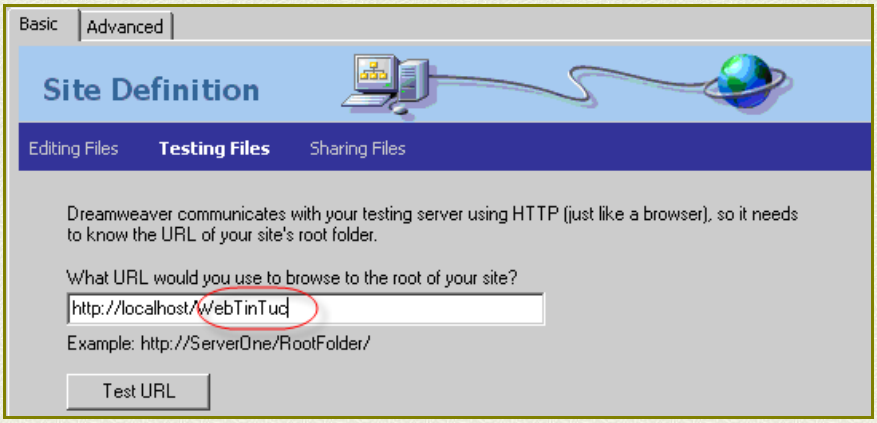
- B3: Chọn Yes rồi chọn PHP MySQL



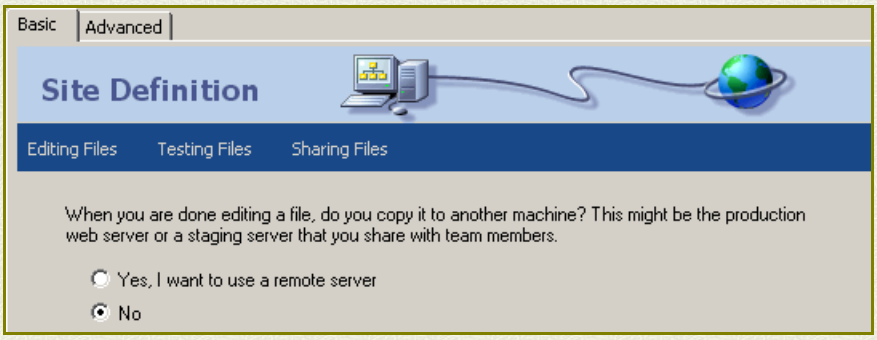
- B4: Chọn Edit and Test locally… rồi chọn folder chứa site ở bên dưới (C:\AppServ\www\WebTinTuc)



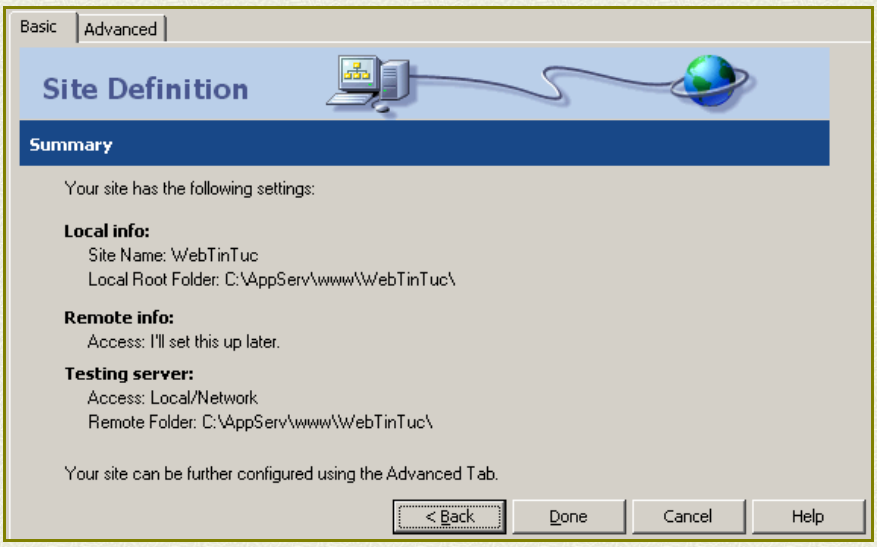
- B5: Gõ địa chỉ website: http://localhost/WebTinTuc



- B6: Nên chọn **No** (Chưa cần kết nối đến đến remote server lúc này) rồi nhắp Next



- B7: Nhắp Done để kết thúc tạo site



- Ghi chú: không nhất thiết phải tạo folder WebTinTuc cho site của bạn nhưng phải đặt trong folder gốc của webserver và tên folder trong B5 và B4 phải giống nhau.

3. Tạo trang PHP

- Nhắp menu File ➔ New ➔ Chọn PHP ➔ Create

4. Biến

- Biến dùng để chứa các giá trị trong quá trình tính toán
- Không cần khai báo kiểu dữ liệu lúc khai báo biến
- Các kiểu dữ liệu mà biến có thể lưu:

  + Số nguyên : int

+ Số thực : float, double

+ Chuỗi : string

+ Object

+ Logic : True hoặc false

+ Kiểu mảng: Array

+ Null : giá trị đặc biệt, nhằm thể hiện tính rỗng, chưa được gán gì

- Khai báo :

$TenBien=<giatri> ;

```
<?
$hoten=  "Tèo";          //Tạo biến hoten và gán giá trị
$dem=1;                  //Tạo biến dem và gán giá trị
```

- Lệnh **echo**

Dùng để hiện giá trị của 1 biến, hiện 1 chuỗi

- Ví dụ :

```
<?
echo $hoten;             //Hiện giá trị biến hoten
echo "Hello";            //Hiện chữ Hello, không hiện dấu nháy
?>
```

5. Hằng

- Là đối tượng chứa giá trị cố định
- Khai báo:

define("TenHang",GiaTri);

- Ví dụ:

```
<hr>
<?
      define("tygia",17000);
      $usd=10;
      $vnd=$usd*tygia;
      echo "VND=$vnd ";
?>
```

6. Các phép toán

- Số học: +. -, *, /, % (chia lấy dư)
- So sánh: ==, >=, <=, >, < , !=
- Ghép chuỗi: .
- Logic: !, && , ||
- Tăng 1 : ++
- Giảm 1 : --
- Gán: =

```
<?
	$soluong=3; //Khai báo biến soluong và gán giá trị
	echo $soluong; //3
	$soluong =5; // biến soluong đã có, chỉ gán giá trị
	echo $soluong; //5
?>
```

7. Lệnh if:

a. Lệnh if đơn giản:

```
if (điều kiện) {
	Khối Lệnh 1;
}
else {
	Khối Lệnh 2;
}
```

- Phần else và khối lệnh 2 có thể không có
- Nếu khối lệnh 1 khối lệnh 2 chỉ 1 gồm lệnh thì có thể bỏ cặp { }
- Khối lệnh có thể là mã lệnh php hoặc mã lệnh html
- Ví dụ 1 :

```
<hr/>
<? $diem=6;
if ($diem>=5) echo "Đậu";
else echo "Rớt";
?>
```

- Ví dụ 2 : Sử dụng template khác nhau cho trang tùy theo thời gian
  - Mở file index1.html rồi save as thành index1.php
  - Qua chế độ code, đến dòng 21 sẽ thấy đoạn code

```
<link href="c1.css" rel="stylesheet" type="text/css">
```

  - Bổ sung để có kết quả sau:

```
<link href="<? if (date('H')<=12) echo "c1.css"; else echo "c2.css"; ?>" rel="stylesheet" type="text/css">
```

  - F12 test thử

b. Lệnh if...elseif..else

- Cú pháp

```
if (Điều kiện1) {
    Khối lệnh 1;
}
elseif (Điều kiện2) {
    Khối lệnh 2
}
…
else {
    Khối lệnh
}
```

- Ví dụ:

```
<hr/>
<?
$diem=4;
if ($diem>=8) echo "Giỏi";
elseif ($diem>=7) echo "Khá";
elseif ($diem>=5) echo "Trung bình";
else echo "Kém";
?>
```

c. Lệnh if ngắn gọn

- Cú pháp:

```
(<dkien>)?<giá trị đúng>:<giá trị sai>;
```

- Ví dụ:

```
<hr/>
<?
        $dtb=5;
        echo ($dtb>=5)? "Được lên lớp":"Ở lại";
?>
```

8. Vòng lặp for:

- Là lệnh lặp 1 khối lệnh với số lần lặp nhất định
- Cú pháp:

```
for (<biến đếm> ; <Điều kiện dừng> ; <Tăng/giảm biến đếm>) {
        Khối lệnh.
}
```

- Ví dụ:

```
<?
        for ($i=1;$i<=10;$i++)
        {
                echo "i=$i<br>";
        }
?>
```

9. Vòng lặp while :

- Là lệnh lặp khối lệnh số lần lặp không xác định.
- Cú pháp:

```
while (<Điều kiện>){
        Khối lệnh
}
```

- Điều kiện lặp được kiểm tra trước, do đó có thể không thực hiện lần nào.
- Ví dụ:

```
<hr>
<?      $tong=0; $i=1;
        while ($i<=10) {
                $tong+=$i;
                $i++;
        }
        echo "Tổng=$tong";
?>
```

10. Vòng lặp Do...While:

- Là lệnh lặp khối lệnh số lần lặp không xác định.

```
do {
        Khối lệnh
} while (<Điều kiện>);
```

- Điều kiện lặp được kiểm tra sau, do đó thực hiện ít nhất 1 lần.
- Ví dụ:

```
<?      $tong=0; $i=1;
        do {
                $tong+=$i;
                $i+=2;
        }while ($i<=10);
        echo "Tổng các số lẻ=$tong";
?>
```

11. Hàm (Function)

a. Giới thiệu:

Hàm là 1 khối lệnh được đặt tên để gọi sử dụng nhiều lần. PHP có khoảng hơn 700 hàm. Chúng ta cũng có thể tự định nghĩa thêm các hàm mới.

Hàm có thể trả về 1 giá trị 1 giá trị nào đó sau khi tính toán hoặc không

b. Khai báo:

```
function <Tên hàm>(<tham số 1>,<tham số 2>,...){
        .....
        [return giá trị;]
}
```

c. Ví dụ:

- Tạo file ham.php
- Định nghĩa hàm: gõ ở đâu đó (trong tag head):

```
<?
function solonnhat($a,$b,$c) { //php không chú ý gì đến các lệnh trong hàm cho đến khi hàm được gọi
      $m=$a;
      if ($b>$m) $m=$b;
      if ($c>$m) $m=$c;
      return $m;
}
?>
```

- Sử dụng hàm: trong tag body

```
Số lớn nhất là :
<?
      $x=8; $y=12; $z=10; $m=4; $n=19;
      $t=solonnhat($x, $y,$z);
      $s=solonnhat($t, $m,$n);
      echo $s;
?>
```

- F12 xem thử (kết quả là ?)

d. Include file

- Người ta thường đưa các hàm có cùng 1 loại (như xử lý chuỗi, tính toán, tiền tệ, bảo mật…) vào 1 file php riêng và include vào trang web khi cần.
- Cú pháp:

```
include "TenFile" ;          hoặc
require "TenFile" ;          hoặc
include_once "TenFile" ;     hoặc   //để đảm bảo include chỉ 1 lần, nếu đã include rồi thì php sẽ không
require_once "TenFile" ;            // không include nữa, nhằm để tránh trùng tên hàm
```

- Tênfile thường là file php, cũng có thể là file html
- Ví dụ:
  - Tạo file ngay.php, xóa hết mọi code html rồi định nghĩa hàm sau:

```
<?php
function LucNayLa() { // Chuyển giờ hệ thống sang tiếng Việt
        $anh = array ( "Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun", "am", "pm", ":" );
        $viet = array ( "Thứ hai", "Thứ ba", "Thứ tư", "Thứ năm", "Thứ sáu", "Thứ bảy", "Chủ nhật", "
phút, sáng", " phút, chiều", " giờ " );
        $timenow = gmdate("D, d/m/Y - g:i a.", time() + 7*3600);
        $t = str_replace( $anh, $viet, $timenow);
        return $t;
}
?>
```

  - Tạo 1 trang php mới, lưu với tên là **testInclude.php**. Vào 1 trong tag body, gõ:

```
<?
include  "ngay.php";
echo LucNayLa();
?>
```

  - F12 xem thử

12. String (chuỗi)

Là 1 dãy ký tự trong nháy kép hoặc nháy đơn

a. Chiều dài chuỗi

strlen(chuỗi) ;//Trả về số ký tự trong chuỗi

```
<hr/>
<?
$a="Chao ban";
echo strlen($a) ; // 8
?>
```

b. Trích chuỗi con

substr(chuỗi gốc, vị trí đầu, chiều dài chuỗi con);

```
<?
$str="Hello World";
echo substr($str, 3,5); // lo wo
?>
```

c. Tìm kiếm chuỗi con

strpos(chuỗi gốc, chuỗi con); // Trả về chỉ số là nơi xuất hiện chu

```
<?
$str="Chao ban";
echo strpos($str,"a"); //2
?>
```

d. Tìm và thay thế

str_replace(<chuỗicầntìm>,<chuỗithaythế>,<chuỗigốc>);

```
<?
$str="Hello World";
$str=str_replace('o','e',$str);
echo str; //Helle Werld
?>
```

13. Array (mảng)

a. Giới thiệu:

Mảng là 1 đối tượng được tạo ra nhằm lưu trữ các dữ liệu có tính tương đồng. Trong mảng có nhiều phần tử với những giá trị khác nhau và được đánh theo số thứ tự từ 0 trở đi (chỉ số phần tử) hoặc được phân biệt theo dạng chuỗi (key)

name1="Teo";

name2="Ty";

name3="Map";

0	1	2
Teo	Ty	Map
a	b	c

Giá trị các key phải khác nhau.

b. Khai báo mảng

$TênMảng=array(); *// khai báo mảng rỗng*

$TênMảng=array(giá tri 1, giá trị 2,…); *//khai báo mảng có các phần tử đánh bằng chỉ số*

$ TênMảng =array("key1" =>giá trị 1, "key2" => giá trị 2, "key3" => giá trị 3); *//Khai báo mảng dùng key*

c. Thể hiện cấu trúc mảng:

Có thể dùng 2 hàm:

var_dump(TenMang); *// Liệt kê chi tiết bao nhiêu phần tử, index , kiểu và giá trị của từng phần tử*

print_r(TenMang); *// liệt kê index và giá trị từng phần tử. Không hiện kiểu và tổng số phần tử*

d. Ví dụ:

```
<?
$giohang=array(); //khai báo mảng rỗng
```

```
$diem=array(0,5,7);   // 3 phần tử chỉ số là 0,1,2.
$sanpham=array("idSP"=>5, "TenSP"=>"Gạo","Gia"=>12000); // đánh theo  key
echo "Cấu trúc mảng diem: <br/>";
var_dump($diem);
echo "<br>";
echo "Cấu trúc mảng sanpham: <br/>";
print_r($sanpham);
?>
```

e. Thêm phần tử vào mảng

```
<hr/>
<?
$diem[]=9;                    //Thêm 9 vào cuối mảng điểm, chỉ số cuối tăng 1 ➔ $diem=array(0,5,7,9)
$sanpham["SoLuong"]=3;        //thêm 3 làm phần tử cuối của mảng sanpham, key là SoLuong
print_r($diem);
echo "<br>";
print_r($sanpham);
?>
```

f. Truy xuất phần tử mảng

$ TênMảng[chiso] ; hoặc

$ TênMảng["key"] ;

```
<hr/>
<?      echo $diem[1]. "<br/>";          //5
        echo $sanpham["TenSP"];          //Gạo
?>
```

g. Đếm phần tử

count($TênMảng);

```
<hr/>
<? echo count($diem) ; //3 ?>
```

h. Liệt kê phần tử mảng dùng vòng lặp for :

Liệt kê mảng mà các phần tử được đánh theo chỉ số liên tục

```
<hr/>
<?
      $giatri=array(8,10,9) ;
      for ($i=0 ;$i<count($giatri) ;$i++) {
         echo "Giá trị thứ $i là $giatri[$i]<br>";
      }
?>
```

i. Dùng vòng lặp while :

Liệt kê các phần tử của mảng có chỉ số dạng key

```
<hr/>
<?
      $sanpham=array("idSP"=>5, "TenSP"=>"Gạo","Gia"=>12000, "SoLuong"=>3);
      reset($sanpham); // đưa con trỏ về phần tử đầu
      while (key($sanpham)!=null) {
         echo key($sanpham) . " =  " . current($sanpham) . "<br/>";
         next($sanpham); //, prev: lui trái, next: ptừ kế
      }
?>
```

```
<hr/>
<?
    $sanpham=array("idSP"=>5, "TenSP"=>"Gạo","Gia"=>12000, "SoLuong"=>3);
    reset($sanpham); // đưa con trỏ về phần tử đầu
?>
<table border=1>
<? while (key($sanpham)!=null) { ?>
<tr><td><? echo key($sanpham); ?> </td><td><? echo current($sanpham); ?> </td></tr>
<? next($sanpham); // prev: lui trái, next: ptừ kế ?>
<? } ?>
</table>
```

Chú ý:

- 1 phần tử của dãy có giá trị null cũng được, ghi vầy $a=array(10,null,7); Ghi vầy là sai: $a=array(10,,7);
- Phần tử có chỉ số thì key là null

14. Xử lý dữ liệu trên Form

a. Đối tượng $_GET, $_POST

- $_GET, $_POST là 2 mảng có sẵn trong PHP để chứa dữ liệu gửi lên từ form. Cả hai là mảng dùng key. Key là tên của các đối tượng trong form
- $_GET là đối tượng chứa dữ liệu gửi lên từ form dùng Method là Get
- $_POST là đối tượng chứa dữ liệu gửi lên từ form dùng Method là Post

b. Cách lấy giá trị từ form

- $_GET["Ten"]
- $_POST["Ten"]

c. Ví dụ 1:

- Tạo 1 file tên dangnhap.php rồi tạo form login trong tag body như dưới đây:

```
<form action="xulydangnhap.php" method="GET" name="form1" id="form1">
  <table width="286" border="0" align="center" cellpadding="4" cellspacing="0">
  <caption>ĐĂNG NHẬP</caption>
  <tr>
     <td width="114">Tên đăng nhập</td>
     <td width="150"><input name="u" type="text" class="txtfield" id="u" /> </td>
  </tr>
  <tr>
     <td width="114">Mật khẩu</td>
     <td><input name="p" type="password" class="txtfield" id="p"  /> </td>
  </tr>
  <tr>
    <td> </td>
    <td><input type="checkbox" name="nho" id="nho" />Nhớ user và pass</td>
  </tr>
  <tr>
     <td> </td>
     <td><input name="button" type="submit" class="btnSubmit" id="button" value="Đăng Nhập"
/></td>
  </tr>
  </table>
</form>
```

- Tạo file xulydangnhap.php rồi gõ mã sau vào tag body

```
<?
echo $_GET["u"]. "<br>";
echo $_GET["p"]. "<br>";
print_r($_GET )
?>
```

- F12 file dangnhap.php, gõ user, pass rồi nhấp Login. Nhìn URL của browser, bạn sẽ thấy file xulydangnhap.php và username, password.
- Chỉnh lại form trên với method là POST,
- Sửa phương thức GET trong file xulydangnhap.php thành POST và thử lại. Chú ý URL trên thanh Address khi nhấp nút Login

15. Đối tượng $_SERVER

Ngoài các thông tin trình duyệt gửi lên thông qua các con đường POST, GET, COOKIES, chúng ta cũng có thể thu được một số thông tin khác có liên quan đến trình duyệt cũng như các thông số liên quan đến máy chủ như: đường dẫn, địa chỉ IP, phiên bản... Các thông tin này được lưu trữ trong biến mảng $_SERVER (đối với các phiên bản mới) hoặc $HTTP_SERVER_VARS (đối với các phiên bản cũ hơn bản 4.1.0):

$_SERVER['HTTP_HOST'] / $_SERVER['SERVER_NAME']: Tên của máy chủ host, nơi mà đoạn script được thực thi.

$_SERVER['PHP_SELF']: Tên file nằm trên folder gốc của website. Ví dụ: biến $_SERVER['PHP_SELF'] trong script đặt tại địa chỉ http://example.com/test.php/foo.bar sẽ là /test.php/foo.bar.

$_SERVER['QUERY_STRING']: Chuỗi chứa tên các tham số và giá trị (sau dấu ?)

$_SERVER['DOCUMENT_ROOT']: Folder gốc của website, nơi mà file script đang được thực thi. Ví dụ: C:/AppServ/www

$_SERVER['HTTP_REFERER']: địa chỉ của trang web mà từ đó user link đến trang này.

$_SERVER['REMOTE_ADDR']: Địa chỉ của máy client, nơi người sử dụng đang duyệt web.

$_SERVER['REQUEST_URI']: Địa chỉ URI (Định danh đối với các tài nguyên mạng, là một dạng thức của URL).

$_SERVER['SERVER_SOFTWARE']: Chuỗi định danh của máy chủ, thường được cấp trong phần header khi trả lời các yêu cầu từ máy khách. Ví dụ: Apache/2.2.4 (Win32) PHP/5.2.3

$_SERVER['REQUEST_METHOD']: Loại yêu cầu mà clientsử dụng để truy cập trang web. VD: 'GET', 'HEAD', 'POST', 'PUT'.

$_SERVER['SCRIPT_FILENAME']: Đường dẫn tuyệt đối của file script đang chạy. Vd: C:/AppServ/www/laptrinhphp/abc.php

$_SERVER['SCRIPT_NAME']: Đường dẫn của file script. Vd /laptrinhphp/abc.php

```php
<?php
echo 'HTTP_HOST: '. $_SERVER['HTTP_HOST'].'<br>';
echo 'PHP_SELF: '.$_SERVER['PHP_SELF'].'<br>';
echo 'QUERY_STRING: '.$_SERVER['QUERY_STRING'].'<br>';
echo 'DOCUMENT_ROOT: '. $_SERVER['DOCUMENT_ROOT'].'<br>';
echo 'HTTP_REFERER: '. $_SERVER['HTTP_REFERER'].'<br>';
echo 'REMOTE_ADDR: '. $_SERVER['REMOTE_ADDR'].'<br>';
echo 'URL: '. $_SERVER['REQUEST_URI'] .'<br>';
echo 'SCRIPT_FILENAME: '. $_SERVER['SCRIPT_FILENAME'] .'<br>';
echo 'SCRIPT_NAME'. $_SERVER['SCRIPT_NAME'] .'<br>';
?>
```

16. Giới thiệu về http header

http header là các thông tin điều khiển nằm trong các http request , http response. Chứa các các chỉ thị báo cho trình duyệt/server biết cách xử lý thông tin. Ví dụ: trình duyệt chuyển qua trang khác, trình duyệt đừng hiện file php, trình duyệt đừng lưu trang vào cache

a. Redirection (chuyển hướng)

- Dùng để chuyển sang trang web khác.
- Cú pháp: **header("location:url");**
- Lệnh này phải dùng trên tất cả các mã lệnh html
- Ví dụ1:

```php
<?
    header("location:http://nhatnghe.com");
    exit();
?>
```

- Ví dụ 2: Nếu buổi sáng chuyển đến trang index1.html ngược lại chuyển đến index2.html

```php
<?
if (date('H')<=12) header('location:index1.html'); else header('location:index2.html');
?>
```

- Ví dụ 3: Nếu 2-4-6 chuyển đến trang index1.html ngược lại chuyển đến index2.html

```php
<?
if ((date('w')==1) || (date('w')==3) || (date('w')==5)) header('location:index1.html');
else header('location:index2.html');
?>
```

- Ví dụ 4: Nếu browser là IE thì chuyển đến trang index1.html, ngược lại trang index2.html

```php
<?
if (strpos($_SERVER['HTTP_USER_AGENT'],'MSIE')>0) header('location:index1.html');
else header('location:index2.html');
?>
```

- Ví dụ 5: Nếu browser là Firefox thì chuyển đến trang index1.html, ngược lại trang index2.html

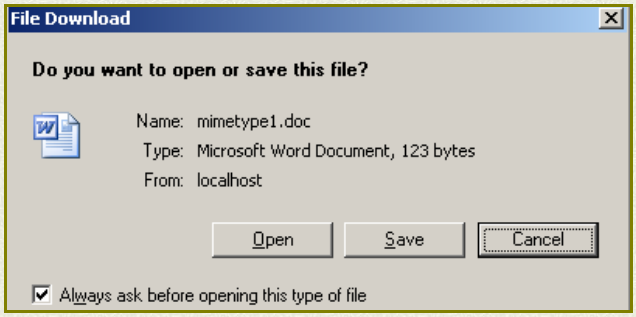
```php
<?
if (strpos($_SERVER['HTTP_USER_AGENT'],'Firefox')>0) header('location:index1.html');
else header('location:index2.html');
?>
```

b. Content-type

- Dùng để báo cho browser biết kiểu dữ liệu mà webserver đang trả về cho browser.
- Cú pháp: **header('Content-type:MimiType);**
- Ví dụ: tạo file mimetype1.html rồi gõ:

```
<?      header('Content-type: application/msword'); ?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head><title>Chào</title></head>
<body>Chào bạn</body>
</html>
```

- Nếu không có dòng đầu (chữ đỏ) thì browse sẽ hiện trang web có chữ chào bạn. Nếu có, broswe sẽ biết đó là 1 file word, nó sẽ hiện ra khung này



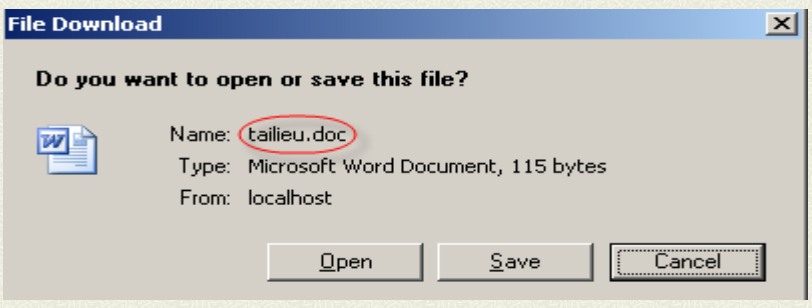
- Lệnh này phải dùng trên tất cả các mã lệnh html
- Một số kiểu Mimtype (xem thêm file mimetype.html)

application/msword	doc
application/pdf	pdf
application/vnd.ms-excel	xls
application/zip	zip

c. Content-disposition

- Cú pháp: **header('Content-disposition: attachment; filename="'.$filename.'");**
- Lệnh sẽ báo cho browser biết là nên download và save file. Tham số "filename=" sẽ báo cho browser biết tên của file nên được save.
- Ví dụ:

```
<?      header('Content-type: application/msword');
        $filename="tailieu.doc";
        header('Content-disposition: attachment; filename="'.$filename.'");
?>
<html xmlns="http://www.w3.org/1999/xhtml">
<head><title>Chào</title></head>
<body>Chào bạn</body>
</html>
```



17. Cookies

a. Giới thiệu:

- Cookie dùng để lưu những thông tin về user.
- Cookie được lưu trong máy client, trong vùng do browser quản lý
- Không dùng cookies để lưu những thông tin quan trọng vì không đảm bảo browser đều cho phép ghi cookies mà chỉ dùng cookie cho mục đích tiện ích mà thôi *(username, password, thời điểm login cuối, danh sách nhạc ưa thích)*

b. Tạo cookie

- setcookie("TenCookie",giá trị [,Thời điểm quá hạn]);
- Nếu không chỉ định thời gian thì cookie sẽ lưu trong bộ nhớ. Và sẽ mất khi user đóng browser.
- Nếu thời điểm quá hạn là 1 thời điểm trong quá khứ thì browser sẽ xóa cookie

c. Sử dụng cookie

- $_COOKIE["Ten"];

d. Ví dụ 1:

- Tạo file setcookies.php và gõ khối lệnh sau (phải trên tất cả mã html)

```
<?php
    setcookie("un", "Nguyen Van Teo", time()+60*60*24*30);
    setcookie("lasttime", time(), time()+60*60*24*30);
?>
```

- Chú ý: Hàm time() cho biết số giây tính từ đầu năm 1970 đến thời điểm hiện tại
- F12 xem thử, bạn sẽ chẳng thấy gì, nhưng 2 cookie đã được tạo (có thể xem thấy trong cache)
- Tạo file viewcookie.php rồi gõ khối lệnh sau trong tag body

```
<?
  if (isset($_COOKIE["un"])==true)   echo "Chào bạn  " . $_COOKIE["un"] ."<br>";
  else   echo "Chào quý khách";
  echo "Bạn vào site lần cuối lúc " . date("d/m/Y H:i:s",$_COOKIE["lasttime"]);
  echo "<hr>";
  print_r($_COOKIE);   // Xem mọi cookie
?>
```

- "F17" xem thử, bạn thấy giá trị 2 cookie

e. Ví dụ 2

- Tạo file xoacookie.php rồi vào dòng đầu tiên:

```
<?
    setcookie("un", "haha", time()-1);
?>
```

- F12 , cookie sẽ bị xóa. Bạn có biết kiểm tra bằng cách nào không?

f. Ví dụ 3:

- Mở file xulydangnhap.php, gõ thêm ở đầu:

```
<?
    if (isset($_POST['nho'])==true){
            setcookie("iu_sơ_nêm",$_POST['u'],  time()+60*60*24*7 );
            setcookie("bát_quợt",$_POST['p'],  time()+60*60*24*7 );
    } else {
```

```
            setcookie("iu_sơ_nêm",$_POST['u'],time()-1);
            setcookie("bát_quợt",$_POST['p'],time()-1);
    }
?>
```

- Lưu file lại
- Mở file 1 file dangnhap.php , chọn tag form rồi kểm tra xem có phải method là POST không
- Nhắp textfield tên u rồi qua code, bạn sẽ thấy đoạn mã:

```
<input name="u" type="text" class="txtfield" id="u" />
```

- Gõ thêm vào (màu đỏ) để có kết quả sau:

```
<input value="<? echo $_COOKIE['iu_sơ_nêm'];?>" name="u" type="text" class="txtfield" id="u" />
```

- Qua Design, nhắp textfield tên p rồi qua code, bạn sẽ thấy đoạn mã:

```
<input name="p" id="p" type="password" class="txtfield" />
```

- Gõ thêm vào (màu đỏ) để có kết quả sau:

```
<input value="<? echo $_COOKIE['bát_quợt'];?>" name="p" id="p" type="password" class="txtfield" />
```

- "F17" trang dangnhap.php. Gõ username và password rồi nhắp Đăng nhập, bạn sẽ thấy user name password. Xem xong đóng browser lại
- "F17" trang dangnhap.php. Gõ username và password , check vào Nhớ username rồi nhắp Đăng nhập, bạn sẽ thấy user name password. Lúc này username và password đã được lưu lại trong cookie. Xem xong đóng browser lại
- "F17" trang dangnhap.php. Bạn sẽ thấy username, password đã được hiện sẵn

18. Session

a. Giới thiệu:

- Là đối tượng chứa thông tin user trên server.
- Mỗi user có 1 session riêng.
- Cấu trúc session của mỗi user là giống nhau (chỉ khác gia trị các biến).
- $_SESSION là dãy toàn cục có sẵn trong php, dùng để chứa các biến session. Do đó dữ liệu trong session có thể được truy xuất từ mọi trang php trong site.

b. Khai báo và sử dụng

- $_SESSION["TênBiến"]
- Trang php nào có dùng session thì phải có hàm < ? session_start() ;?> Hàm này phải nằm trên tất cả mã lệnh html
- Vd:

```
<?
$_SESSION["login"]=1 ; //tạo biến, mọi trang đều đọc được
?>
```

c. Ví dụ 1:

```
<?
```

```
if(isset($_SESSION['views'])==true)
  $_SESSION['views']=$_SESSION['views']+1;
else
  $_SESSION['views']=1;
echo "Views=". $_SESSION['views'];
if ($_SESSION['views']>=10) unset($_SESSION['views']);  //session_destroy();
?>
```

d. Ví dụ 2: Ghi nhận số lượng sản phâm được chọn vào giỏ hàng

- **Mở trang index_banhang.php**
  - Qua code, lên dòng đầu gõ lệnh

```
<? session_start(); ?>
```

  - Qua design. Ở cột phải, trong dòng Giỏ hàng có 0 sản phẩm, chọn số 0 xóa rồi qua code gõ code (màu đỏ)

```
Giỏ hàng có <? echo $_SESSION['SoSP']; ?> sản phẩm
```

  - Kiểm tra xem từng nút đặt mua có link đến trang luuugiohang.php hay không
- **Tạo trang luugiohang.php**
  - Qua code, xóa hết code html và gõ lệnh sau

```
<?
session_start();
$_SESSION['SoSP']+=1;
header("location:".$_SERVER['HTTP_REFERER']);
?>
```

- **Test thử**

e. Ví dụ 3 :

- Tạo file index_admin.php – Trang chủ

```
<?
session_start();
if (isset($_SESSION['login'])==false) {
  header("Location: login.html");
  exit();
}
?>
<body>
Đây là trang chủ quản trị.<br/>
Chào <b><?=$_SESSION["username"]?></b>. Bạn đã login thành cong.
<hr> <a href=signout.php>Logout</a>;
</body>
```

- Tạo file login.html – Form login

```
<form action="signin.php" method="POST">
Username: <INPUT TYPE="text" name="username"><br>
Password: <INPUT TYPE="password" name="password"><br>
<INPUT TYPE="submit" name="submit" value="Login">
</form>
```

- Tạo file signin.php – Kiểm tra thông tin

```
<?
    session_start();
    if (checkLogin($_POST['username'], $_POST['password'])==true){
        $_SESSION["username"]=$_POST['username'];
        $_SESSION["login"]=true;
        header('Location: index_admin.php');
        exit();
    }
    else {
        header('Location: login.html');
        exit();
    }

    function checkLogin($theUsername, $thePassword){
            if (($theUsername=="teo") && ($thePassword=="123")) return true;
            else return false;
    }
?>
```

- Tạo file signout.php

```
<?
    session_start();
    session_destroy();
    header('Location: login.html');
?>
```

LẬP TRÌNH WEB ĐỘNG VỚI

PHP / MySQL

- GUESTBOOK
- CATALOG
- FORUM
- SHOPPING CART

PHẦN 1

Tống Phước Khải (tổng hợp& biên dịch)

PDF processed with CutePDF evaluation edition www.CutePDF.com

Giới thiệu

Chúng ta hãy thực hiện một chuyến đi thần thoại, trong chuyến đi này chúng ta sẽ khám phá ngoại hình cũng như nội tại của MySQL và PHP một cách thật tỉ mỉ. Đây là một cuộc hành trình đầy những thú vị và bất ngờ.

Okie, có lẽ tôi có vẻ hơi lạc quan phải không các bạn. Nếu như bạn đồng quan điểm với tôi trên một phương diện nào đó, trong cuộc hành trình này bạn sẽ có ngay sự giúp đỡ mỗi khi gặp phải những sự nhàm chán. Hãy đối mặt sự thật ngay nhé: Trò chơi lập trình ứng dụng không phải lúc nào cũng dễ nuốt đâu. Trong bất kỳ cuộc thám hiểm nào thì chắc chắn các bạn sẽ phải có những giây phút nản lòng, đó là lúc gặp phải sự cố lỗi cú pháp hoặc đôi khi là những đoạn mã không cho kết quả như mong muốn. Nhưng ngoài những việc đó ra, tôi nghĩ là có một lý do thật chính đáng đến các bạn đến với chúng tôi ở đây. Lập trình Web đang là một cuộc chơi đầy hứa hẹn hiện nay cũng như tương lai. Bất kể bạn có kiến thức cơ sở lập trình cho bất kỳ loại ngôn ngữ nào như Visual Basic, Cobol, hay bạn chỉ biết về HTML và JavaScript, thì hôm nay bạn vẫn có cơ hội để nắm bắt các kinh nghiệm mới mẻ về lập trình ứng dụng Web. Tôi nghĩ là không có sự kết hợp nào tốt hơn giữa PHP và MySQL. Số lượng người sử

dụng ngôn ngữ này càng gia tăng, PHP và MySQL đã trở thành rất thông dụng, những đòi hỏi lượng người biết các công cụ lập trình này cũng tăng theo. Một chút xíu nữa tôi sẽ nói rõ cho bạn biết tại sao lại phải sử dụng PHP và MySQL. Nhưng trước hết tôi muốn bạn hãy khảo sát qua kiến trúc sơ bộ của ứng dụng Web. Vì chỉ khi bạn nắm bắt được điều này thì tôi mới có thể tiếp tục trình bày chi tiết rằng tại sao PHP và MySQL là trung tâm của môi trường phát triển ứng dụng Web.

Trước khi tiếp tục, tôi nghĩ rằng bạn đã đọc những gì tôi đã giới thiệu và hiểu nó. Chúng ta tiếp tục đi thôi!

Kiến trúc cơ bản

Kiến trúc căn bản nhất để trang Dynamic Web hoạt động được là nó phải làm việc trên mô hình client/server. Nôm na là mỗi thứ client hay server đều đảm đương một chức năng riêng để hoàn thành công việc chung đó là cho ra một trang Web động. Các bạn có lẽ đã quen thuộc với chương trình WinWord để soạn văn bản, nó có thể hoạt động độc lập trên bất kỳ máy tính nào chẳng cần quan tâm tới cái gì là client hay cái gì là server. Ứng dụng Web thì khác hẳn, phải có một mô hình server có thể là

một máy tính làm server thôi, nhằm tập trung hoá việc xử lý dữ liệu. Còn các client, còn được hiểu nôm na là máy tính của người sử dụng phải được nối mạng với server, giả sử các máy này truy cập vào một Website chẳng hạn, thì có nghĩa họ đã truy cập vào server, sau đó lấy dữ liệu từ server về thể hiện lên máy mình. Cùng một lúc có thể có hàng trăm người (client) truy cập vào cùng một Website được xử lý tập trung trên server, tương tự như một đám trẻ xúm nhau giành phần của mình từ một cái bánh.

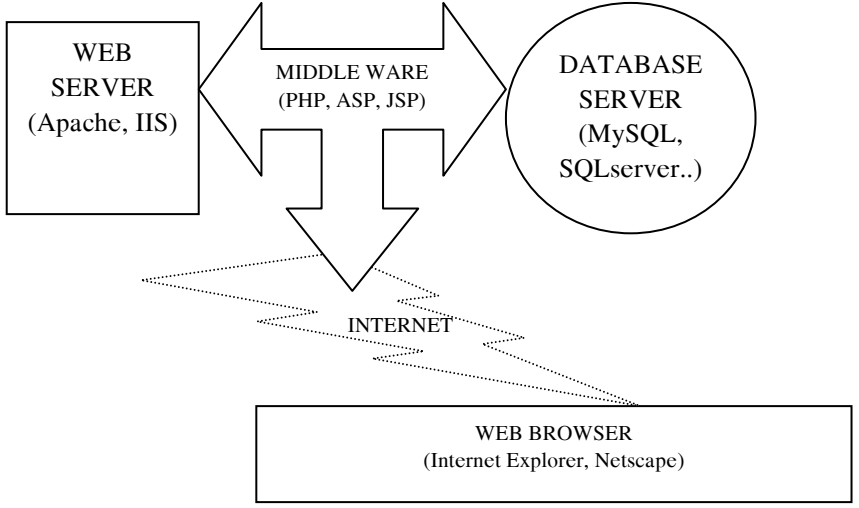
Client (người Việt tạm đọc là klai-ờn)

Các ứng dụng mà bạn phát triển trên nền MySQL và PHP sử dụng tính năng single client đó là trình duyệt Web. Tuy nhiên, không phải đây chỉ là ngôn ngữ duy nhất để phát triển ứng dụng Web. Đối với những ứng dụng phức tạp đòi hỏi multi-client hoặc cần các tính năng bảo trì (chúng ta sẽ bàn tính năng này sau), thì ứng dụng Java applet sẽ hữu dụng cho việc này. Chỉ trừ trường hợp bạn cần sử dụng ứng dụng thời gian thực như ứng dụng chat chẳng hạn, thì bạn Java Applet mới cần thiết. Ở đây chúng ta không bàn tới lập ứng dụng cho chuyện tán gẫu mà chỉ tập trung vào ứng dụng duyệt Web nên không đụng chạm gì tới Java Applet cả.

Như bạn đã biết ngôn ngữ khởi thuỷ cho việc duyệt Web là HTML. HTML cung cấp hàng tá những thẻ lệnh (Tag) cho phép thể hiện trang Web theo nhiều kiểu cách khác nhau. Nếu bạn chưa có kiến thức cơ sở về HTML thì có thể chạy ra ngoài mua ngay một quyển sách hoặc download trên internet xuống các bài học hướng dẫn. Bạn không nên bỏ ra quá nhiều thời gian để học về HTML. Ngoài HTML ra các trình duyệt Web còn cho phép các add-in hỗ trợ nhiều thứ khác như RealPlayer, Flash, Shockwave, hoặc hỗ trợ về Javascript hoặc XML. Nhưng ở đây chúng tôi chỉ tập trung trên những gì cần thiết cho sự hội nhập của bạn – đó là HTML.

Server (người Việt tạm đọc là sơ-vơ)

Hầu hết các ứng dựng Web đều hoạt động tập trung trên Server. Một ứng dụng đặc trưng gọi là Web Server sẽ đảm trách việc giao tiếp với các trình duyệt. Một Cơ sở dữ liệu (CSDL) trên Server sẽ lưu trữ tất cả những thông tin đáp ứng yêu cầu cho công việc của ứng dụng Web. Kế tiếp, bạn cần phải có một ngôn ngữ làm vai trò **"chú bé liên lạc"** giữa **Web Server** và **CSDL** trên server. Ngôn ngữ này cũng thực hiện các công việc xử lý thông tin đến và đi từ Web Server.



Và dĩ nhiên là các thứ này sẽ chẳng hoạt động được nếu như không chạy trên một Hệ Điều Hành (HĐH). Các thứ như Web Server, Ngôn ngữ lập trình, CSDL phải hoạt động tốt trên một HĐH nào đó.

Hệ điều hành

Có rất nhiều chủng loại HĐH. Windows 98/XP và Linux có lẽ rất phổ biến với tất cả mọi người. Có trường hợp bạn làm việc trên HĐH mà ít ai biết tới và bạn chỉ có ấn tượng thích sử dụng nó mà thôi. Hãy gác qua những ý tưởng đó nếu như bạn thật sự muốn đi trên con đường thiết kế web. Hãy trang bị cho mình kiến thức về HĐH WinNT /2000 /2003 và Unix đi. Việc làm này sẽ rất có ích hơn là chuyện bảo mọi người nên đi học một khoá về AS/400.

Bạn sẽ sử dụng loại nào trong các thứ nói trên đây? Okie, đây là một câu hỏi hơn rắc rối đấy. Câu trả lời ở đây là tuỳ thuộc bạn là "tín đồ" của HĐH nào. Nếu như bạn vẫn chưa rõ ràng về điều này, hãy để tôi nói cho bạn nghe về "chiến tranh giáo phái HĐH".

Nếu bạn chưa hiểu được tôi đang nói gì, thì đây là các kiến thức cơ bản: PHP và MySQL thuộc nhóm phần mềm ứng dụng có tên gọi là **open source** (nguồn mở). Việc này có nghĩa là người dùng sẽ xem được mã nguồn của các ứng dụng sử dụng PHP/MySQL. Chúng tận dụng được mô hình phát triển dựa vào nguồn mở, cho phép người nào cảm thấy thích nó đều có thể góp phần vào việc phát triển các dự án.

Trong trường hợp của PHP, các lập trình viên trên toàn thế giới tham gia vào việc phát triển ngôn ngữ và không trông chờ một khoản lợi nhuận nào. Phần lớn những người tham gia công việc đều có niềm đam mê việc tạo ra một sản phẩm phần mềm tốt, họ sẽ cảm thấy thích thú khi thấy người khác sử dụng các công cụ của họ như tôi và bạn chẳng hạn.

Phương pháp nguồn mở này ban đầu chỉ còn là những vòng lẩn quẩn mà thôi, nhưng về sau đã trở thành đầy tiềm lực khi có sự ra đời và trở nên phổ biến của bộ nguồn mở Linux. Hầu như các nguồn mở đều miễn phí, bạn có thể download, cài đặt và sử dụng chúng mà không cần phải đợi sự cho phép hay phải trả tiền cho bất kỳ ai. Phương thức này thì Microsoft, Oracle hay một số các công ty lập trình nào khác không thể đáp ứng được.

Nếu bạn không phải là tín đồ của phái nguồn mở, thì hãy chọn công cụ được coi là béo bở: NT/2000/2003. Nếu công ty của bạn đã sử dụng sản phẩm của Microsoft nhiều năm rồi thì mọi việc sẽ trở nên dễ dàng nếu bạn muốn duy trì làm việc với môi trường này. Nếu bạn là thành viên của nhóm lập trình Visual Basic, có lẽ bạn sẽ gắn bó với NT/2000/2003. Ngay cả trong trường hợp này, không có trở lực nào ngăn cản

bạn trong công việc phát triển với công cụ PHP và MySQL. Bạn cũng có thể thử nghiệm **PHP/MySQL** trên nền HĐH **Windows 95, 98, XP**.

Web Server

Chức năng của Web Server có vẻ không phức tạp mấy. Nó chỉ ở tại chỗ, chạy trên nền của HĐH, lắng nghe các yêu cầu ai đó trên Web gởi đến, sau đó trả lời những yêu cầu này, và cấp phát những trang Web thích ứng. Thực tế thì nó không quá đơn giản như vậy, bởi vì nhiệm vụ của Web Server là phải cung cấp tính ổn định cho môi trường Web cho nên đòi hỏi này phải được đáp ứng một cách rất nghiêm túc.
Có nhiều loại Web Server khác nhau, nhưng chủ yếu trên thị trường chỉ thường sử dụng Apache và IIS (Internet Information Server của Microsoft).

INTERNET INFORMATION SERVER (IIS) được gắn liền với môi trường Windows và nó là thành phần không thể thiếu của Active Server Pages (ASP). Nếu bạn chọn con đường của Microsoft thì có lẽ bạn đã hiểu rõ về IIS.
Có một sự tích hợp nhất định giữa một ngôn ngữ lập trình và một Web Server. Cũng vậy, PHP4 được tích hợp rất tối đối với IIS. Trước đây, có một số vấn đề cần phải bàn

về tính ổn định của PHP/IIS với việc truyền tải lớn, nhưng PHP và IIS cũng đã được cải thiện liên tục nên việc này không còn đáng phải bận tâm.

APACHE là một kiểu mẫu Web Server rất phổ biến. Giống như Linux, PHP, MySQL nó là một dự án nguồn mở. Không có gì ngạc nhiên khi người ta thấy Apache được hỗ trợ **rất tốt trên môi trường Unix**, nhưng chỉ **khá tốt trong Windows**.
Apache tận dụng được tính năng của third-party. Bởi vì đây là nguồn mở nên bất kỳ ai có khả năng đều có thể viết chương trình mở rộng tính năng của Apache. PHP hoạt động với tư cách là một phần mở rộng của Apache, và người ta gọi là một module của Apache.
Apache có tính ổn định và tốc độ đáng phải nói. Tuy nhiên, cũng có một số sự phàn nàn về nó là không hỗ trợ công cụ đồ hoạ trực quan, điều có thể giúp người ta làm việc một cách dễ dàng hơn. Bạn phải thực hiện các thay đổi đối với Apache bằng cách sử dụng dòng lệnh, hoặc sử các tập tin text trong folder chương trình Apache. Nếu lần đầu đến với Apache thì bạn sẽ gặp một chút lạ lẫm.
Mặc dù Apache chỉ làm việc tốt trên Unix, nhưng cũng có những phiên bản chạy tốt trên hệ Windows. Không một ai, kể cả các nhà phát triển Apache đề nghị rằng

Apache nên được chạy trên một server Windows bận rộn. Nếu bạn quyết định chọn HĐH Windows cho server thì bạn nên sử dụng IIS. Nếu bạn thử nghiệm ứng dụng trên Windows và sau đó đem upload và chạy trên Unix/Apache của nhà cung cấp host thì cũng không hề hấn gì, ứng dụng của bạn vẫn chạy ngon lành.

Middleware

PHP thuộc lớp ngôn ngữ lập trình gọi là middleware. Các ngôn ngữ này hoạt động cận kề với Web Server để thông dịch các yêu cầu từ trên World Wide Web, sau đó nhận các trả lời từ Web Server chuyển tải đến trình duyệt Web nhằm đáp ứng các yêu cầu đó.

Middleware là nơi mà bạn sẽ thực hiện các khối lượng rất lớn công việc chính yếu của bạn. Với hỗ trợ này Web Server của bạn sẽ không phải cán đáng quá nhiều khối lượng công việc. Nhưng khi bạn phát triển ứng dụng của bạn, bạn sẽ tốn nhiều thời gian viết mã chương trình để cho chương trình của bạn có thể hoạt động được. Ngoài PHP ra có một số ngôn ngữ khác có chức năng tương đương như ASP, Perl, ColdFusion.

Hệ CSDL quan hệ

Relational Database Management Systems (Hệ Quản trị Cơ Sở Dữ Liệu Quan hệ - RDBMSs) cung cấp phương thức tuyệt vời để lưu trữ và truy xuất lượng thông tin lớn và phức tạp. Nó đã ra đời khá lâu. Thực tế, nó có trước Web, Linux và WindowsNT, cho nên không có gì ngạc nhiên khi có quá nhiều hệ CSDL để chọn lựa. Tất cả các CSDL này đề dựa trên cơ sở SQL (Structure Query Language).

Một số hệ phổ biến như **Oracle, Sysbase, Informix, Ms SQL Server, IBM's DB2**. Hệ nguồn mở thông dụng hiện nay là **MySQL** mà quyển sách này đề cập đến, ngoài ra còn có hai hệ nguồn mở khác là **PostgresSQL** đã một thời thay thế MySQL và Interbase là bộ nguồn mở của Borland giới thiệu vào tháng 8/1999.

Tại sao sử dụng PHP và MySQL

Tại sao có quá nhiều chọn lựa như vậy mà chúng ta lại phải chỉ lấy ra cặp bài trùng PHP/MySQL mà thôi? Tôi sẽ giải thích điều nay ở phần sau.

Nói về PHP

Các ngôn ngữ lập trình xem ra giống như các loại giày dép. Có loại có vẻ bắt mắt với một số người này, nhưng lại khó ưa với người khác và ngược lại. Một số người chỉ thích sử dụng một hiệu giày nào đó đã quen thuộc và ngôn ngữ lập trình cũng tương tự như vậy.

Ở đây tôi muốn ngụ ý với các bạn là khi lập trình Web, các ngôn ngữ lập trình đều cho kết quả gần giống nhau. Câu hỏi ngôn ngữ nào tốt nhất không phải là vấn đề nó không có khả năng thực hiện một số chức năng nào đó mà thường là nó có làm cho bạn thực hiện công việc một cách nhanh chóng và đỡ nhọc công hay không?

Tốc độ nhanh, dễ sử dụng

Chúng ta hãy bàn về tốc độ. Có 3 thứ mà tôi chắc chắn khi bàn về việc so sánh tốc độ giữa các ngôn ngữ lập trình Web. Thứ nhất, ứng dụng viết bằng C chạy nhanh nhất. Thứ hai, công việc lập trình C khá là phức tạp, và sẽ ngốn nhiều thời gian hơn. Thứ ba, việc so sánh giữa các ngôn ngữ là một điều khó khăn. Tất cả những gì tôi biết là tôi cảm thấy yên tâm khi nói rằng PHP cũng nhanh như các ngôn ngữ khác. Trở lại ví

dụ so sánh với các loại giày dép: Vina, Đông Hải, Kiến Hoa, Hồng Thạnh, Italy v.v., chắn chắn bạn sẽ chọn loại tiện dụng nhất? Nếu bạn giống như tôi, bạn sẽ cảm thấy rằng PHP có đầy đủ các đặc tính như khả năng, cấu trúc và dễ sử dụng. Xin nói thêm, đây chỉ là cách nhìn riêng của tôi thì tôi tin rằng cú pháp PHP tuyệt hơn ASP hay JSP. Và theo tôi thì việc gõ lệnh PHP nhanh hơn ColdFusion và nó không khó học như Perl. Tóm lại, tôi cho rằng PHP cung cấp các tính năng mạnh mẽ để thực hiện ứng dụng Web một cách nhanh chóng.

Chạy trên nhiều hệ điều hành

Như đã trình bày ở phần kiến trúc web, tôi có nói là PHP có thể chạy trên WindowsNT/2000/2003 và Unix với sự hỗ trợ của IIS và Apache. Nhưng ngoài ra nó có thể chạy trên một số các platform khác như Netscape, Roxen, hay một vài thứ khác. Như chúng ta biết ASP có thể chạy trên Unix, ColdFusion có thể chạy trên Solaris và Linux, JSP có thể chạy trên khá nhiều loại platform. Đối với PHP, nó có thể chạy tốt trên những platform hỗ trợ các chủng loại trên.

Truy cập bất kỳ loại CSDL nào

Ứng dụng của bạn dự định sẽ truy cập những loại dữ liệu dịch vụ nào? LDAP, IMAP mail server, DB2, hay XML parser hay WDDX.

Bất kể bạn cần đến thứ gì thì PHP cũng sẵn sàng hỗ trợ thông qua các hàm được xây dựng sẵn nó sẽ làm công việc của bạn trở nên rất dễ dàng và tiện lợi. Nhưng nếu như có một số thứ chưa được xây dựng sẵn thì sao? Ta tiếp tục sang phần sau sẽ rõ.

Luôn được cải tiến & cập nhật

Nếu như bạn cảm thấy bỡ ngỡ đối với việc phát triển nguồn mở, bạn có lẽ sẽ ngạc nhiên đối với chất lượng của loại phần mềm này. Có hàng ngàn những chuyên gia lập trình xuất sắc đợi sẵn và họ sẵn sàng bỏ thời gian ra để tạo những phần mềm tuyệt vời và hầu như miễn phí. Đối với ngôn ngữ thịnh hành như PHP thì ắt hẳn là các rất nhiều các nhà lập trình đang thực hiện phát triển nó hằng ngày.

Sự thật có một việc rất ấn tượng là nếu như bạn có một sự cố kỹ thuật, bạn có thể gởi email đến một nhà phát triển PHP các chi tiết sự cố đó. Chỉ trong vòng vài giờ bạn sẽ nhận được sự trả lời thoả đáng.
Khi PHP4 được phổ biến, nó đã trở thành một hiện tượng của ngôn ngữ lập trình. Nó giúp cho việc bổ sung số lượng lớn các hàm chức năng một cách dễ dàng. Nếu như ngôn ngữ đã có sẵn nhiều hàm đặc thù cho công việc thì bạn sẽ đỡ tốn công hơn cho việc lập trình của mình.

Được hướng dẫn kỹ thuật bất cứ lúc nào

Hầu hết các ngôn ngữ đều hỗ trợ active mailing list (hiểu nôm na là danh sách mail những thành viên trực chiến hỗ trợ kỹ thuật) và các development site (trang web hỗ trợ giải quyết kỹ thuật). PHP cũng không ngoại lệ. Nếu bạn gặp phải sự cố - gặp những lỗi trong chương trình và không tìm ra cách khắc phục - sẽ có hàng trăm người có tên trong danh sách mail luôn sẵn lòng kiểm tra và khắc phục sự cố cho bạn.
Bộ nguồn mở PHP thật sự đã tạo ra một tình cảm của cả cộng đồng. Khi bạn gặp phải khó khăn đối với nó thì lúc nào cũng có những đồng môn chia sẻ nỗi lòng đó và giúp bạn khắc phục nhằm đem lại niềm vui cho bạn.

Hoàn toàn miễn phi$

Bạn không ngại gì về vấn đề bản quyền khi bạn sắm một máy vi tính và cài lên đó những phần mềm như Linux, Apache, PHP vì tất cả đều miễn phí.

Nói về MySQL

Mặc dù MySQL được phổ biến rất nhiều nhưng nó vẫn có những đối thủ đáng gờm đang cạnh tranh với nó. Những đối thủ của nó có thể trội hơn về một phương diện đặc thù nào đó.

Trong phần trên, chúng ta đã bàn sơ qua MySQL. Trong phần này, bạn sẽ được biết về những đặc điểm của những Hệ quản trị CSDL khác mà MySQL không hỗ trợ.

Với những hạn chế đó đã làm cho MySQL không được chọn để chạy trên một số các môi trường. Nếu bạn đang có kế hoạch bắt đầu cho một ngân hàng chẳng hạn, thì tôi khuyên bạn là MySQL không thích hợp cho bạn.

Nhưng đối với phần đông mọi người và phần lớn các ứng dụng, MySQL là sự chọn lựa của họ bởi nó rất thích hợp cho những ứng dụng Web.

Vừa túi tiền

Hãy nghĩ bạn cần cài đặt Oracle. Hãy chuẩn bị hầu bao của mình khoảng 30.000 đến 100.000 USD hoặc thậm chí còn hơn thế nữa. Điều hiển nhiên là Oracle, Sysbase và

Informix là những Hệ Quản trị CSDL tuyệt vời, nhưng giá thành quá cao, không hợp với túi tiền của phần đông mọi người.
MySQL hoàn toàn miễn phí. Bạn có thể sử dụng mà không cần chuẩn bị bất kỳ khoản tiền nào.

Nhanh và mạnh

MySQL không có đầy đủ những cơ sở vật chất cho một Hệ Quản trị CSDL chính tông, nhưng đối với công việc thường nhật của phần đông mọi người thì nó cung cấp cũng khá nhiều thứ. Nếu công việc của bạn là lưu trữ dữ liệu trên Web hoặc làm một trang Thương mại Điện tử cỡ vừa, thì MySQL có đủ những thứ bạn cần.
Đối với những CSDL cỡ trung bình thì MySQL hỗ trợ tuyệt vời về tốc độ. Các nhà phát triển MySQL rất tự hào về tốc độ sản phẩm của họ. Với các ứng dụng mà tôi giới thiệu trong phần III và IV của quyển sách này, thì bạn khó có thể kiếm được một Hệ Quản trị CSDL nào đạt được tốc độ nhanh hơn nó.

Cải tiến liên tục

MySQL được cải thiện liên tục với một tần số không ngờ. Các nhà phát triển cập nhật nó thường xuyên, ngoài ra còn bổ sung các tính năng rất ấn tượng cho nó mọi lúc mọi nơi.

Hiện tại, MySQL đã được bổ sung thêm hỗ trợ transaction. Như vậy là MySQL đã thực thụ trở thành một Hệ Quản trị CSDL chuyên nghiệp.

Thực hành ứng dụng đầu tiên

Phần mở đầu như vậy là tạm đủ. Bây giờ chúng ta hãy tiếp tục sang phần viết một ứng dụng thử nghiệm để biết được cách thức hoạt động của ngôn ngữ này như thế nào. Có lẽ đọc qua phần giới thiệu bạn cũng đã có một số khái niệm nhất định về sự hoạt động của chúng.

Công cụ cần thiết

Có một số thành phần cần thiết mà bạn phải có trước hết. Tôi sẽ giới thiệu ngay sau đây và bạn sẽ biết mình cần đến những gì.

PHP Webserver

Đây là ứng dụng chạy trên Web, cho nên bạn điều hiển nhiên là bạn cần phải có một Web Server. Bạn sử dụng Apache, bạn cài lên Winserver2000/ 2003 hoặc 98, XP thông dụng của bạn. Có một số phiên bản Apache có sẵn bộ cài đặt PHP trong đó.

Nếu chưa có bạn phải cài đặt thêm PHP. Còn nữa, bạn phải cài MySQL. Như vậy bộ ba **Apache, PHP và MySQL** luôn đồng hành với nhau.
Bạn xem thêm phần cài Apache server trên các CD thực hành PHP hoặc xem trên các Diễn đàn Tin học . Sau khi cài đặt xong bạn khởi động Apache. Nếu từ trình duyệt gõ vào http://localhost trang web thông tin của Apache hiển thị thì coi như thành công.
Lưu ý: Bạn cần phải xác định thư mục gốc của localhost để chứa các file **.php** của bạn sau này (xem trong hướng dẫn cài đặt Apache).

Text Editor

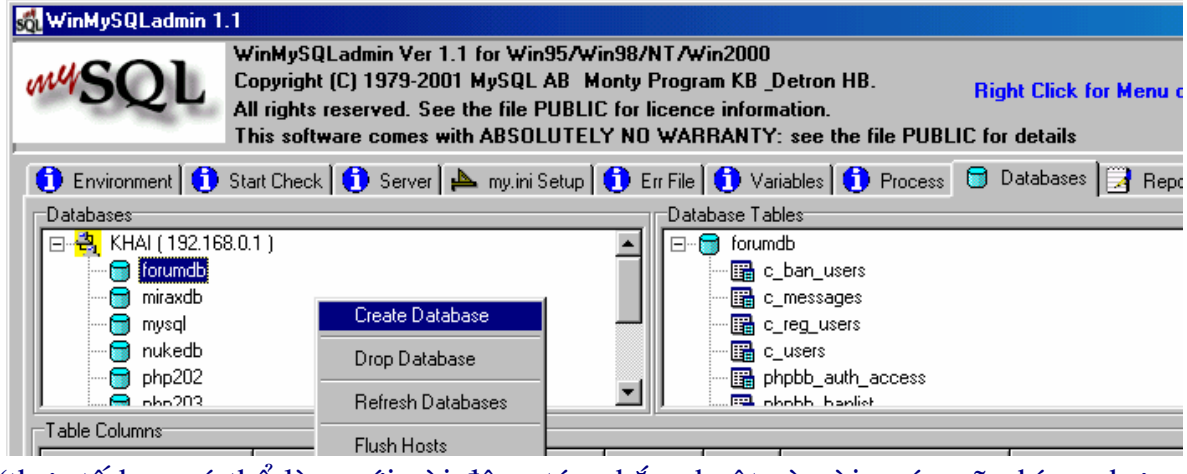
Để soạn thảo các dòng lệnh PHP bạn cần có một chương trình soạn thảo text đơn giản thôi, như Notepad trong Windows chẳng hạn.
Có một số chương trình soạn thảo hỗ trợ PHP, các dòng lệnh được ngời sáng bằng nhiều màu khác nhau nhằm làm cho việc lập trình của bạn dễ dàng hơn. Bạn có thể vào các website của Allairre (www.allaire.com) hoặc Editplus (www.editplus.com). Hiện nay, chương trình Dreamweaver MX của Macromedia cũng hỗ trợ soạn thảo mã lệnh PHP rất tốt.

BẮT ĐẦU LÀM

Tôi nghĩ là tôi đã khởi đầu quyển sách với những mớ lý thuyết suông. Bây giờ chúng ta hãy bắt tay vào thực đi thôi. Như bạn đã biết khi truy cập vào một trang Web có thể bạn sẽ được yêu cầu hay chính bạn muốn ghi lại ý kiến cùng với một mớ các thông tin nhận dạng về mình như họ tên, địa chỉ website, email v.v. Tất cả các thông tin này sẽ được lưu trữ vào một CSDL trên Web. Nhờ vậy, người quản trị Web hoặc những người truy cập khác sẽ biết thông tin cá nhân cùng những ý kiến của bạn. Người ta gọi thông tin này là GuestBook (hiểu nôm na là Sổ vàng để khách viếng thăm ghi chép). Bây giờ chúng ta bắt tay vào việc tạo một GuestBook.

Tạo một Database (quan trọng!)

Bây giờ bạn cần biết phải làm gì rồi. Chuyện đầu tiên là phải tạo một CSDL lưu trữ thông tin của khách. Để làm được điều này bạn cần phải dùng đến ngôn ngữ SQL



(thực tế bạn có thể làm với vài động tác nhắp chuột và vài ngón gõ phím, nhưng hãy tập làm quen với SQL vì nó sẽ hữu dụng về sau). Bạn sẽ được học kỹ về SQL trong các chương sau. Do đó bạn đừng lo lắng khi chưa hiểu gì về nó.

Bây giờ bạn hãy khởi động MySQL. Nếu bạn đã cài đặt MySQL trong Windows thì nó sẽ có biểu tượng để khởi động hoặc nó sẽ được tự động khởi động khi mở Windows lên. Đối với MySQLAdmin version 1.1 cho phép bạn làm việc trong 2 giao diện: Windows và Dos. Đối với giao diện Widows thì biểu tượng MySQL (biểu tượng đèn giao thông) nằm ở SystemTray, bạn chỉ việc click chuột phải lên nó và chọn **Show me**. Cửa sổ làm việc của MySQL hiện lên, tuy nhiên trong cửa sổ này chỉ cho phép bạn thực hiện một số thao tác có hạn đối với CSDL. Hình trên là cách tạo Database mới trong MySQL theo giao diện Windows.

Tuy nhiên, tôi khuyên các bạn nên dùng tiện ích **PhpMyAdmin**, chương trình này hỗ trợ **các thao tác đối với CSDL** trong MySQL với giao diện dễ sử dụng.

Trong phần này tôi hướng dẫn thêm bạn thực hiện thao tác với Database trong giao diện **dòng lệnh MSDOS** bởi vì các giao diện khác tôi nghĩ tự bạn có thể làm được. Bật màn hình dòng lệnh DOS lên, chuyển sang thư mục cài đặt MySQL có chứa tập tin **mysql.exe** (/mysql/bin) và gõ vào mysql **<Enter>.**

Tại dấu nhấu nhắc lệnh hãy gõ lệnh để tạo ra một database mới:

```
mysql> create database guestbook;
Query OK, 1 row affected (0.00 sec)
mysql>
```

Bây giờ trong CSDL guestbook bạn cần phải có table chứa thông tin của khách. Bạn hãy dùng lệnh **create table** từ dấu nhắc lệnh:

```
mysql> use guestbook
Database changed
mysql> create table guestbook
-> (
-> name varchar(40) null,
-> location varchar(40) null,
-> email varchar(40) null,
-> url varchar(40) null,
-> comments text null
-> )
-> ;
Query OK, 0 rows affected (0.00 sec)
mysql>
```

Bạn đã có một CSDL tên là **guestbook** và một table ở trong CSDL này cũng tên là guestbook. Bây giờ đã đến lúc chúng ta viết một ứng dụng bằng PHP để thực hiện các thao tác: *xem, chèn, sửa, xoá* các thông tin trong CSDL **guestbook**.

Viết lệnh PHP

Bạn hãy dùng một chương trình soạn thảo văn bản đơn giản như Notepad chẳng hạn. Đặt tên cho các tập tin là **.php**, các tập tin này được lưu trữ trong thư mục gốc của **web local** trên máy bạn **(nên tham khảo CD cài Apache để rõ hơn).**

Cú pháp cơ bản

Điều thú vị đối với PHP là cho phép bạn xen kẽ giữa lệnh HTML và lệnh PHP. Do đó, PHP được xem như là một script giống như Javascript hay Vbscript. Các lệnh của PHP được gói trong thẻ mở: <? và thẻ đóng: ?>

Bây giờ bạn hãy thử chạy tập tin hi.php sau:

```
<?
echo “Hi,”;
?>
mom.
```

Khi chạy bạn sẽ gõ vào: localhost/hi.php

Kết quả cho ra là "Hi, mom". Ta thấy chữ "Hi," nằm trong tag lệnh PHP còn chữ "mom" thuộc về HTML.

Tuy nhiên, PHP còn làm được nhiều điều khác nữa, cũng giống như các ngôn ngữ lập trình khác, nó có thể làm việc với các loại biến, kiểu dữ liệu, chứa rất nhiều hàm chức năng. Hãy tìm hiểu ví dụ sau:

```
<?
echo “hi, mom.” ;
$var = date(“H”);
if ($var <= 11)
{
echo “good morning”;
}
elseif ($var > 11 and $var < 18)
{
echo “good afternoon”;
}
else
{
echo “good evening”;
}
?>
```

Nếu như bạn thấy khó hiểu thì cũng không sao. Chúng ta sẽ biết tường tận hơn ở phần sau.

Trang kết quả sẽ hiển thị các lời chào tuỳ thuộc vào giờ giấc hiện tại. Tôi đã dùng hàm date của PHP để lấy ra được giờ giấc hiện tại. Giá trị giờ được đem gán cho biến $var. Kế đến là các chọn lựa được sử dụng để đưa ra lời chào thích hợp.
Các bạn hãy để ý một chút, các lệnh của PHP đều được kết thúc với dấu chấm phẩy (;). Trong phát biểu IF chúng ta thấy dấu ngoặc nhọn { } chứa các lệnh tuỳ sẽ được thi hành tuỳ thuộc vào điều kiện. Các điều kiện thì được bao trong dấu ngoặc đơn ( ). Hàm date() và lệnh echo chỉ là 2 trong hàng trăm các hàm và lệnh có trong PHP mà các bạn sẽ được học một số cần thiết của chúng trong các chương sau. Bây giờ bạn hãy tìm hiểu thêm một số lệnh về CSDL.

Lệnh PHP để kết nối Database (quan trọng)

Bởi vì PHP và MySQL hiện tại trên máy của bạn vẫn còn là hai thế giới tách biệt nhau. Do đó, muốn dùng PHP để làm việc được với CSDL bạn cần phải tạo ra sợi dây liên kết giữa hai chiến hữu này.
Vì có thể có rất nhiều database trong MySQL, do đó bạn cần phải chỉ ra bạn muốn sử dụng database nào trong MySQL. Chúng ta hãy thực hiện như sau:

```
<?
mysql_connect(“localhost”, “khai”,”kkk”) or
die (“Could not connect to database”);
mysql_select_db(“guestbook”) or
die (“Could not select database”);
?>
```

Dòng đầu tiên thực hiện việc kết nối với Database Server đang nằm trên máy **localhost**, có username là **khai**, password là **kkk**.

Nếu kết nối thành công, nó sẽ thực hiện bước kế tiếp là kết nối với database nằm trong đó là **guestbook** bằng lệnh `mysql_select_db()`

Các bạn nên lưu ý là các lệnh trên bạn sử dụng thường xuyên cho mọi kết nối CSDL của bạn, do đó tôi khuyên bạn nên lưu nó vào một tập tin (dbconnect.php chẳng hạn), sau này cần thì chỉ việc dùng lệnh *`include('dbconnect.php');`*

Nhập dữ liệu vào Database

Bởi vì hiện tại database của bạn vẫn chưa có user nào, cho nên tôi sẽ hướng dẫn bạn viết các lệnh để thực hiện việc này. Nhưng trước tiên, bạn cần phải biết thêm một chút ít về biến trong PHP. Ở phần trước bạn đã xem qua một ví dụ trong đó có chứa biến, tuy nhiên đối với môi trường client/server, bạn cần phải làm việc với biến data từ client. Bạn sẽ thường xuyên làm việc với form HTML (bạn có thể tìm hiểu kỹ hơn ở phần Phục lục A. Chúng ta nên biết là mỗi phần tử của form đều có một cái tên, và khi bạn submit một form nào đó thì các tên của các phần tử trong đó trở thành một biến trong script PHP được form submit đến. Với form như sau, khi được submit, các biến $surname và $submit sẽ được tạo ra trong *myscript.php*. Giá trị $surname sẽ mang giá trị mà user đã nhập vào. Giá trị của $submit sẽ là chuỗi "submit".

```
<form action=“myscript.php”>
<input type=“text” name=“surnmae”>
<input type=“submit” name=“submit” value=“submit”>
</form>
```

Tôi xin lưu ý với các bạn là lập trình Web không giống như các dạng lập trình khác ở chỗ nó không ở trạng thái tĩnh. Để thể hiện một trang, Web Server phải trước hết nhận một thỉnh cầu từ trình duyệt. Giao thức sử dụng của chúng là HTTP, Hypertext Transfer Protocol. Các yêu cầu sẽ bao gồm: trang web mà trình duyệt sẽ thấy, form data, loại trình duyệt đang được sử dụng, địa chỉ IP mà trình duyệt sử dụng. Dựa vào thông tin này mà Web Server sẽ quyết định phục vụ những gì. Một khi server phục vụ yêu cầu trang web, nó sẽ duy trì sự kết nối với trình duyệt. Thông thường, bạn cần biết cách thức để chuyển các biến từ trang này sang trang khác. Bạn sẽ tìm thấy thao tác này trong ứng dụng tiếp theo. Ứng dụng của chúng ta sẽ giải quyết vấn đề dựa theo 1 trong 3 cách thức: *chuyển giao theo phần tử form ẩn, sử dụng cookies, sử dụng session*.

Bây giờ trở lại script sau:

```
<form action="myscript.php">
<input type="text" name="surnmae">
<input type="submit" name="submit" value="submit">
</form>
```

Bạn có thể quyết định cho hiển thị trên site những gì dựa vào các biến thông tin từ form HTML. Thông thường, bạn có thể kiểm tra nếu form đã được submit hay chưa bằng cách kiểm tra biến $submit có chứa giá trị "submit" hay không.
Hãy bắt tay vào công việc đi thôi. Trang đầu tiên trong ứng dụng được gọi là **sign.php** có chứa một form HTML. Action của nó là *create_entry.php*. Sau đây là chi tiết dòng lệnh:

```
<h2>Sign my Guest Book!!!</h2>
<form method=post action=”create_entry.php”>
<b>Name:</b>
<input type=text size=40 name=name>
<br>
<b>Location:</b>
<input type=text size=40 name=location>
<br>
<b>Email:</b>
<input type=text size=40 name=email>
<br>
<b>Home Page URL:</b>
<input type=text size=40 name=url>
<br>
<b>Comments:</b>
```

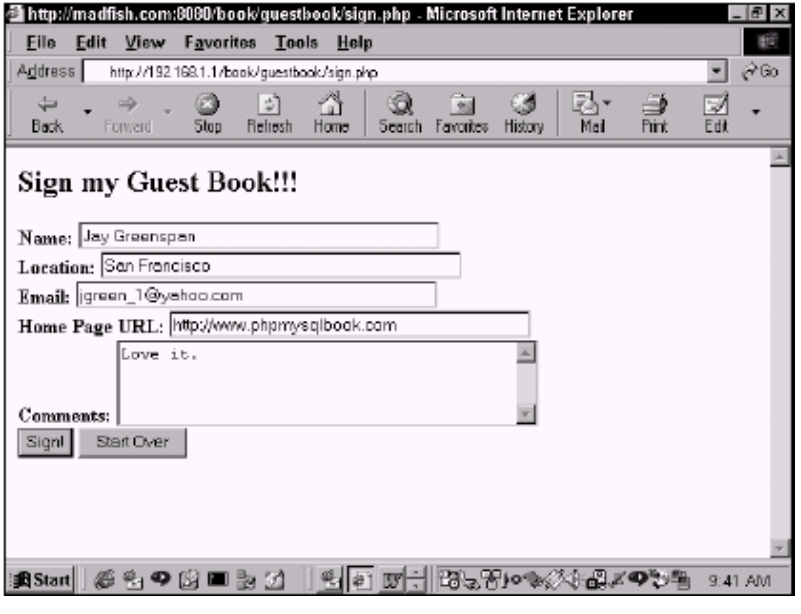
```
<textarea name=comments cols=40 rows=4 wrap=virtual></textarea>
<br>
<input type=submit name=submit value=”Sign!”>
<input type=reset name=reset value=”Start Over”>
</form>
```

Khi bạn điền đầy đủ thông tin ở trong form, thì các thông tin sẽ được chuyển đổi tới *create_entry.php*. Chuyện đầu tiên phải làm trên trang này là kiểm tra xem form đã được submit chưa. Nếu rồi, nhận lấy giá trị đã nhập vào trong form và sử dụng chúng để tạo một query đồng thời gởi đến MySQL. Bạn đừng lo lắng là không biết các lệnh SQL, điều trước tiên là chỉ cần biết là nó sẽ thực hiện việc chèn dữ liệu vào table của guestbook. Tập tin **create_entry.php** như sau:

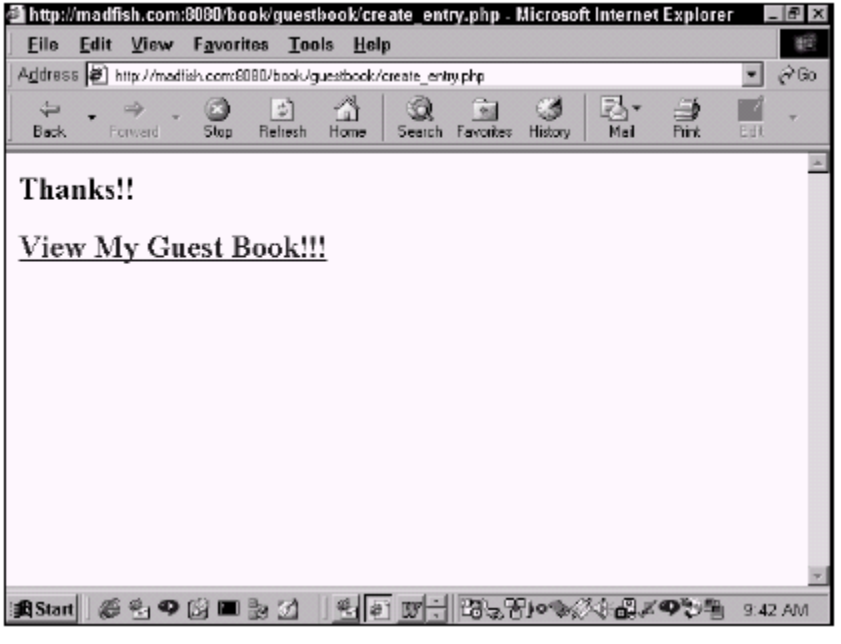
```
<?php
include(“dbconnect.php”);
if ($submit == “Sign!”)
{
$query = “insert into guestbook
(name,location,email,url,comments) values
(‘$name’, ‘$location’, ‘$email’, ‘$url’, ‘$comments’)”;
mysql_query($query) or
```

```
die (mysql_error());
?>
<h2>Thanks!!</h2>
<h2><a href=”view.php”>View My Guest Book!!!</a></h2>
<?php
}
else
{
include(“sign.php”);
}
?>
```

Trong lần đầu tiên *create_entry.php* được gọi, form *sign.php* sẽ được hiển thị. Kế tiếp, dữ liệu sẽ được cập nhật vào CSDL.
Hình sau minh hoạ các form được hiển thị:



sign.php



create_entry.php

Hiển thị dữ liệu trong Database lên màn hình

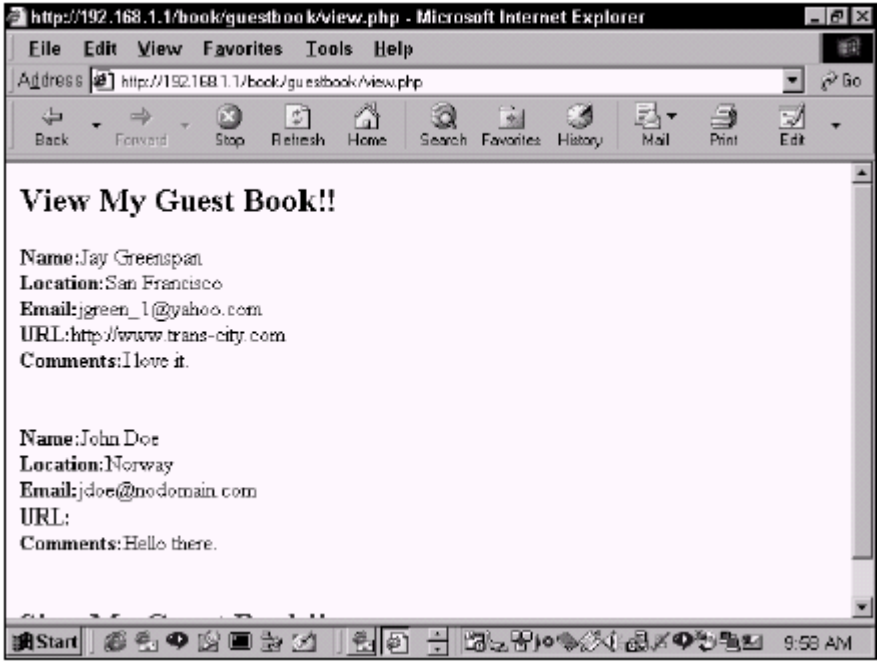
Bây giờ dữ liệu đã được ghi vào CSDL guestbook . Bạn cần thực hiện việc xem các dữ liệu đó. Nên nhớ là chúng ta lại phải sử dụng *dbconnect.php* như tôi đã nói với bạn trước đây. Bạn cần phải cho hiển thị tất cả các record trong table thông tin của khách viếng thăm đã nhập vào. Chúng ta thực hiện script sau và đặt tên là **view.php**:

```
<?php include("dbconnect.php"); ?>
<h2>View My Guest Book!!</h2>
<?php
$result = mysql_query("select * from guestbook") or
die (mysql_error());
while ($row = mysql_fetch_array($result))
{
echo "<b>Name:</b>";
echo $row["name"];
echo "<br>\n";
echo "<b>Location:</b>";
echo $row["location"];
echo "<br>\n";
echo "<b>Email:</b>";
echo $row["email"];
echo "<br>\n";
```

```
echo “<b>URL:</b>”;
echo $row[“url”];
echo “<br>\n”;
echo “<b>Comments:</b>”;
echo $row[“comments”];
echo “<br>\n”;
echo “<br>\n”;
echo “<br>\n”;
}
mysql_free_result($result);
?>
<h2><a href=”sign.php”>Sign My Guest Book!!</a></h2>
```

Như chúng ta thấy query trong MySQL truy cập tất cả các hàng trong database. Script thực hiện việc này bằng cách sử dụng vòng lập thông qua biến $row.

Trong mỗi vòng lặp thì mỗi field trong từng record được hiển thị. Vd: print $row["email"] sẽ ghi ra màn hình đối với record đang truy cập. Khi chạy chương trình, tất cả các field của từng record sẽ được hiển thị:


http://192.168.1.1/book/guestbook/view.php - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address http://192.168.1.1/book/guestbook/view.php
Go
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit
View My Guest Book!!
Name:Jay Greenspan
Location:San Francisco
Email:jgreen_1@yahoo.com
URL:http://www.trans-city.com
Comments:I love it.
Name:John Doe
Location:Norway
Email:jdoe@nodomain.com
URL:
Comments:Hello there.
Start
9:58 AM

view.php

Chương trình này upload lên internet được chưa?

Bây giờ thì ứng dụng đầu tiên của bạn đã hoàn tất. Nếu bạn muốn upload web vừa tạo lên trên Server để thử nghiệm thì cũng được thôi. Nhưng muốn để nó trở thành một site guestbook đúng nghĩa thì chưa được đâu! Bạn còn cần phải làm nhiều thứ để dữ liệu bạn không bị hacker quấy phá.... **(Còn tiếp)**

Lưu ý:
Bạn có thể tìm thấy các CD thiết kế web PHP tại các cửa hàng CD (Tôn Thất Tùng, tp.HCM chẳng hạn). Trên CD có trình cài đặt Apache, PHP, MySQL thông dụng là FOX. Ngoài ra còn có các Website PHP mẫu.

CÁC LỆNH Ở BÀI HỌC TRÊN TỐT NHẤT BẠN NÊN GÕ LẠI, KHÔNG NÊN COPY!

tongphuockhai@mail15.com

- Kiến thức là kho báu không phải của riêng ai. Vì vậy bạn đừng ngần ngại khi chia sẻ cho mọi người.

LẬP TRÌNH WEB ĐỘNG VỚI

PHP / MySQL

- GUESTBOOK
- CATALOG
- FORUM
- SHOPPING CART

PHẦN 2

Tống Phước Khải (tổng hợp& biên dịch)

Phương pháp truy xuất CSDL

MySQL

1- Từ Command Prompt

2- Bằng lệnh PHP

3- Dùng phpMyadmin

Để hiểu được chương này một cách rõ ràng, trước tiên bạn cần phải có một số kiến thức cơ bản về **Cơ Sở Dữ Liệu quan hệ**. Nếu bạn đã học qua một khoá căn bản về MS Access trong chương trình đào tạo chứng chỉ B chẳng hạn thì hẳn nhiên bạn có thể tiếp tục. Còn nếu như bạn chưa biết gì về nó thì tôi sẽ bàn đến nó trong phần Phụ Lục của giáo trình này hoặc bạn có thể tìm ngay một tài liệu tham khảo về CSDL, dễ nhất là tài liệu và MS Access ...

... Tôi chắc rằng bây giờ bạn đã có kiến thức về CSDL và hiểu biết Table là gì rồi! Có hàng khối công việc bạn sẽ phải làm việc đối với các Table và bạn sẽ được hướng dẫn cặn kẽ trong quyển sách này. Bạn sẽ phải vượt qua một số kiến thức về nó để mới có thể thành thạo trong thao tác với Table. Như bạn biết đấy: *Con đường đi đến thành công không có trải thảm sẵn đâu!*

Nếu bạn đã từng làm việc với *MS SQL Server* hay *Access* chúng đều có hỗ trợ việc tạo CSDL rất là dễ dàng với giao diện trực quan. Đối với MySQL bạn cũng có thể sử dụng công cụ trực quan đó là *phpMyadmin*.

Tuy nhiên, bạn phải học cách thao tác với CSDL bằng dòng lệnh, tôi chắc rằng điều này sẽ rất có ích cho bạn. Muốn chương trình của bạn trong lúc chạy thao tác tự động với CSDL thì bạn cần hàng tá lệnh PHP/SQL để thực hiện các yêu cầu của chương trình.

Trước khi chúng ta tạo các table trong CSDL của MySQL, có một vài thứ bạn cần phải hiểu rõ. Những khái niệm cơ bản mà tôi sắp giới thiệu sau đây rất quan trọng. Bạn hãy chắc rằng mình đã nắm kỹ về chúng trước khi thực hiện việc thiết kế dữ liệu.

Null

Việc đầu tiên bạn phải làm trong việc thiết kế một table là quyết định xem một field có cho phép giá trị NULL hay không.

Trong CSDL quan hệ, giá trị NULL của một field đồng nghĩa với nó có thể chấp nhận không có dữ liệu trong đó. Nên nhớ rằng giá trị NULL khác với giá trị của một chuỗi không có ký tự trong đó hoặc số có giá trị 0.

Đôi khi trong chương trình, bạn sẽ thực hiện một số động tác so sánh xem một chuỗi nào đó có chứa giá trị hay không, nó có thể là một câu lệnh IF. Xét một ví dụ PHP như sau:

```
$var //this is a variable used in the test
if ($var == “”)
{
echo “Var is an empty string”;
} else {
echo $var;
}
```

Nếu bạn thực hiện việc so sánh xem giá trị số có phải 0 hay không thì cũng thực hiện tương tự.

Động tác so sánh trên sẽ không làm việc được đối với giá trị NULL. Bạn nên hiểu rằng NULL là không chứa bất kỳ giá trị gì trong đó, cho nên việc so sánh trị không mang ý nghĩa gì cả. Trong chương 3 bạn sẽ thấy rằng đối với giá trị NULL đòi hỏi lập trình viên phải rất cân nhắc khi viết lệnh liên kết table.

Trong lệnh SELECT của SQL, có một số cách để bạn có thể kiểm tra nếu như một field chứa giá trị NULL. Trước hết bạn hãy sử dụng hàm Isnull(). Giả sử tìm một record trong table mà giá trị midle_name là NULL, bạn có thể sử dụng query sau:

```
select * from names where isnull(middle_name);
```

Hoặc lấy các record mà middle_name khác NULL:

```
select * from names where !isnull(middle_name);
```

Bạn cũng có thể sử dụng ***is null*** *và* ***is not null:***

```
select * from users were addr2 is null;
select * from users where addr2 is not null;
```

Để rõ hơn, bạn hãy xem chuyện gì xảy ra khi tôi cố gắng liên kết hai table sau:

Khach_hang

ten	ho_lot	ba_xa
Trung	**Le Anh**	**1321**
Khai	**Tong Phuoc**	**Null**

Hon_nhan

ba_xa	ten	ho_lot
1321	**Diem**	**Nguyen Thuy**

Nếu bạn muốn tìm tên các **khách hàng** và tên những **bà xã** của họ, bạn sẽ phải liên kết 2 table này thông qua field ba_xa. (Xin bạn chớ lo lắng khi chưa hiểu về cú pháp, bạn sẽ học ngay ở phần tiếp theo thôi).

```
SELECT * FROM khach_hang, hon_nhan
WHERE khach_hang.ba_xa = hon_nhan.ba_xa
```

Việc thực hiện này chỉ đúng đối với Trung, nhưng sẽ có vấn đề đối với Khai bởi vì anh ta hãy còn độc thân và ba_xa của anh ta là NULL.
Trong chương 3 bạn sẽ khảo sát kỹ hơn về vấn đề này.

Index

Người ta nói rằng ưu điểm vượt trội của Hệ quản trị CSDL quan hệ là nó thực hiện các việc tìm kiếm hay sắp xếp những khối lượng dữ liệu khổng lồ một cách rất nhanh chóng. Sở dĩ nó thực hiện được việc này là do nó có chứa một cơ cấu lưu trữ dữ liệu gọi là INDEX.
INDEX cho phép database server tạo được một field đặc trưng tìm kiếm với tốc độ khó ngờ. Các INDEX đặc biệt hỗ trợ một hoặc một nhóm các record trong một table chứa số lượng lớn các record. Chúng cũng hỗ trợ tốc độ cho các hàm liên kết hoặc tách nhóm dữ liệu như min(), max(), bạn sẽ tìm hiểu ở Chương 3.
Với các tính năng vượt trội này, tại sao người ta lại không tạo index trong tất cả các field của một table? Có một số điều trở ngại như sau: Thứ nhất, index có sẽ làm chậm một số tiến trình trong CSDL. Mỗi lần bảo trì các index Database Server phải mất khá nhiều thời gian. Có một vài trường hợp chính các index làm cho chúng chậm hẳn. Nếu như trên table của bạn tất cả các record đều giống y như nhau thì không có lý do gì để bạn tạo index. Các index dư thừa chỉ làm cho tốn thêm không gian đĩa của bạn mà thôi.

Đối với một table có gắn khoá chính (primary key) thì thông thường field có khoá này được dùng vào việc tìm kiếm cho nên index sẽ được gán tự động trên field này.
Bạn sẽ gặp lại rất nhiều công việc tạo index ở các phần tiếp theo sau.

Lệnh CREATE Database

Trước khi tạo được một Table thì điều tất yếu là bạn phải tạo được một Database cái đã. Việc này dễ dàng và nhanh chóng thôi. Lệnh CREATE được sử dụng như sau:

mysql> create database database_name;

Nếu như bạn thắc mắc rằng sau khi tạo database nó sẽ nằm ở trong thư mục nào trên ổ đĩa của bạn thì bạn hãy tìm trong ..\mysql\data xem có không.

Khi đặt tên cho database, hay đặt tên cho field và index gì đấy tránh trường tránh trường hợp đặt những cái tên khó nhớ hoặc dễ bị lẫn lộn. Đối với một số hệ thống Unix chẳng hạn có sự phân biệt chữ HOA/thường thì CSDL chạy trên nó cũng ảnh hưởng theo.
Bạn hãy chọn một quy ước cho riêng mình trong việc đặt tên để khỏi nhầm lẫn về sau. Chẳng hạn tên của table và field đều đặt chữ thường chẳng hạn. Nên nhớ là không được sử dụng khoảng trắng.

Bây giờ bạn tìm hiểu cả hai cách tạo database: Cách thứ nhất tạo thủ công từ dấu nhắc dòng lệnh DOS, cách thứ hai sử dụng các lệnh trong PHP.
Cách thứ nhất tôi đã có trình bày ở chương giới thiệu và bạn đã tạo một database tên là guestbook. Cú pháp tạo như sau:

mysql> create database guestbook;

Cách thứ hai là sử dụng lệnh trong PHP, bạn có thể dùng hàm mysql_create_db() hoặc mysql_query(). Nhưng nên nhớ trước khi tạo bạn phải thực hiện được kết nối với database server.

```
$conn = mysql_connect("localhost","username", "password")
or die ("Could not connect to localhost");
mysql_create_db("my_database") or
die ("Could not create database");
$string = "create database my_other_db";
mysql_query($string) or
die(mysql_error());
```

Lệnh USE Database

Sau khi đã tạo được một database mới trong database server bạn sẽ bắt đầu chọn nó để sử dụng cho công việc của mình. Cách thực hiện như sau:

1. Command Prompt:

mysql> use database_name;

2. Trong PHP:

```
$conn = mysql_connect("localhost","username", "password")
or die ("Could not connect to localhost");
mysql_select_db("test", $conn) or
die ("Could not select database");
```

Lệnh CREATE Table

Lưu ý: *Lệnh này thực hiện sau khi đã có lệnh* ***CREATE Database****.*

Một khi bạn đã tạo và chọn database, việc tiếp theo là tạo một table. Bạn sẽ sử dụng lệnh Create Table như sau:

```
create  table  table_name
(
column_1  column_type  column_attributes,
column_2  column_type  column_attributes,
primary key (column_name),
index index_name(column_name)
)
```

Đối với thuộc tính các field (cột) chúng ta cần bàn về:

- **null** hoặc **not null**
- **default**

Nếu bạn không định nghĩa NULL hay NOT NULL thì NULL sẽ được chọn làm giá trị mặc định. Hãy xét ví dụ sau:

```
create  table  topics2 (
topic_id  integer  not null  auto_increment,
parent_id  integer  default 0  not null,
root_id  integer  default 0,
name  varchar(255),
description  text  null,
create_dt  timestamp,
modify_dt  timestamp,
author  varchar(255)  null,
author_host  varchar(255)  null,
primary  key(topic_id),
index  my_index(parent_id))
```

Trong ví dụ trên bạn tạo ra một table có tên topics2, có tất cả 8 field và có 2 index, một index cho khoá chính và một cho parent_id. Type của các field trên lần lượt là integer, varchar, text, timestamp. Giá trị đứng sau defaul là giá trị mặc định bạn gán cho một ô trong record khi không nhập liệu vào.

Bây giờ chúng ta áp dụng các lệnh này vào một chương trình PHP để tạo table, hàm mysql_query () được sử dụng:

```
$conn = mysql_connect("localhost","username", "password") or
die ("Could not connect to localhost");
mysql_select_db("test", $conn) or
die("could not select database");
$query = "create table my_table (col_1 int not null primary key,
col_2 text)";
mysql_query($query) or
die(mysql_error());
```

** Lưu ý: username và password tuỳ thuộc vào MySQL của bạn.Ví dụ:*

```
$conn = mysql_connect("localhost","minhtrung", "zadfdfaked") or
```

Thông thường các nhà cung cấp host PHP sử dụng localhost, tuy nhiên một số hosting không sử dụng localhost như Yahoo chẳng hạn. Do đó bạn cần xem hướng dẫn của nhà cung cấp host.

Kiểu dữ liệu

Bây giờ chúng ta hãy bàn về các kiểu dữ liệu (type) gán cho field trong table. Có rất nhiều kiểu khác nhau chúng ta sẽ lần lượt khảo sát từng kiểu một.

Kiểu chuỗi văn bản

MySQL có 7 kiểu dành cho dữ liệu kiểu chuỗi văn bản:

- char
- varchar
- tinytext
- text
- mediumtext
- longtext
- enum

CHAR

Cách sử dụng: char(length)

Length có giá trị tối đa là 255. Giả sử bạn sử dụng khai báo char(10) thì bạn chỉ được phép nhập vào tối đa 10 ký tự mà thôi.

VARCHAR

Cách sử dụng: varchar(length)

Kiểu này cũng gần giống như kiểu CHAR có độ dài tối đa cũng là 255. Điểm khác biệt của varchar là nó chỉ là biến lưu trữ độ dài, cho nên nó sẽ không thay đổi khi giá trị của của ô dữ liệu dài hay ngắn. MySQL sẽ sinh ra một ký dùng làm biến chứa độ dài của field kiểu varchar. Đồng thời MySQL sẽ thực hiện chức năng loại bỏ các khoảng trống trong mỗi ô dữ liệu nếu như không được sử dụng hết.

USING CHAR OR VARCHAR

Có sự khác nhau trong việc sử dụng CHAR và VARCHAR. Sau đây là phương hướng lựa chọn của bạn.

Giả dụ bạn tạo một field là ĐỊA CHỈ và bạn dự tính độ dài tối đa là 150. Có những trường hợp địa chỉ rất ngắn ví dụ: *1 Lê Lợi, Q.1, TPHCM*. Bạn chỉ sử dụng có 20 ký tự, như vậy còn

trống rất nhiều ký tự không dùng đến. Trong trường hợp này bạn nên sử dụng kiểu VARCHAR (150).
Trường hợp field của bạn là MÃ SỐ chẳng hạn, và bạn cho độ dài tối đa là 6 theo quy ước tạo mã của bạn ví dụ: KH0001 . Trong trường hợp này các ô khác đều được nhập theo chuẩn định sẵn luôn luôn là 6 ký tự cho nên không việc gì bạn phải sử dụng VARCHAR để MySQL phải nhọc công theo dõi độ dài của các ô mỗi khi nhập vào. Bạn chỉ sử dụng VAR(6) là được.

> Trong trường hợp bạn chọn kiểu dữ liệu là varchar(4) thì MySQL sẽ tự động đổi lại là kiểu char.

TINYTEXT
Cách sử dụng: tinytext
Đây là một trong bốn kiểu text nhị phân. Tất cả 4 kiểu này (tinytext, text, mediumtext, largetext) đều là kiểu biến tương tự như varchar. Chúng khỉ khác nhau về độ dài của cho phép của ký tự mà thôi. Tuy nhiên, đối với TINYTEXT thì cho phép độ dài tối đa là 255, giống như varchar(255). Cho phép tạo index trên toàn bộ các ký tự của field này.

TEXT

Cách sử dụng: text

Cho phép độ dài tối đa là 65,535 ký tự. Có thể tạo index trên 255 ký tự đầu.

MEDIUMTEXT

Cách sử dụng: midiumtext

Cho phép độ dài tối đa là 16,777,215 ký tự. Có thể tạo index trên 255 ký tự đầu.

LONGTEXT

Cách sử dụng: longtext

Cho phép độ dài tối đa là 4,294,967,295 ký tự. Có thể tạo index trên 255 ký tự đầu. Tuy nhiên loại này không thông dụng bởi vì MySQL chỉ hỗ trợ chuỗi 16 triệu bytes.

ENUM

Cách sử dụng: enum (‘value1’, ‘value2’, ‘value3’ ...) [default ‘value’]

Với enum bạn có thể giới hạn các giá trị được định sẵn cho một field. Cho phép bạn định trước tối đa 65.535 giá trị.

Thông thường người ta dùng kiểu này cho field chứa giá trị Yes hoặc No. Ví dụ:

```
create  table my_table (
id  int  auto_increment  primary key,
answer enum (‘yes’, ‘no’) default ‘no’
);
```

SET

Cách sử dụng: set (‘value1’, ‘value2’, ‘value3’ ...) [default ‘value’]

Kiểu này định nghĩa một tập hợp hàng loạt các giá trị định trước. Tuy nhiên, cách này ít được dùng bởi vì nó phá vỡ cấu trúc thiết kế CSDL (một field có quá nhiều kiểu) và các bạn sẽ không thấy tôi sử dụng trong quyển sách này.

Kiểu dữ liệu số

MySQL có tất cả 7 kiểu số. Lưu ý rằng các kiểu sau đây là giống nhau: int/ integer, double/double precision/real, decimal/numeric

- int/integer
- tinyint
- mediumint
- bigint

- float
- double/double precision/real
- decimal/numeric

Đối với tất cả kiểu số, giá trị lớn nhất cho phép là 255. Đối với một số kiểu cho phép bạn thể hiện các số 0 đứng đầu. Giả sử bạn có một field có độ dài là 10 thì số 25 trong ô dữ liệu sẽ được thể hiện 0000000025. Field số còn được định nghĩa là có dấu (signed) hoặc không dấu (unsigned). Mặc định là có.

INT/INTEGER

Cách sử dụng: int(display size) [unsigned] [zerofill]

Nếu bạn dùng không dấu thì giá trị của field cho phép là từ 0 đến 4.294.967.295. Nếu dùng có dấu thì giá trị từ –2.147.483.648 đến 2.147.483.647. Kiểu Int sẽ sử dụng auto_increment (tự động theo chiều tăng) để định nghĩa khoá chính của table.

```
create table my_table (
table_id int unsigned auto_increment primary key,
next_column text
);
```

Để ý rằng bạn sử dụng không dấu (unsigned) bởi vì auto_increment không dùng cho các giá trị âm.

TINYINT

Cách sử dụng: tinyint(display size) [unsigned] [zerofill]

Nếu không dấu, tinyint sẽ chứa các giá trị nguyên từ 0 đến 255. Nếu có dấu thì từ -128 đến 127.

MEDIUMINT

Cách sử dụng: mediumint(display size) [unsigned] [zerofill]

Có dấu: có giá trị từ -8.388.608 đến 8.388.607

Không dấu: có giá trị từ 0 đến 1677215

BIGINT

Cách sử dụng: bigint(display size) [unsigned] [zerofill]

Có dấu: -9.223.372.036.854.775.808 đến 9.223.372.036.854.775.807

Không dấu: from 0 to 18.446.744.073.709.551.615

FLOAT

Cách sử dụng 1: FLOAT(precision) [zerofill]

Với cách sử dụng này, cho phép chứa các số thập phân không dấu. Số lượng phần thập phân có thể là <=24 đối với loại single và 25 đến 53 đối với loại double. Các version trước đây của MySQL, luôn chia làm 2 loại:

Cách sử dụng 2: FLOAT[(M,D)] [ZEROFILL]

Đây là loại single và giá trị có thể là từ -3,402823466E+38 đến -1,175494351E-38, số 0, và từ 1,175494351E-38 đến 3,402823466E+38. M là phần nguyên, D là phần thập phân. ????

DOUBLE/DOUBLE PRECISION/REAL

Cách sử dụng 1: DOUBLE[(M,D)] [zerofill]

Cho phép giá trị từ -1,7976931348623157E+308 đến -2,2250738585072014E-308, số 0 và 2,2250738585072014E-308 đến 1,7976931348623157E+308. M là phần nguyên, D là phần thập phân.

Cách sử dụng 2: DECIMAL[(M[,D])] [ZEROFILL]

Các số trong phần thập phân được lưu trữ như ký tự. Mỗi số được xem như một ký tự chuỗi. Nếu D = 0 thì sẽ không có phần thập phân. Giá trị thập phân giống như dạng Double. ????

Kiểu dữ liệu ngày, giờ

MySQL có 5 dạng ngày giờ:

- date _ time
- datetime _ year
- timestamp
- time
- year

Ngày và giờ trong MySQL rất uyển chuyển, nó có thể chấp nhận kiểu chuỗi hoặc số hãy xét ví dụ sau:

```
create table date_test(
id  int  unsigned auto_increment,
a_date  date
);
```

Sau đó dùng insert để đưa giá trị ngày vào a_date:

```
insert  into  date_test (a_date) values ('00-06-01');
insert  into  date_test (a_date) values ('2000-06-01');
```

```
insert into date_test (a_date) values ('20000601');
insert into test6 (a_date) values (000601);
```

MySQL tương thích với việc nhận giá trị ngày là kiểu chuỗi hơn. Cho nên "000501 là chọn lựa thích hợp hơn là việc nhập một số nguyên. Sử dụng giá trị chuỗi cho ngày giúp bạn sẽ tránh được một số sự cố về sau.

MySQL có hỗ trợ một số hàm giúp bạn trong việc rút trích dữ liệu dạng ngày.

DATE

Cách sử dụng: date

Định dạng của ngày như sau: YYYY-MM-DD. Cho phép bạn các giá trị từ 1000-01-01 đến 9999-12-31.

DATETIME

Cách sử dụng: datetime [null | not null] [default]

Định dạng của ngày giờ như sau: YYYY-MM-DD HH:MM:SS. Cho phép bạn các giá trị từ 1000-01-01 00:00:00 đến 9999-12-31 23:59:59.

TIMESTAMP

Cách sử dụng: timestamp(size)

Đây là kiểu dữ liệu ghi nhận tự động giờ giấc sửa đổi gần nhất đối với một record, bất khi khi nào nó được tạo ra, hoặc cập nhật đều xảy ra việc ghi nhận này. Size của nó có thể định nghĩa trong khoảng từ 2 đến 14. Bảng sau trình bày các size. Giá trị mặc định là 14.

Size	Định dạng
2	YY
4	YYMM
6	YYMMDD
8	YYYYMMDD
10	YYMMDDHHMM
12	YYMMDDHHMMSS
14	YYYYMMDDHHMMSS

TIME

Cách sử dụng: time

Lưu trữ dạng giờ theo định dạng HH:MM:SS và có giá trị từ –838:59:59 đến 838:59:59. Lý do mà giá trị này lớn như vậy là để nó có thể chứa được các kết quả tính toán giờ giấc.

YEAR

Cách sử dụng: year[(2|4)]

Chứa dữ liệu dạng năm. Nếu sử dụng hai ký tự để biểu thị năm thì biểu diễn được từ Có giá trị từ 1970 cho đến 2069, nên nhớ: 70 đến 99 biểu thị từ 1970 đến 1999, còn 01 đến 69 biểu thị từ 2001 đến 2069.

Dùng 4 ký tự thì biểu diễn được từ 1901 đến 2155.

Tạo chỉ mục INDEX

Bắt đầu từ phiên bản 3.23.6 của MySQL bạn có thể tạo index trên bất kỳ field nào. Cho phép 1 table có 16 field chứa index. Cú pháp như sau:

```
index index_name (indexed_column)
```

> Mặc dù index_name là tuỳ chọn, nhưng bạn nên luôn luôn cho nó một cái tên. Sau này nó rất cần thiết khi bạn muốn xoá bỏ index của một field nào đó trong lệnh SQL của bạn. Nếu bạn không cho tên thì MySQL sẽ chọn tên index của field đầu tiên.

Còn một cách nữa để tạo index là khai báo khoá chính trên field đó. Chú ý rằng bất kỳ field auto_increment (sort tự động) cũng phải được index, và bạn nên khai báo nó là khoá chính. Trong ví dụ sau id_col được index:

```
create table my_table (
id_col int unsigned auto_increment primary key,
another_col text
);
```

Khoá chính có thể cũng được khai báo giống như các index khác ngay sau khi định nghĩa một field:

```
create table my_table (
id_col int unsigned not null auto_increment,
another_col text,
primary key(id_col)
);
```

Index có thể trải rộng ra hơn một cột. Nếu như query sử dụng 2 hàng phối hợp với nhau trong khi thực hiện việc tìm kiếm, bạn có thể tạo một index bao gồm luôn cả 2 với các lệnh sau:

```
create table mytable(
id_col  int  unsigned not null,
another_col  char(200) not null,
index  dual_col_index(id_col, another_col)
);
```

Index này sẽ được sử dụng cho việc tìm kiếm vừa trên id_col vừa trên another_col. Các index này làm việc từ trái sang phải. Do đó index này sẽ được sử dụng cho việc tìm kiếm exclusive trên id_col. Tuy nhiên, nó sẽ không exclusive cho việc tìm kiến trên another_col.

Còn một điều về index nữa là bạn có thể tạo nó chỉ trên một phần của field. Bắt đầu từ phiên bản 3.23 của MySQL bạn có thể tạo index các field kiểu index tinytext, text, mediumtext và longtext trên 255 ký tự đầu. Đối với char và varchar, bạn có thể tạo index trên một số ký tự đầu của field. Cú pháp của nó như sau:

index index_name (column_name(column_length))

Ví dụ:

```
create table my_table(
char_column char (255) not null,
text_column text not null,
index index_on_char (char_column(20)),
index index_on_text (text_column(200))
);
```

Một index có thể đảm bảo giá trị duy nhất tồn tại trong mọi record của table bằng cách sử dụng lệnh *unique*.

```
create table my_table(
char_column char (255) not null,
text_column text not null,
unique index index_on_char (char_column));
```

Các loại Table

MySQL hỗ trợ các dạng talble sau: ISAM, MyISAM, BDB và Heap. ISAM là dạng table đã xưa và trong các ứng dụng mới không được sử dụng. Dạng table mặc định là MyISAM. Cú pháp để khai báo loại table này là:

```
create table table_name type=table_type(
col_name column attribute
);
```

Table dạng MyISAM có tốc độ tốt và tính ổn định cao. Không cần thiết phải định nghĩa một dạng table mới nào khác trừ phi bạn cần dùng loại khác cho vì một lý do đặc biệt nào đó.
Heap là dạng table thường trú trong bộ nhớ. Chúng không được lưu trữ trong đĩa cứng hay các thiết bị dùng để trữ tin. Cho nên nếu bị mất điện heap sẽ mất theo. Vì được trữ trong bộ nhớ nên heap có tốc độ truy cập cao. Tuy nhiên bạn chỉ dùng cho các table tạm thời trong lúc chạy chương trình.

Lệnh Alter table

Nếu như bạn muốn thay đổi các thành phần của table bạn sẽ sử dụng lệnh *alter table*. Bạn có thể thực hiện các thay đổi như: đổi tên table, field, index; thêm hoặc xoá field và index; định nghĩa lại các field và index. Cú pháp cơ bản của lệnh này là:

```
alter table table_name
```

Các lệnh còn lại tuỳ thuộc vào thao tác mà chúng ta sẽ bàn tiếp theo đây:

Đổi tên Table

Cú pháp như sau:

```
alter table table_name rename new_table_name
If you have MySQL version 3.23.27 or higher you can make use of the
rename statement.The basic syntax is
rename table_name to new_table_name
```

Thêm và xoá cột trong Table

Khi thêm field vào trong table, bạn sẽ cần phải có những định nghĩa cần thiết cho field đó. Từ phiên bản 3.22 của MySQL cho phép bạn chọn vị trí để đặt field mới vào trong table. Chức năng này không bắt buộc.

alter table table_name add column column_name column attributes

Ví dụ:

alter table my_table add column my_column text not null

Cách sử dụng định vị một field trong table: Sử dụng lệnh first để chèn field mới vào vị trí đầu của table. After để chèn vào vị trí sau cùng trên table:

alter table my_table add column my_next_col text not null first
alter table my_table add column my_next_col text not null after
my_other_column

Để xoá một field, bạn thực hiện lệnh sau:

```
alter table table_name drop column column name
```

Khi thực hiện lệnh alter đối với một table, bạn chỉ nên thực hiện một thao tác đối với table mà thôi. Ví dụ: Bạn thực hiện việc xoá một index, sau đó tạo một index mới, thì không nên gộp chung vào một lệnh alter mà nên thực hiện 2 lần.

Thêm và xoá Index

Bạn có thể thêm index bằng cách sử dụng lệnh index, unique và primary key, tương tự như việc sử dụng lệnh create vậy.

```
alter table my_table add index index_name (column_name1,
column_name2, ...)
alter table my_table add unique index_name(column_name)
alter table my_table add primary key(my_column)
```

Bạn cũng có thể bỏ index bằng cách sử dụng lệnh drop:

```
alter table table_name drop index index_name
alter table_name test10 drop primary key
```

Đổi thuộc tính của cột (field)

Thay đổi cách thành tố của field bằng lệnh change hoặc modify:

```
alter  table table_name  change    original_column_name  new_column_name
int not null
```

Lệnh sau là sai:

```
alter   table table_name  change  my_col2  my_col3;
```

Nếu bạn muốn chỉ đổi thuộc tính của field thì dùng lệnh change và tạo field mới cùng tên nhưng thay đổi thuộc tính. Giả dụ bạn đổi field col_1 từ kiểu char(200) sang varchar(200):

alter ***table*** *table_name* **change** *col_1 col_1 varchar(200)*

Với MySQL phiên bản 2.22.16 trở đi bạn có thể dùng lệnh modify:

```
alter table table_name modify 1 col_1 varchar(200)
```

Lệnh Insert

Bây giờ bạn đã biết tất cả những kiến thức cần thiết để tạo và thực hiện các thay đổi trên định dạng table, bây giờ bạn sẽ tìm hiểu cách thức để nhập thông tin vào table. Bạn sẽ thực hiện việc này bằng lệnh INSERT:

```
insert into table_name (column_1, column2, column3,...) values
(value1, value2, value3 ...)
```

Nếu trong một fiedl cho phép giá trị NULL bạn có thể không cần đưa vào trong lệnh INSERT.

Như bạn thấy các chuỗi tên field và giá trị đều được đặt trong dấu ngoặc đơn (). Ngoài ra giá trị kiểu chuỗi phải được bao bọc bởi dấu nháy đơn. Như vậy nếu như trong chuỗi có chứa dấu nháy đơn và dấu ngoặc đơn thì sao? Để tránh trường hợp lẫm lẫn giữa dấu chuỗi và dấu phân định nghĩa lệnh SQL có các qui định rằng muốn đưa ký hiệu đặc biệt vào chuỗi thì phải đặt chúng sau dấu \

‘ (single quote)
“ (double quote)
\ (backslash)

% (percent sign)
_ (underscore)

Bạn có thể thoát ra khỏi dấu nháy đơn bằng cách sử dụng 2 dấu ngoặc đơn đóng mở ().
Bạn sử dụng các ký tự đặc biệt sau để thực hiện một thao tác đặc thù trong câu lệnh:

\n (newline)
\t (tab)
\r (carriage return)
\b (back space)

Nên lưu ý một điều là, bạn không cần phải lo lắng về các ký tự thoát ở đây trong khi lập trình PHP. Bạn sẽ gặp được những hàm và xác lập trong PHP dùng để thực hiện việc này một cách tự động. Hàm addslashes() và các xác lập trong php.ini sẽ hỗ trợ việc này.

Lệnh Update

Lệnh UPDATE có một chút khác biệt so với các lệnh mà chúng ta đã khảo sát, nó thực hiện thông qua lệnh WHERE. Cú pháp thông thường là:

update table_name set col_1=value1, col_2=value_2 where col=value

Xin nhắc lại lần nữa, nếu bạn muốn thêm vào một chuỗi, bạn cần phải bao nó trong các dấu nháy đơn và dấu thoát. Nên nhớ rằng lệnh WHERE trong câu lệnh UPDATE có thể thực hiện bất kỳ phép so sánh nào ở phía sau nó. Thông thường nó hay được dùng để định xác định một record đơn với khoá chính. Trong table **folks** sau ID là khoá chính.

id	fname	lname	salary
1	Don	Liu	25,000
2	Don	Corleone	800,000
3	Don	Juan	32,000
4	Don	Johnson	44,500

Câu lệnh sau sẽ tác động đến Don Corleone:

```
update folks set fname='Vito' where id=2
```

Như bạn thấy, nếu như bạn dùng lệnh UPDATE dựa trên field fname thì thật không nên chút nào, vì bạn có thể cập nhật tất cả các field trong table này.

```
update folks set fname='Vito' where fname='don'
```

Bạn cũng có thể dùng UPDATE để thực hiện việc điều chỉnh tăng lương đối với nhân viên chẳng hạn:

```
update folks set salary=50000 where salary<50,000
```

Lệnh drop table/drop database

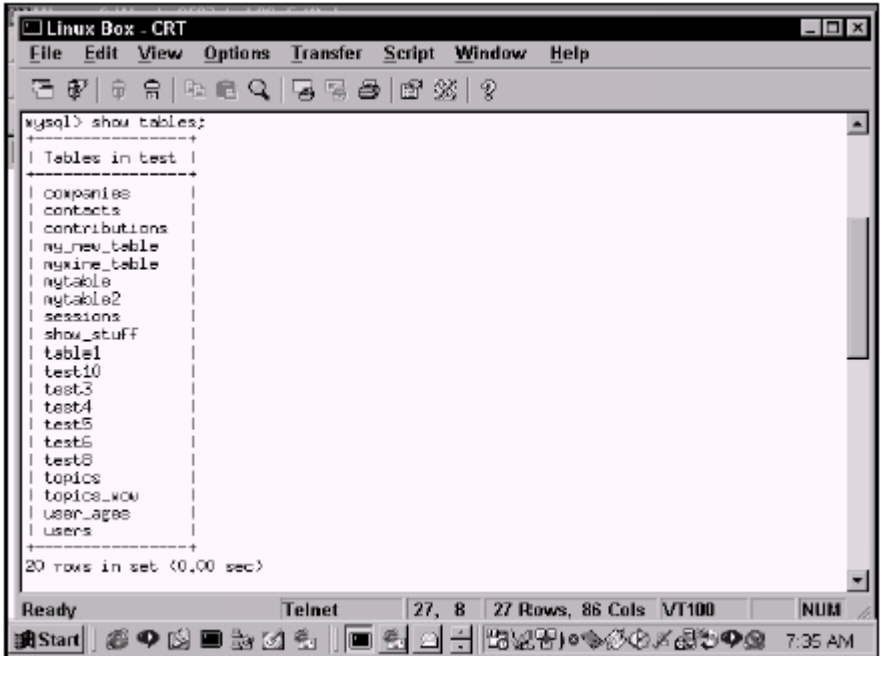
Lệnh DROP dùng để xoá table hoặc cả database. Nên nhớ một điều là một khi bạn thực hiện lệnh này rồi thì bạn không thể khôi phục lại dữ liệu của bạn. Hãy cẩn trọng!

drop table table_name
drop database database_name

Lệnh DROP TABLE được chuyển đổi sang PHP thông qua hàm mysqlquery(). Nếu bạn muốn xoá database trong PHP, bạn cần sử dụng hàm mysql_drop_db(). Sẽ được trình bày căn kẽ ở các phần sau!

Lệnh show tables

Để trình bày một danh sách các table có trong database, bạn sử dụng lệnh SHOW TABLES. Để thực hiện được lệnh này bạn nên lưu ý là phải chọn database trước đã bằng lệnh USE DATABASE
Hình sau là kết quả của việc thực hiện lệnh SHOW TABLES từ dấu nhắc lệnh.



Lệnh Show table

Trong PHP, bạn có thể cho hiển thị một danh sách table bằng cách sử dụng hàm MYSQL_LIST_TABLES():

```
<?
mysql_connect(“localhost”, “root”, “”);
$result = mysql_list_tables(“test”);
while($row = mysql_fetch_array($result))
{
echo $row[0] . “<br>\n”;
}
?>
```

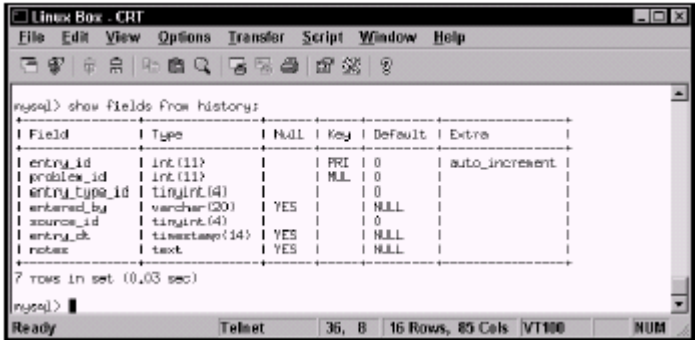
Lệnh show columns /show fields

Các lệnh cho kết quả giống nhau. Bây giờ bạn hãy dùng lệnh CREATE đã học để tạo một table sau đó dùng một trong hai lệnh trên để cho hiển thị các field của nó:

```
create table topics (
topic_id integer not null auto_increment primary key,
parent_id integer default 0 not null,
root_id integer default 0,
name varchar(255),
```

```
description text null,
create_dt timestamp,
modify_dt timestamp,
author varchar(255) null,
author_host varchar(255) null,
index my_index(parent_id)
)
```

Hình sau trình bày kết quả của lệnh SHOW FIELDS thực hiện từ dấu nhắc lệnh:



Lệnh Show Fields

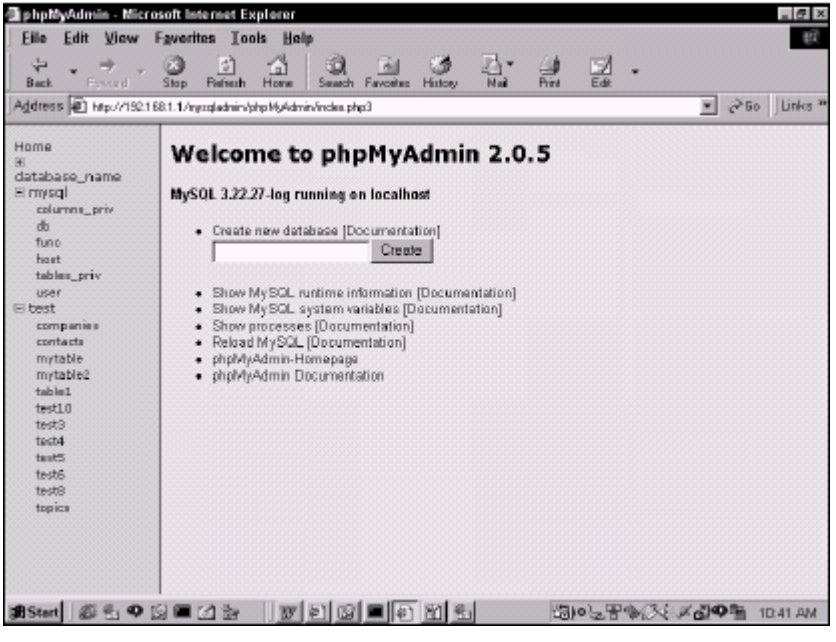
Bạn có thể cho ra kế quả tương tự trong PHP bằng cách sử dụng hàm MYSQL_FIELD_NAME(), MYSQL_FIELD_TYPE(), và MYSQL_FIELD_LEN(). Tất cả các các hàm này được trình bày cặn kẽ ở các phần sau.

```
$db = mysql_connect(“localhost”,”root”, “”)
or die (“Could not connect to localhost”);
mysql_select_db(“test”, $db)
or die (“Could not find test”);
$db_name =”topics”;
$query = “select * from $db_name”;
$result = mysql_query($query);
$num_fields = mysql_num_fields($result);
//create table header
echo “<table border = 1>”;
echo “<tr>”;
for ($i=0; $i<$num_fields; $i++)
{
echo “<th>”;
echo mysql_field_name ($result, $i);
echo “</th>”;
```

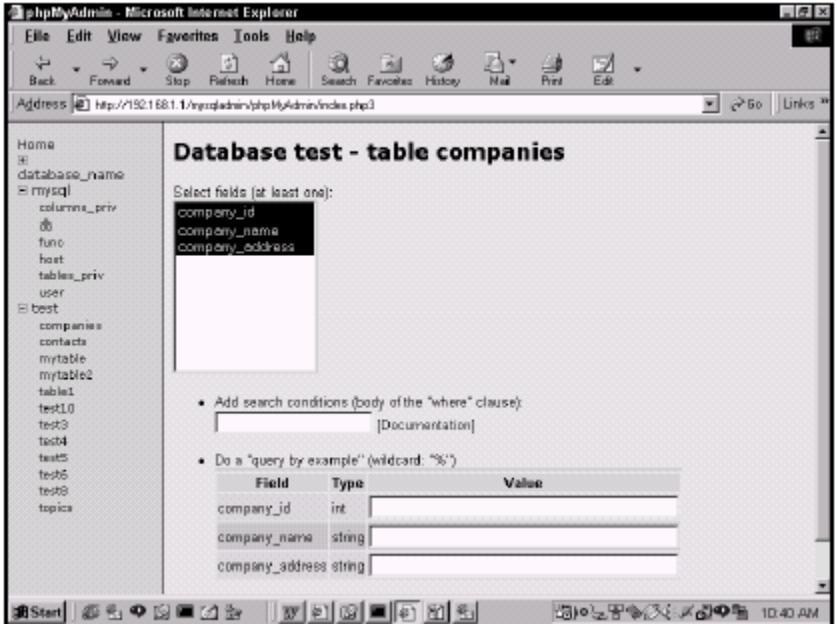
```
}
echo “</tr>”;
//end table header
//create table body
echo “<tr>”;
for ($i=0; $i<$num_fields; $i++)
{
echo “<td valign = top>”;
echo mysql_field_type ($result, $i) . “<br> \n”;
echo “(“ . mysql_field_len ($result, $i) . “)<br> \n”;
echo mysql_field_flags ($result, $i) . “<br> \n”;
echo “</td>”;
}
echo “</tr>”;
//end table body
echo “</table>”;
```

Sử dụng phpMyAdmin

Tôi không loại trừ trường hợp bạn là tín đầu của DẤU NHẮC LỆNH. Có khả năng là bạn không thích GIAO DIỆN ĐỒ HOẠ vì có thể bạn cho rằng giao diện này dành cho những tay mơ mà thôi. Nhưng tôi khuyên bạn rằng bạn hãy tập làm việc với GIAO DIỆN ĐỒ HOẠ. Bản thân tôi xuất thân từ thời dấu nhắc cổ lỗ, và có thể nói nếu như nhắm mắt tôi vẫn gõ được tất cả các ký tự trên phím nhanh hơn cả việc nhìn để gõ chúng. Tuy nhiên, trời phú cho bạn đôi mắt bạn hãy biết hưởng thụ khả năng của nó đừng nên bỏ qua! Và tôi cũng vậy!
Được vậy thì chúng ta hãy bắt tay vào việc sử dụng chương trình phpMyAdmin. Đây là chương trình có giao diện đồ hoạ hỗ trợ mọi truy cập trên CSDL MySQL. Nếu như bạn chưa có thì hãy chạy ra các cửa hàng ngoài Bùi Thị Xuân hay Tôn Thất Tùng kiếm ngay một đĩa đi. Không thôi thì download trên các website phpMyAdmin. Chương trình này rất dễ sử dụng. Nếu như bạn cảm thấy khó thì hãy xem phần Phụ Lục của quyển sách, tôi có hướng dẫn chi tiết trong đó.


phpMyAdmin - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit
Address http://192.168.1.1/mysqladmin/phpMyAdmin/index.php3
Go Links
Home
database_name
mysql
columns_priv
db
func
host
tables_priv
user
test
companies
contacts
mytable
mytable2
table1
test10
test3
test4
test5
test6
test8
topics
Welcome to phpMyAdmin 2.0.5
MySQL 3.22.27-log running on localhost
Create new database [Documentation]
Create
Show MySQL runtime information [Documentation]
Show MySQL system variables [Documentation]
Show processes [Documentation]
Reload MySQL [Documentation]
phpMyAdmin-Homepage
phpMyAdmin Documentation
Start
10:41 AM

phpMyAdmin


phpMyAdmin - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Forward Stop Refresh Home Search Favorites History Mail Print Edit
Address http://192.168.1.1/mysqladmin/phpMyAdmin/index.php3
Go Links
Home
database_name
mysql
columns_priv
db
func
host
tables_priv
user
test
companies
contacts
mytable
mytable2
table1
test10
test3
test4
test5
test6
test8
topics
Database test - table companies
Select fields (at least one):
company_id
company_name
company_address
Add search conditions (body of the "where" clause):
[Documentation]
Do a "query by example" (wildcard: "%")
Field Type Value
company_id int
company_name string
company_address string
Start
10:40 AM

Tóm tắt

Trong chương này bạn đã tìm hiểu tất cả các lệnh cần thiết để tạo và thao tác với CSDL trong môi trường MySQL. Bạn đã biết mục đích của index. Biết các kiểu dữ liệu và mục đích sử dụng theo tuỳ trường hợp. Ngoài ra tôi cũng giới thiệu một chương trình quản lý CSDL có giao diện đồ hoạ phpMyAdmin giúp bạn thao tác thủ công trên CSDL một cách dễ dàng. Trong phần III và IV bạn sẽ tìm hiểu kỹ hơn và thực hiện những ứng dụng quan trọng của CSDL trong chương trình. **(Còn tiếp)**

Lưu ý:
Bạn có thể tìm thấy các CD thiết kế web PHP tại các cửa hàng CD (Tôn Thất Tùng, tp.HCM chẳng hạn). Trên CD có trình cài đặt Apache, PHP, MySQL thông dụng là FOX. Ngoài ra còn có các Website PHP mẫu.

CÁC LỆNH Ở BÀI HỌC TRÊN TỐT NHẤT BẠN NÊN GÕ LẠI, KHÔNG NÊN COPY!

tongphuockhai@mail15.com

- Kiến thức là kho báu không phải của riêng ai. Vì vậy bạn đừng ngần ngại khi chia sẻ cho mọi người.

LẬP TRÌNH WEB ĐỘNG VỚI

PHP / MySQL

- GUESTBOOK
- CATALOG
- FORUM
- SHOPPING CART

PHẦN 3

Tống Phước Khải (tổng hợp & biên dịch)

BIẾN (variables)

và các phép xử lý trên biến

PHP

1- Biến và cách sử dụng Biến

2- Xử lý dữ liệu từ FORM

3- Tìm hiểu sâu hơn về Biến

Bạn đọc thân mến,

Vừa qua tôi đã soạn xong phần 1 và phần 2 của giáo trình tự học PHP/MySQL. Tôi đã nhận được email của những bạn quan tâm, chờ đợi phần 3 của giáo trình này. Đáng lẽ phần 3 đã cho ra đời sớm nhưng vì bận rộn quá nhiều công việc (hiện tôi đang phụ trách và có rất nhiều công việc trong nhóm Hanosoft - software Hán Nôm) nên việc biên soạn sách tự học này ít nhiều bị trì hoãn.

Do hoàn cảnh trên, chắc chắn công việc biên soạn này không tránh được những sai sót. Nhưng dù sao đi nữa, biết được các bạn có thể áp dụng giáo trình này vào thực tế thì tôi phần nào cũng lấy đó làm sự khích lệ cho riêng mình.

Đúng lý ra phần 3 này là dành trọn cho việc nói về ngôn ngữ SQL nhưng phần 2 đã bàn về CSDL rồi, nếu phần này nếu cũng bàn về nó thì có vẻ hơi nhàm phải không các bạn? Với ý nghĩ này tôi đã dành trọn phần 3 để nói về biến trong PHP.

Hẳn nhiên tôi biết một số bạn mong mỏi những gì trong đây có thể áp dụng liền thì đỡ chán hơn. Nhưng theo tôi nghĩ trước tiên hết bạn cần phải nắm rõ mọi ngóc ngách của PHP và MySQL thì mới có thể thiết kế được những chương trình đạt tiêu chuẩn. Do vậy mong các bạn hãy kiên nhẫn khi đọc những chương hướng dẫn suông như thế này! Đừng nản lòng và nên ghi nhớ đây là cội rễ cho các ứng dụng thực tế của các bạn.

Để bắt đầu chương này ít nhất bạn cũng phải có chút đỉnh kiến thức về **Cơ Sở Lập Trình**. Tôi nghĩ nếu bạn đã học qua một khoá lập trình căn bản thì bạn có thể hiểu được. Nếu không, đòi hỏi bạn cần phải động não hoặc tìm tòi hơi nhiều. Nào, chúng ta bắt đầu đi thôi!

PHP xử lý các biến rất linh động. Nó có thể nhận biết được kiểu của biến và làm cho cú pháp câu lệnh đơn giản hơn. Ai đã từng lập trình với C, Java hoặc Perl sẽ cảm thấy rất dễ dàng khi sử dụng PHP. Tuy nhiên việc dễ dãi này cũng gây ra một số trở ngại nhất định.

Tất cả những biến khai báo trong PHP đều được bắt đầu với dấu đô la ($). Dù cho biến của bạn kiểu chuỗi, nguyên hay thập phân hoặc thậm chí là mảng thì chúng không có gì khác biệt nhau. PHP chỉ theo dõi dữ liệu chứa trong biến thay đổi như thế nào thôi.

Nói chung, khi làm việc với PHP bạn sẽ quan tâm đến 3 vị trí khác nhau của biến đó là: (1) *khai báo ngay trong mã lệnh PHP*, (2) *chuyển tiếp từ một trang HTML* hoặc (3) là *biến sẵn có trong của hệ thống PHP*.

Chúng ta sẽ tìm hiểu về từng loại trên ở phần tiếp theo. Nên lưu ý rằng biến cũng có thể được chuyển tiếp từ các nơi khác như từ các URL hoặc từ các SESSION.

Gán biến trong một Script

Bạn không cần phải khai báo EXPLICIT cho biến như trong một số ngôn ngữ khác. Chỉ cần khai báo tên biến là nó sẽ sẵn sàng làm việc. Bạn hãy xét các ví dụ sau để hiểu cách khai báo biến trong PHP **uyển chuyển** như thế nào:

```
$a = "Toi thich hoc PHP"; //day la bien chuoi
$b = 4; //day la bien so
$c = 4.837; //day la bien so thuc
$d = "2"; //day cung la bien chuoi
```

Để ý rằng dấu = là dấu dùng để gán. Còn khi thực hiện phép so sánh bằng thì bạn dùng hai dấu bằng (= =). Ví dụ: IF ($x= =1)

PHP rất thông minh trong việc biến đổi kiểu. Ví dụ, bạn thực hiện phép cộng một số nguyên với một chuỗi chứa ký tự số (trong ví dụ trên là $b và $d).


```
$a = "Toi thich hoc PHP"; //day la bien chuoi
$b = 4; //day la bien so
$c = 4.837; //day la bien so thuc
$d = "2"; //day cung la bien chuoi
```

```
$e = $b + $d;
echo $e;
```

PHP sẽ nhận ra rằng bạn muốn xem chuỗi trong $d (chuỗi "2") như là một số nguyên. Thế là nó sẽ hoán chuyển sang trị nguyên và thực hiện phép toán cộng cho ra kết quả là $e = 6. Ngoài ra, PHP còn có thể hiểu được các chuỗi vừa số vừa chữ như ví dụ sau:

```
$a = 2;
$b = "2 con heo con";
$c = $a + $b;
```

Kết quả cho ra là $c = 4. Nếu một số nguyên hay thập phân đứng ở vị trí đầu một chuỗi thì PHP có thể hiểu được như ví dụ trên. Tương tự, PHP thực hiện tương tự đối với các kiểu số khác nhau:

```
$f = 2; //$f la mot so nguyen
$g = 1.444; // $g la mot so thuc
$f = $f + $g; //$f tu bien doi thanh so thuc
```

Việc xử lý này thật là hay, nhưng nó có thể dẫn đến một số rắc rối đó là sẽ có những lúc bạn không biết ở tại thời điểm nào bạn sẽ làm việc với kiểu của biến là kiểu gì. Tôi sẽ trình bày trong phần Kiểm Tra Biến.

Qui định về chuỗi

Trong các ví dụ trên, bạn thấy tất cả các chuỗi đều được bao trong dấu nháy đôi. Có hai cách khác để bạn thể hiện một cho PHP hiểu đó là chuỗi.
Trong một chuỗi mà bạn đã bao lại bằng cặp nháy đôi "...", xong bạn chèn một biến vào giữa, thì PHP vẫn hiểu được biến đó. Ví dụ:

```
$my_name = "Jay";
$phrase = "Hello, my name is, $my_name";
echo $phrase;
```

Kết quả cho ra là: Hello, my name is, Jay. Thật khác thường phải không các bạn?! (Đáng lẽ ra dấu nháy " thứ hai phải sau chữ *is* rồi đặt một dấu cộng chuỗi với biến *$my_name*)

Trong trường hợp sau đây, tôi muốn xuất ra một chuỗi: *Tôi đăng ký hosting hết $20* thì phải làm sao? Bởi vì trong chuỗi này có chứa $, điều này sẽ làm cho PHP hiểu đó là một biến mới. Chúng ta xem cách giải quyết như sau:
Nếu như trong chuỗi bạn muốn có chứa các ký tự đặc biệt như: dấu nháy đôi "", dấu slash \, dấu đô la $, bạn phải sử dụng đến ký tự chuyển (gọi là dấu escape) đó là dấu slash (\).
Tôi quen đọc dấu / là "dấu suyệt trái" và \ là "dấu suyệt phải".

Giả sử, để xuất ra màn hình một dòng chữ: *<form action="mypage.php" method="get">*, như bạn thấy trong đó chứa tới 4 dấu nháy đôi - thuộc dạng ký tự đặc biệt. Ta phải sử dụng tới 4 dấu suyệt phải như sau:

```
echo "<form action=\"mypage.php\" method=\"get\">";
```

Thì đến khi chạy chương trình mới mong cho ra kết quả như mong muốn.

Tác dụng của dấu nháy đơn đối với PHP:
Bạn sẽ thấy dấu nháy đơn trong PHP có tác dụng hơn dấu nháy đôi như thế nào! Nếu chuỗi của bạn có chứa các biến (bắt đầu bằng $), bạn bao lại bằng dấu nháy đơn thì biến đó sẽ bị biến thành chuỗi luôn, chớ không được hiểu là một biến như cách bao bằng dấu nháy đôi:

```
$my_name = "Jay";
echo 'Hello, my name is, $my_name';
```

Kết quả cho ra là *Hello, my name is, $my_name* chứ không phải Hello, my name is Jay.

Cuối cùng, trong PHP4 bạn có thể sử dụng dấu Here Documents. Đây là một loại ký hiệu tương tợ hai loại nháy đơn và nháy đôi. Trong một số trường hợp khi sử dụng nó bạn sẽ cảm thấy rất tiện lợi. Here Docs xác định giới hạn ở đầu chuỗi với 3 dấu nhỏ hơn <<< và ký hiệu nhận dạng (trong sách này tôi sử dụng ký hiệu nhận dạng EOQ) Chuỗi được kết thúc cũng với ký hiệu nhận dạng như vậy và kèm theo là dấu chấm phẩy (;). Sau đây là ví dụ chuỗi *Toi thich hoc PHP* được gán cho biến $mystring được xác định bằng cách sử dụng Here Doc.

```
$my_string = <<<EOQ
Toi thich hoc PHP.
EOQ;
```

Sử dụng Here Doc, các biến sẽ chỉ ảnh hưởng trong chuỗi cho nên khi thể hiện dấu nháy đôi trong chuỗi thì không cần sử dựng dấu escape.

```
$element = <<<EOQ
<textarea name="$name" cols="$cols" rows="$rows"
wrap="$wrap">$value</textarea>
EOQ;
```

Như ví dụ trên các bạn thấy không cần phải hao phí nhiều dấu suyệt (\), chúng ta vẫn có thể có được một chuỗi chứa các ký hiệu dạng biến không có tầm ảnh hưởng ra bên ngoài.

Các phần tử mảng sử dụng khoá liên hợp (bạn sẽ tìm hiểu ở phần tiếp theo) không thể sử dụng Here Doc được. Ví dụ sau đây sẽ xuất hiện lỗi:

```
$array = array ("fname"=>"jay", "lname"=>"greenspan");
$str = <<<EOQ
print my string $array["fname"]
EOQ;
```

Mảng (array) trong PHP

Mảng là một dạng của biến trong đó có chứa nhiều giá trị. Ví dụ một dạng đơn giản của mảng là tháng:

```
$thang = array("Gieng", "Hai", "Ba", "Bon", "Nam",
"Sau", "Bay", "Tam", "Chin", "Muoi", "Muoi Mot", "Muoi Hai");
```

Mảng này có chứa 12 phần tử, và bạn có thể định vị chúng bằng thứ tự ở trong mảng, bắt đầu bằng vị trí 0. Do đó lệnh echo $thang[0] sẽ cho ra là Gieng và echo $thang[11] sẽ cho ra Muoi Hai. Để truy xuất được tất cả các phần tử trong mảng, bạn có thể tính ra chiều dài của mảng và thực hiện vòng lặp:

```
for ($i=0; $i<count($months); $i++)
{
echo $thang[$i] . "<br>\n" ;
}
```

Chi tiết về vòng lặp sẽ được trình bày ở các phần sau. Bạn có thể gán giá trị vào mảng với một phép toán đơn giản như sau:

```
$dogs = array();
$dogs[0] = "kiki";
$dogs[1] = "lulu";
```

Nếu bạn không xác định chỉ số bên trong ngoặc vuông thì giá trị sẽ được gán cho phần tử cuối mảng. Trong ví dụ sau "nana sẽ được gán vào $dogs[2]:

```
$dogs[] = "nana";
```

Mảng liên hợp

Cũng giống như các ngôn ngữ khác, PHP tận dụng khả năng của mảng liên hợp (associative array). Có thể bạn cảm thấy mới mẻ với khái niệm này. Để tôi nói sơ qua một chút: Mỗi phần tử trong mảng liên hợp mang khoá(key) riêng. Các phần tử của mảng sẽ được truy cập thông qua khoá. Điều này giống như cách thức truy xuất trong các query khi làm việc với Database. Trong ví dụ sau, bạn sẽ thấy các phần tử first_name, last_name, e-mail sử dụng các key:

```
$person = array (
"first_name" => "Jay",
```

```
"last_name" => "Greenspan",
"e-mail" => "jgreen_1@yahoo.com"
);
```

Nếu như bạn muốn thêm phần tử vào mảng, bạn có thể gán tiếp một giá trị khác. Dòng lệnh sau sẽ thêm một số nguyên vào trong mảng, do đó mảng này sẽ chứa tất cả 4 phần tử.

```
$person["age"] = 32;
```

Nếu bạn muốn truy cập cả khoá và giá trị của một mảng liên hợp, bạn sẽ dùng **list() = each()** như sau:

```
while (list($key, $value) = each($person))
{
echo "<b>key :</b> $key, value = $value <br>\n";
}
```

Các chương sau này tôi sẽ nói kỹ về `list() = each()` một cách chi tiết hơn. Trên cơ bản `each()` truy xuất được cả khoá và giá trị của phần tử trong mảng. List() giữ các giá trị

và gán vào $key và $value. Tiến trình này tiếp tục cho đến khi mỗi phần tử trong mảng được truy cập. Nếu bạn muốn duyệt qua hết mảng bạn cần phải sử dụng reset($person).

Nếu bạn chỉ muốn sử dụng giá trị của phần tử trong mảng mà thôi hoặc bạn muốn sử dụng mảng không liên lợp và vẫn muốn sử dụng cấu trúc list()=each() bạn phải thực hiện như sau:

```
while (list( , $value) = each($person))
{
echo "value = $value <br>\n";
}
```

Hoặc bạn chỉ muốn truy xuất khoá, bạn sẽ làm như sau:

```
while (list($key) = each($person))
{
echo "key = $key <br>\n";
}
```

Hãy nhận định về mảng trong PHP như sau:
- Tất cả các mảng trong PHP đều là mảng liên hợp. Tại vì sao? Bởi vì những mảng không phải là liên hợp thì PHP cũng sẽ tự động gán cho chúng các key. Ví dụ: $x= array ("pug", "poodle"), PHP sẽ tự gán cho $x các khoá là các con số nguyên theo thứ tự bắt đầu từ số 0. Bạn sẽ được tìm hiểu kỹ ở chương 6.

Mảng đa chiều

PHP cũng hỗ trợ mảng đa chiều. Mảng đa chiều thường sử dụng nhất đó là mảng hai chiều. Chúng chứa thông tin dựa trên hai khoá. Giả sử, nếu chúng ta chứa thông tin hai người trở lên thì mảng hai chiều sẽ hỗ trợ việc này rất tốt. Chúng ta sẽ xác lập một mảng $people. Trong mảng $people lại chứa mảng cho từng cá nhân:

```
$people = array (
"khai" => array ("ho_lot" => "tongphuoc", "tuoi" => 30),
"minh" => array ("ho_lot" => "leanh" , "tuoi" => 52)
);
```

Ta thấy $people chứa các thông tin của 2 người, Khai và Minh. Để truy cập một trị trong bất kỳ thông tin của cá nhân nào bạn sẽ phải dùng cả hai khoá. Ví dụ để truy xuất tuổi của Minh bạn sẽ thực hiện lệnh như sau:

```
echo $people["minh"]["tuoi"];
```

Bạn có thể truy cập tất cả các phần tử trong mảng hai chiều bằng cách sử dụng vòng lặp trên cả hai chiều của mảng:

```
while(list($person, $person_array) = each($people))
{
echo "<b>Ban biet gi ve $person</b><br>\n";
while(list($person_attribute, $value) = each($person_array))
{
echo "$person_attribute = $value<br>\n";
}
}
```

Biến gán từ trình duyệt (web browser)

Quan điểm chung của việc sử dụng PHP cũng như các ngôn ngữ khác là cung cấp khả năng nhập thông tin theo ý muốn của khách. Thông thường các thông tin này được nhập vào thông qua một form HTML. Nhưng cũng có thể chúng xuất phát từ các nguồn khác như: HTML, cookie, session.

Biến từ Form của HTML

Dạng thông thường nhất để khách có thể nhập thông tin riêng là thông qua một form HTML. Trong phần phụ lục A có trình bày chi tiết về các tạo một form HTML. Nếu bạn chưa biết gì về cách tạo form này thì hãy đọc phần phụ lục. Bạn hãy tạo trang sign.php chỉ chứa 100% mã lệnh HTML như sau (có thể đặt là sign.htm cũng được):

```
<form action=mypage.php action=post>
<input type=text name=email>
<input type=text name=first_name>
<input type=submit name=submit value=OK>
<input type=submit name=reset value=Cancel> </form>
```

Một khi khách nhấp chuột vào nút SUBMIT (chấp nhận) thì các biến như $email, $first_name, và $submit sẽ được chuyển giao sang trang action là mypage.php. Sau đó, trong trang mypage.php bạn sẽ xử lý các biến này tuỳ thuộc vào mục đích chương trình. Để ý rằng phần lớn các ứng dụng trong sách này đều sử dụng giá trị của nút lệnh SUBMIT.

Trong trang mypage.php bạn phải viết các lệnh để xử các thao tác của người truy cập. Bạn hãy xem cách xử lý trong trang mypage.php mẫu như sau:

```
<?php
if (isset($submit) && $submit=="OK")
{
echo "Cam on ban da gui thong tin cho chung toi.";
} else {
?>       <form action=mypage.php action=post>
         <input type=text name=email>
         <input type=text name=first_name>
         <input type=submit name=submit value=OK>
         <input type=submit name=reset value=Cancel> </form>

         <?php
         }
?>
```

Bạn hãy xem kỹ ví dụ trên, nếu như người truy cập nhập đủ thông tin và nhấn nút **OK** từ trang sign.php (chứa toàn mã lệnh html), thì nó chuyển sang trang mypage.php và xuất ra dòng thông báo: Cam on ban da gui thong tin cho chung toi. Ngược lại, nếu như nhấn nút **Cancel** thì nó sẽ thực hiện mã lệnh trong lệnh Else và sẽ hiển thị form để buộc nhập lại.

Chú ý: Bạn hãy xem lại cách thức submit trong ví dụ GuestBook ở tập một. Trong tập 1, nếu bạn không chọn Submit thì chương trình sẽ gọi lại trang sign.php là trang chứa Form nhập liệu

bằng lệnh include. Còn ở đây không gọi lại trang sign.php nữa, bởi vì chúng ta làm theo kiểu khác là gắn Form nhập liệu ngay trong file Action là mypage.php.

Các biến cũng có thể được truy xuất thông qua mảng $HTTP_POST_VARS hoặc $HTTP_GET_VARS, dựa vào method sử dụng trong form của bạn. Việc này rất thuận tiện, nếu các biến từ các forms có thể mang cùng tên với biến trong script của bạn, hoặc nếu bạn có các biến chưa định nghĩa được chuyển giao thì bạn sẽ tìm được ở đó.

Bạn có thể truy cập bất kỳ phần tử riêng biệt nào như đã làm trong mảng liên hợp ($HTTP_POST_VARS["e-mail"]). Hoặc bạn có thể tạo vòng lặp duyệt qua tất cả các phần tử của mảng:

```
while (list($key, $value) = each($HTTP_POST_VARS))
{
echo "variable = $key value = $value <br>";
}
```

Truyền mảng

Có những trường hợp khi việc chuyển giao biến không thể thực hiện được. Ví dụ như khi bạn chọn cả hai giá trị cho cùng một biến. Việc này thường xảy ra khi làm việc với form có chứa listbox và có thể là bạn sẽ giữ phím Ctrl để chọn phần tử thứ 2 trong list. Ta giải quyết bằng cách sử dụng phép truyền mảng.

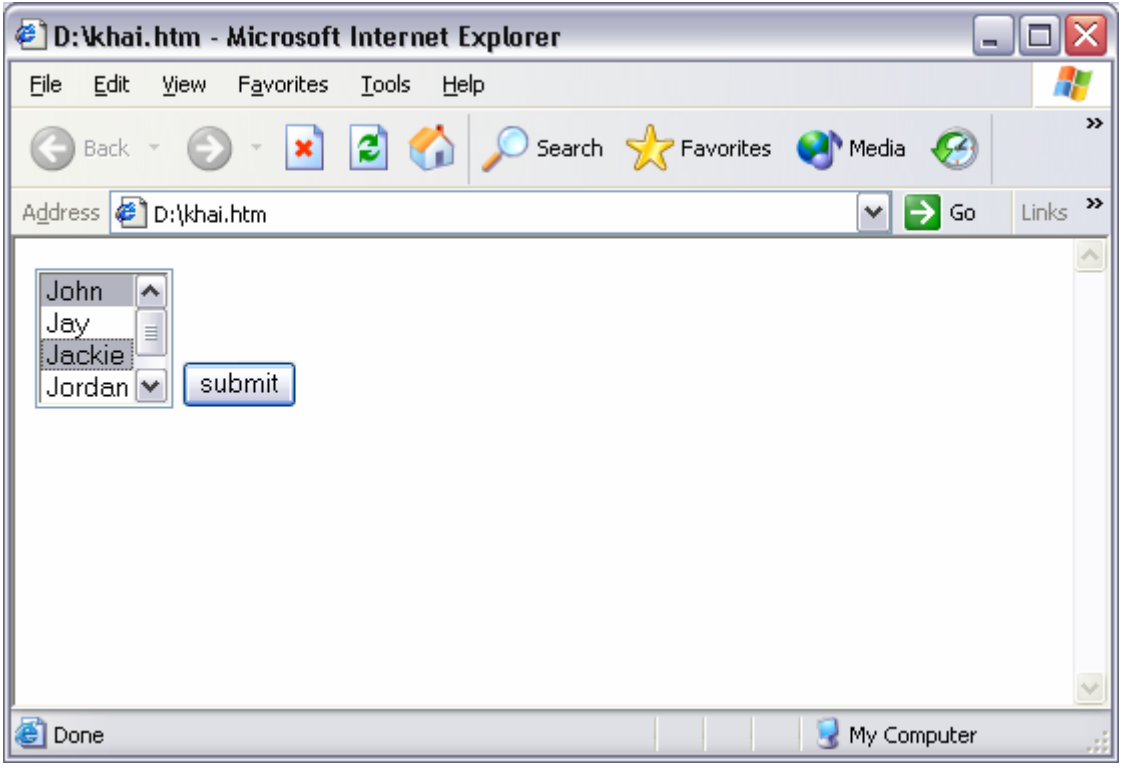
Các lệnh được sử dụng như sau:

```
<form action ="mypage.php" method="post">
<select name="j_names[]" size=4 multiple>
<option value="2">John
<option value="3">Jay
<option value="4">Jackie
<option value="5">Jordan
<option value="6">Julia
</select>
<input type="submit" value="submit">
</form>
```

Để ý rằng trong trong lệnh câu lệnh select name, tôi sử dụng dấu ngoặc vuông để bảo PHP biết rằng đây là một mảng. Nếu không sử dụng dấu [ ] thì sẽ có tới 2 giá trị gán cho cùng một biến.

Một khi được SUBMIT bạn có thể truy cập mảng như sau:

```
if (is_array($j_names))
{
echo "<b>the select values are:<br> <br>";
while(list($key, $value) = each($j_names))
{
echo $value . "<br>\n";
}
}
```


D:\khai.htm - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address D:\khai.htm
Go
Links
John
Jay
Jackie
Jordan
submit
Done
My Computer

Việc truyền mảng rất thông dụng khi bạn Submit Form có một loạt các checkbox (tên các mặt hàng chẳng hạn). Khách truy cập có thể sẽ nhấp chuột vào nhiều checkbox hoặc không có checkbox nào. Trong chương 8, có ví dụ cho phép người quản trị có thể sử dụng checkbox để chọn và xoá các phần tử.
Mảng được chuyển giao từ form có thể có khoá liên hợp, ngay cả đối với mảng nhiều chiều. Tên của phần tử form thường có tên là name = "array_name[element_name]". Hoặc đối với mảng nhiều chiều là array_name[element_name] [subelement_name]".

Cookies

Cookie là những **file nhỏ** chứa một số các thông tin truy cập Web. Các cookie do Websever phát sinh, lưu giữ lại, sau đó sẽ được đọc ở những lần truy cập về sau.
Cookie đơn thuần chỉ là *thông tin ghi nhận lại những động tác truy cập web của khách*. Khi tồn tại trên đĩa cứng, cookie trở thành các thỉnh cầu của giao thức HTTP, được gởi đến Webserver.
Để có thể phát sinh một cookie bạn cần phải sử dụng hàm setcookie() như sau:

```
setcookie(name, value, time_to_expire, path, domain, security
setting);
```

Chúng ta sẽ tìm hiểu chi tiết về cookie ở chương 6. Còn bây giờ bạn tìm hiểu sơ lược các chức năng thông qua một ví dụ:

```
setcookie("my_cookie",
"my_id",time()+(60*60*24*30),"/",".mydomain.com", 0)
```

Lệnh trên sẽ phát sinh một cookie với các chức năng sau:
- Chứa một biến tên là my_cookie
- Giá trị của mycookie my_id
- Cookie tồn tại trong vòng 30 ngày kể từ ngày nó phát sinh (`time()+(30*24*60*60)` ngày giờ hiện tại + 30 ngày được quy ra giây).
- Cookie có tác dụng đến tất cả các trang trong domain. Bạn có thể hạn chế lại bằng các chỉ ra đường dẫn đến một số trang nào đó trong domain.
- Nó sẽ hiện diện trong tất cả các website có địa chỉ http://mydomain.com
- Không có xác lập đặc biệt nào về bảo mật.

Một khi cookie được xác lập, các biến phát sinh từ cookie có tác dụng giống như biến phát sinh từ form mà chúng ta đã bàn trước đây. Chúng sẽ hiện diện với chức năng là biến global.

Sau khi script PHP xác lập cookie, các script khác trong domain có thể truy cập cookie một cách tự động.
Nếu như bạn muốn cẩn thận hơn để $mycookie không xung đột với một biến nào khác cũng có tên $mycookie, bạn có thể truy xuất nó thông qua mảng HTTP_COOKIE_VARS và sử dụng lệnh: HTTP_COOKIE_VARS["mycookie"].

Bạn có thể xác lập cookie cung cấp khả năng truy xuất như là một mảng:

```
setcookie("mycookie[first]","dddd",time()+2592000,"/","192.168.1.
1", 0);

setcookie("mycookie[second]","my_second_id",time()+2592000,"/","1
92.168.1.1", 0);
```

Cả hai biến trên đều có thể truy cập đến như là một mảng liên hợp.

Sessions

PHP4 cũng giống như ASP và ColdFusion đều có hỗ trợ session, việc này giúp ích rất nhiều cho việc truy cập web. Vậy session là gì?

Đơn giản nó chỉ là **một cách thức để duy trì và truyền biến** trong khi chuyển tiếp giữa các trang web. Chương trình của bạn khai báo một session được bắt đầu với hàm start_session(). PHP đăng ký một SesssionID duy nhất, và thường thì ID này được gởi đến user thông qua một cookie. PHP sau đó tạo một tập tin trên server để theo dõi sự thay đổi của biến. Tập tin này có tên giống như tên của SessionID.

Một khi session được tạo, bạn có thể đăng ký bất kỳ số lượng biến. Các giá trị của những biến này được lưu giữ trong tập tin trên server. Cũng như sự tồn tại của cookie, các biến trong session sẽ hiện diện trên bất kỳ trang nào được truy cập đến trong phạm vi một domain. Việc xác lập này rất thuận tiện hơn là chuyển tiếp các biến từ trang này sang trang khác thông qua các phần tử ẩn trong form hay cookie.

Session nói chung là khá đơn giản. Hãy xem script sau sẽ đăng ký một biến session tên là $my_var, và sẽ gán cho nó một giá trị là "hello world".

```
<?
session_start();
session_register("my_var");
$my_var = "hello world";
?>
```

Trên những trang tiếp theo biến $my_var sẽ hiện diện, nhưng chỉ sau khi bạn chạy hàm sesssion_start(). Hàm này bảo PHP tìm kiếm một session xem có tồn tại hay không, rồi làm cho các biến session trở thành global. Nó có thể sử dụng câu lệnh IF để làm cho các biến session hoàn toàn có thể truy cập được. Hãy xem xét ví dụ sau:

```
<?php
session_start();
session_register("your_name");
//check to see if $your name contains anything
if(!empty($your_name))
{
echo "I already know your name, $your_name";
}
//this portion will probaby run the first time to
//this page.
elseif(empty($your_name) && !isset($submit))
{
echo "<form name=myform method=post action=$PHP_SELF>
<input type=text name=first_name> first name<br>
<input type=text name=last_name> last name<br>
<input type=submit name=submit value=submit>
</form>";
//if the form has been submitted, this portion will
```

```
//run and make an assignment to $your_name.
} elseif (isset($submit) && empty($your_name))
{
$your_name = $first_name . " " . $last_name;
echo "Thank you, $your_name";
}
```

Sau khi chạy chương trình này, chọn **refresh** trên trình duyệt. Bạn sẽ thấy script sẽ nhớ được rằng bạn là ai.

Các hàm setcookie() và session_start() nên ở vị trí gần đầu tập tin. Nếu bạn thử chuyển đến trình duyệt trước để xác lập một cookie bạn sẽ nhận được một thông báo lỗi.

Biến sẵn có

Có rất nhiều biến sẵn có của PHP và Server. Bạn có thể liệt kê một danh sách đầy đủ bằng cách sử dụng lệnh phpinfo() để xem. Bạn hãy tạo một file php và cho chạy thử xem:

```
<?php
phpinfo();
?>
```

Bạn có thể sử dụng các biến này bằng nhiều cách thức khác nhau. Tôi sẽ trình bày một sau ngay sau đây, và sẽ chỉ ra bạn nên dùng vào trường hợp nào. Một số biến đến từ PHP engine, một số khác bắt nguồn từ Webserver.

Biến sẵn có của PHP

PHP_SELF

Biến này nhận giá trị là địa chỉ hiện tại của tập tin .php đang được duyệt. Địa chỉ này sẽ là địa chỉ đầy đủ từ gốc (bắt đầu từ http://) . Bạn sẽ sử dụng nó khi muốn truy cập lại chính trang web đang thi thành.

Xét ví dụ sau, đây là một form tương tự như form sign.php mà các bạn đã có dịp xét qua. Nếu khách thực hiện thao tác khác với submit thì chính form này sẽ được thi hành lại:

```
<?
if(isset($submit))
{
//Xuat ra thong bao tai day
echo "Cam on ban da submit";
} else {
?>
<form name=myform method=post action=<?=$PHP_SELF?>>
<input type=text name=first_name> first name<br>
<input type=text name=last_name> last name<br>
<input type=submit name=submit value=submit>
```

```
</form>
<?
}
?>
```

HTTP_POST_VARS

Đây là một mảng chứa tất cả các biến được chuyển tiếp thông qua POST method từ một form. Bạn có thể truy cập từng biến riêng rẽ như là một phần tử của mảng liên hợp (ví dụ: $PHP_POST_VARS["myname"]).

HTTP_GET_VARS

Đây là một mảng chứa tất cả các biến được chuyển tiếp thông qua GET method. Bạn có thể truy cập từng biến riêng rẽ như là một phần tử của mảng liên hợp (ví dụ: $PHP_GET_VARS["myname"]).

HTTP_COOKIE_VARS

Tất cả các cookie chuyển đến trình duyệt đều có thể được truy xuất trong mảng liên hợp này. Nó bao gồm cả session cookie. Nếu bạn còn thắc mắc cookie sẽ thi hành như thế nào thì hãy xem hàm phpinfo() để biết được trình duyệt của bạn đang chuyển đến server những gì.

BIẾN CỦA APACHE

Apache có sẵn rất nhiều biến. Tôi không trình bày đầy đủ tất cả các biến ra đây. Các biến bạn sử dụng, chúng tuỳ thuộc vào xác lập hiện tại của bạn như thế nào. Sau đây là một số biến mà có lẽ bạn sẽ sử dụng thường xuyên trong chương trình của bạn.

DOCUMENT_ROOT

Biến này trả về đường dẫn của Webserver. Biến này được tôi sử dụng trong xuyên suốt quyển sách này. Hãy xét ví dụ sau:

```
include"$DOCUMENT_ROOT/book/functions/charset.php";
```

Bằng cách sử dụng biến DOCUMENT_ROOT thay vì dùng đường dẫn tuyệt đối, chúng ta có thể di chuyển toàn bộ một thư mục sang một Apache Server khác mà không lo lắng rằng đường dẫn sẽ bị sai lệch trong include path. Nên nhớ rằng nếu như bạn không sử dụng Apache Server thì biến này không sử dụng được. Nếu bạn sử dụng include_path trong tập tin php.ini,

bạn không cần phải lo lắng phải xác định đường dẫn như thế nào bởi vì PHP sẽ duyệt hết tất cả các thư mục và tìm ra tập tin bạn đã chỉ định.

HTTP_USER_AGENT

Bất kỳ ai đã từng thiết kết Web site đều hiểu rằng tầm quan trọng của việc nhận dạng được trình duyệt của người sử dụng là gì. Một số trình duyệt thì không sử dụng được JavaScript, một số khác thì đòi hỏi dạng HTML đơn giản. Biến user_agent cung cấp cho bạn khả năng uyển chuyển đối với từng trình duyệt khác nhau. Một user_agent chuẩn có dạng như thế này:

```
Mozilla/4.0 (compatible; MSIE 5.01; Windows 98)
```

Nếu bạn phân tích chuỗi này ra bạn sẽ biết được những gì bạn cần tìm. Có thể bạn chỉ thích hàm get_browser() của PHP. Về lý thuyết mà nói, hàm này định nghĩa khả năng cho phép của trình duyệt của user đang sử dụng. Cho nên bạn có thể biết được là chương trình của bạn đang phục vụ tốt hay không. Các sách PHP có những hướng dẫn về cách cài đặt và sử dụng get_browser(), nhưng tôi khuyên bạn không nên sử dụng nó. Bởi vì sử dụng get_browse bạn sẽ được bảo rằng IE 5 dùng cho PC và Netscape 4.01 dùng cho Mac có hỗ trợ CSS (cascading stylesheets) và JavaScript. Nhưng bất kỳ người sử dụng nào cũng biết rằng: viết lệnh DHTML

để chạy trên cả hai môi trường trình duyệt này là một công việc phức tạp. Thông tin bạn nhận được từ get_browser() có thể dẫn đến những tính năng giả trong bảo mật. Cách tốt nhất là bạn sử dụng HTTP_USER_AGENT và thực hiện quyết định của mình dựa trên trình duyệt hoặc platform xác định nào đó.

REMOTE_ADDR

Dùng để lấy địa chỉ IP của user. Tuy nhiên có những user am hiểu chuyện này và có thể họ thay đổi IP của máy mình. Cho nên không lấy gì để đảm bảo rằng: một địa chỉ IP chắn chắn là của một user nào đó. Bạn sử dụng biến này để theo dõi sự truy nhập của một user nhưng nó chỉ mang tính tương đối thôi.

REQUEST_URI

Biến này cũng giống như biến PHP_SELF. Ngoài ra nó còn chứa thêm tham số trong địa chỉ truy vấn . Nếu bạn truy cập vào địa chỉ:

`http://www.mydomain.com/info/products/index.php?id=6`

Thì biến REQUEST_URI của bạn có giá trị là: `info/products/index.php?id=6`

SCRIPT_FILENAME

Biến này chứa toàn bộ đường dẫn của tập tin.

Kiểm tra biến

Ở trên chúng ta đã nói nhiều về Biến. Như các bạn biết đó, tên của một biến không quan trọng bằng giá trị mà nó chứa trong đó. Như tôi đã nói Biến trong PHP rất uyển chuyển. Điều này phát sinh sự bất lợi là bạn sẽ không biết ở tại một thời điểm nào đó thì biến này sẽ mang giá trị gì. Do đó bạn cần phải thực hiện thao tác kiểm tra biến.

isset( )

Hàm này thực hiện việc kiểm tra biến có chứa giá trị hay không. Nó sẽ trả về giá trị TRUE hoặc FALSE. Nếu biến chưa được xác lập thì trị isset() sẽ là false.

Bạn hãy xem xét ví dụ sau, nó thi hành một query MySQL. Bạn đã biết rằng một field trong database có thể chứa trị null hoặc chuỗi rỗng. Với việc sử dụng hàm isset() bạn sẽ kiểm tra và phân biệt được hai giá trị trên. Trong đoạn lệnh PHP bên dưới. Trong đó biến $query là một phát biểu SELECT lấy dữ liệu submit từ form của user.

```
$result = mysql_query($query) or
die (mysql_error());
$number_cols = mysql_num_fields($result);
```

```
echo "<b>query: $query</b><br>\n";
//layout table header
echo "<table border = 1>\n";
echo "<tr align=center>\n";
for ($i=0; $i<$number_cols; $i++)
{
echo "<th>", mysql_field_name($result, $i), "</th>\n";
}
echo "</tr>\n";//end table header
//layout table body
while ($row = mysql_fetch_row($result))
{
echo "<tr align=left>\n";
for ($i=0; $i<$number_cols; $i++)
{
echo "<td>";
if (!isset($row[$i])) //test for null value
{echo "NULL";}
else
{echo $row[$i];}
echo "</td>\n";
```

```
}
echo "</tr>\n";
}
echo "</table>";
```

Lưu ý rằng dấu chấm than (!) có nghĩa là phủ định.
Tức là nếu $var có giá trị null thì:
isset($var) cho ra giá trị False
!isset($var) cho ra giá trị True

empty()

Hàm empty() có vẻ ngược ngạo so với hàm isset(). Nó sẽ cho ra trị True nếu $var có trị null, chuỗi rỗng hoặc số 0. Hàm này thường được sử dụng để kiểm tra xem user có nhập trị vào trong form hay không:

```
if(empty($first_name))
{
echo "Ban can phai nhap ten cua minh";
exit;
```

```
}
```

is_int( )

Hàm này để kiểm tra biến có phải là số nguyên hay không. Có 2 cú pháp khác cho cùng kết quả như nó là: is_integer và is_long(). Bạn sử dụng hàm này khi không chắc rằng biến là một trị nguyên hay chuỗi. Ví dụ:

```
$a = "222";
$b = 22;
```

is_int($a) cho ra trị False
is_int($b) cho ra trị True

Tương tự bạn sẽ có một loạt hàm kiểm tra kiểu của biến sau đây:

is_double()

Kiểm tra số kiểu double (dấu phẩy động). Hàm thay thế: `is_float()` và `is_real()`.

is_string( )
Kiểm tra kiểu chuỗi.

is_array( )
Kiểm tra kiểu mảng.

is_bool( )
Kiểm tra kiểu boolean (TRUE và FALSE)

is_object( )
Kiểm tra biến kiểu object. Bạn sẽ tìm hiểu kiểu object trong các phần sau.

gettype( )
Hàm này sẽ cho bạn biết kiểu của biến như: string, double, integer, array, hoặc boolean. Ngoài ra nó có trả về các kiểu như object, class. Bạn sẽ khảo sát kỹ về việc lập trình hướng đối tượng trong các phần sau để biết thêm về object và class.

Lưu ý trị của hàn gettype() trả về luôn là một chuỗi: "string", "integer", "double" v.v.
Bạn hãy xem ví dụ sau:

```
$str = "Day la mot chuoi";
$type = gettype($str);
if ($type == "string")
{
echo "Dung vay";
}
```

Đổi kiểu của biến

Bạn sẽ sử dụng 3 cách để đổi kiểu của biến.

Phương pháp type casting

Phương pháp này rất đơn giản: Bạn chỉ cần ghi tên kiểu ra, đóng ngoặc đơn lại, rồi đặt trước biến. Tức khắc biến sẽ bị đổi theo kiểu mà bạn muốn.

Cách thức: (kiểu) $biến

Ví dụ:

```
$a = 1;
$b = (string)$a; //số 1 sẽ biến thành chuỗi 1
echo gettype($a), "<br>\n";
echo gettype($b), "<br>\n";
```

Kết quả cho ra là:

```
integer
string
```

Sử dụng hàm settype( )

Hàm này có 2 đối số. Thứ nhất là tên biến, thứ nhì là kiểu. Ưu điểm của nó là nó có thể cho ra kết quả FALSE nếu như việc hoán đổi không được.

Cách thức: settype($biến, "kiểu")

Ví dụ:

```
$a = 1;
settype($a, "string");
```

Sử dụng hàm intval( ), doubleval( ), và stringval( )

Phương pháp này thường để bạn áp dụng nhanh trong khi tính toán. Có lẽ nhìn tên hàm bạn cũng biết được chức năng của nó rồi. Hãy xét ví dụ sau:

```
$a = "43";//43 là kiểu chuỗi
$b = (intval($a) * 2);
```

Biến của biến

Nghe qua có vẻ lạ lạ, nhưng đây là một "độc chiêu" của PHP. Với cách thức này bạn sẽ lấy giá trị của một biến để hình thành tên của một biến mới.
Cách thức: $$biến
Ví dụ:

```
$a = ‘khai’;
$$a = ‘Chao moi nguoi’;
```

Bạn sẽ thấy trong ví dụ trên một biến mới được hình thành đó là $khai chứa giá trị là "Chao moi nguoi"

Xét thêm ví dụ sau, trong đó $tacgia là một mảng liên hợp.

```
<?
$tacgia = array ("ho"=>"Tong", "ten"=>"Khai");
while (list($field,$value) = each($tacgia))
```

```
{
$field = "bien_$field";
$$field = $value;
}
echo $bien_ho, " ", $bien_ten;
?>
```

Khi chạy chương trình, các biến sau sẽ được tạo `$bien_ho, $bien_ten` và ghi ra màn hình: Tong Khai

Tóm tắt

Bạn đã tìm hiểu các biến trong PHP. Bạn thấy PHP xử lý các biến linh hoạt hơn nhiều so với các ngôn ngữ khác. Còn một vấn đề khá quan trọng đối với biến đó là scope bạn cũng sẽ biết kỹ về nó ở trong các phần sau của giáo trình này.

(Còn tiếp)

ỨNG DỤNG CƠ SỞ DỮ LIỆU TRÊN WEB VỚI PHP VÀ MYSQL

Giảng viên: Hoàng Văn Hiệp
Bộ môn Kỹ thuật Máy tính
Khoa CNTT – ĐH Bách Khoa Hà Nội

NỘI DUNG

- Giới thiệu về MySQL
- Lập trình CSDL với PHP và MySQL

CÁC KIẾN THỨC CƠ BẢN VỀ CSDL

- Cơ sở dữ liệu
  - Bảng
  - Trường (các cột)
  - Bản ghi (các dòng)
  - Khóa (key)
- Truy vấn
  - Thực hiện thao tác: tạo mới, cập nhật, xóa,…
  - Trả về kết quả

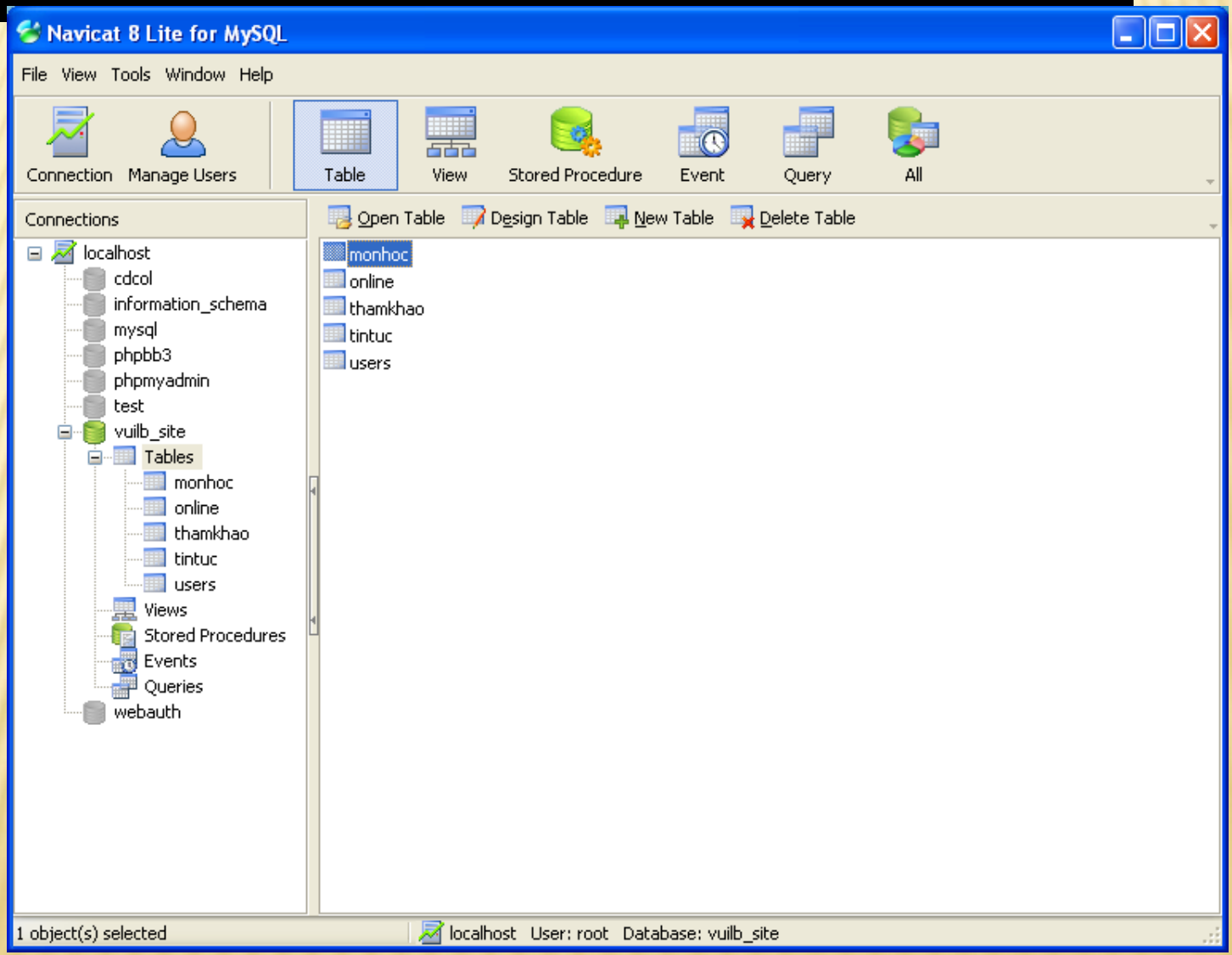
GIỚI THIỆU VỀ MYSQL

- Là hệ quản trị CSDL thích hợp với PHP
- Được cung cấp hoàn toàn miễn phí
- Có tốc độ thực hiện nhanh
- Chạy được trên nhiều nền hệ điều hành Windows, Unix, Linux
- Được cải thiện liên tục

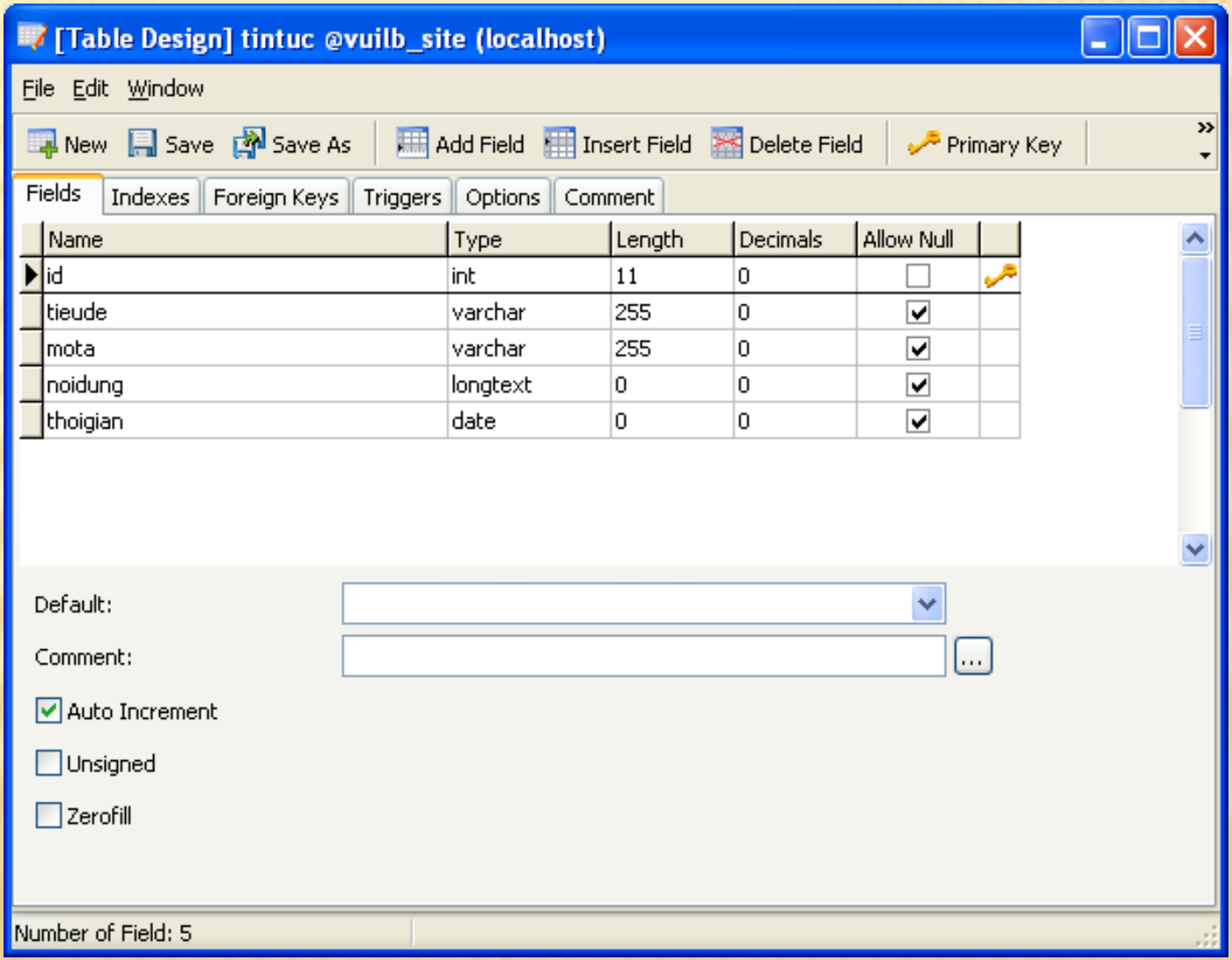
CÀI ĐẶT VÀ SỬ DỤNG

- Có thể cài đặt riêng hoặc cài đặt cùng gói phần mềm XAMPP.
- Sử dụng
  - Thao tác cửa sổ console, sử dụng các lệnh
  - Sử dụng các phần mềm hỗ trợ

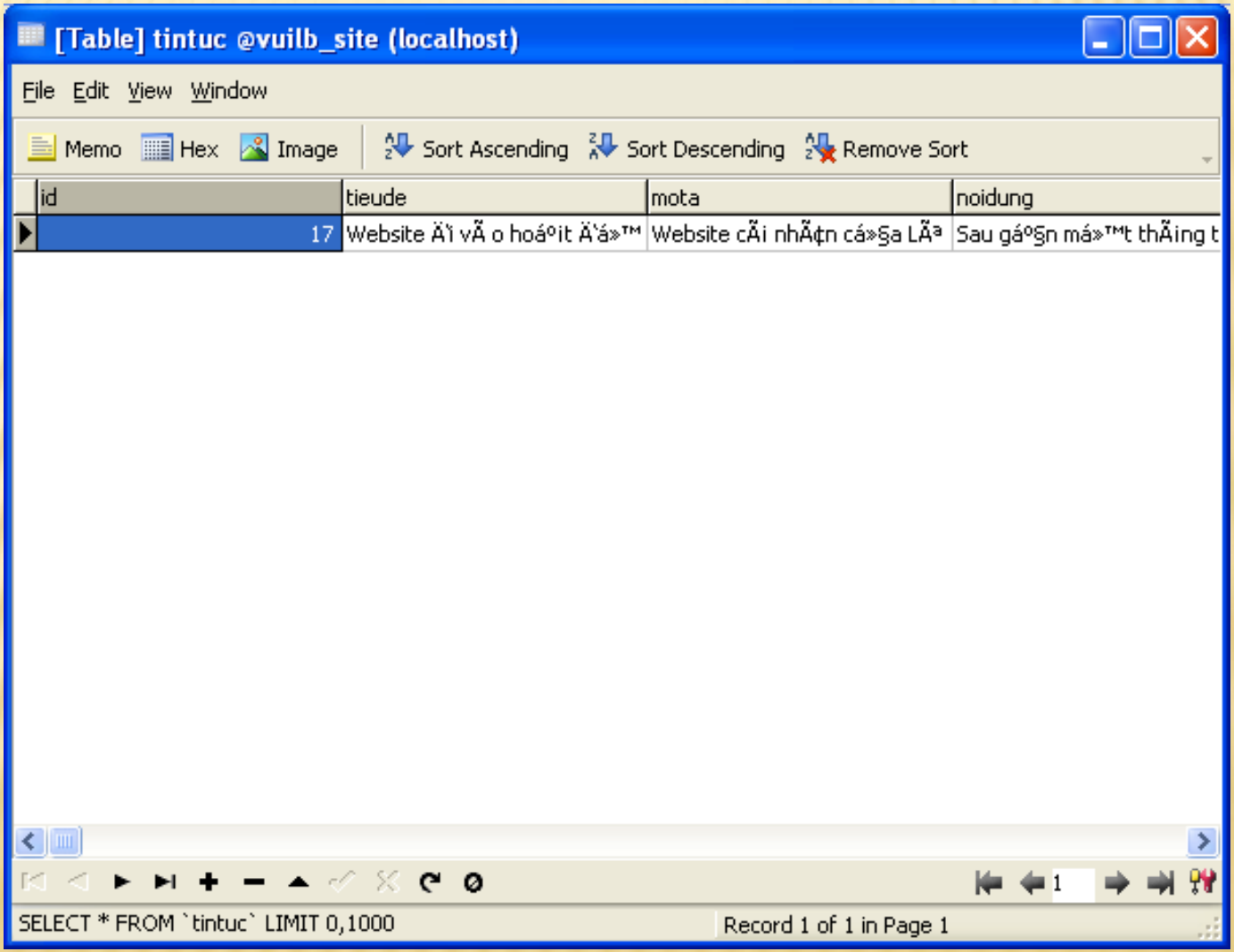
SỬ DỤNG PHẦN MỀM NAVICAT



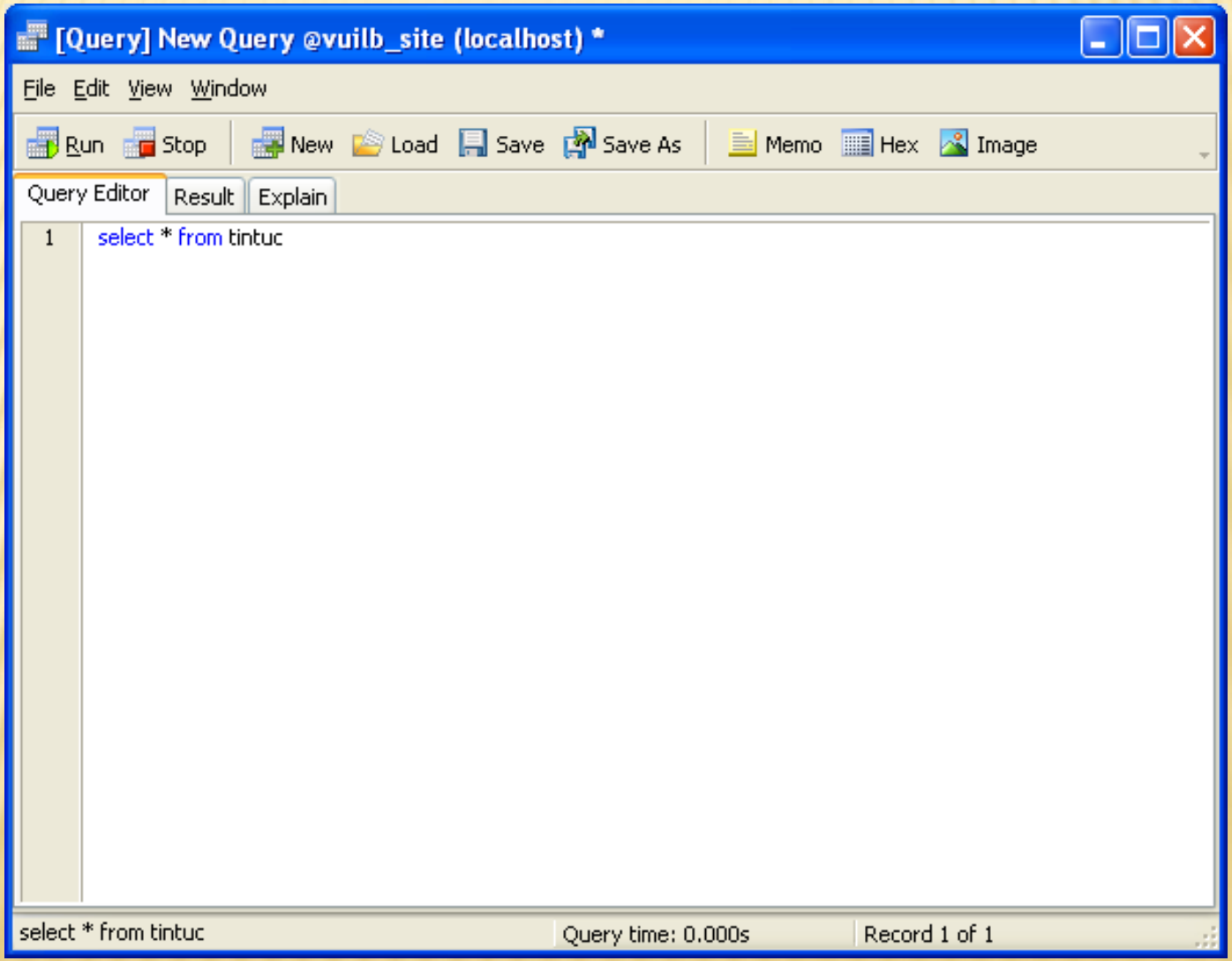
THIẾT KẾ BẢNG



THAO TÁC DỮ LIỆU



TẠO TRUY VẤN


[Query] New Query @vuilb_site (localhost) *
File Edit View Window
Run Stop New Load Save Save As Memo Hex Image
Query Editor Result Explain
1 select * from tintuc
select * from tintuc
Query time: 0.000s
Record 1 of 1

CÁC KIỂU DỮ LIỆU ĐỐI VỚI MYSQL

- Kiểu chuỗi ký tự: có thể chứa các ký tự, các chữ số và các ký tự đặc biệt
- Kiểu số
- Kiểu thời gian

DỮ LIỆU KIỂU VĂN BẢN

Kiểu dữ liệu	Mô tả
CHAR(size)	Chuỗi có độ dài cố định, tham số size chỉ ra độ dài của chuỗi, có giá trị lớn nhất là 255
VARCHAR(size)	Chuỗi có độ dài thay đổi, tham số size chỉ ra độ dài lớn nhất của chuỗi, có giá trị lớn nhất là 255
TINYTEXT	Chuỗi có độ dài lớn nhất là 255 ký tự
TEXT	Chuỗi có độ dài lớn nhất là 65,535 ký tự
MEDIUMTEXT	Chuỗi có độ dài lớn nhất 16,777,215 ký tự
LONGTEXT	Chuỗi có độ dài lớn nhất 4,294,967,295 ký tự

DỮ LIỆU KIỂU SỐ

Kiểu dữ liệu	Mô tả
TINYINT	Kiểu nguyên có giá trị từ -128 đến 127; từ 0 đến 255 đối với số không dấu
SMALLINT	Kiểu nguyên có giá trị từ -32,768 đến 32,767; từ 0 đến 65,535 đối với số không dấu
MEDIUMINT	Kiểu nguyên có giá trị từ -8,388,608 đến 8,388,607; từ 0 đến 16,777,215 đối với số không dấu
INT	Kiểu nguyên có giá trị từ -2,147,483,648 đến 2,147,483,647; từ 0 đến 4,294,967,295 đối với số không dấu
FLOAT	Số thực độ chính xác đơn
DOUBLE	Số thực độ chính xác kép

KIỂU DỮ LIỆU THỜI GIAN

Kiểu dữ liệu	Mô tả
DATE	Ngày tháng, định dạng YYYY-MM-DD
DATETIME	Kết hợp ngày tháng và thời gian, định dạng YYYY-MM-DD HH:MM:SS
TIME	Thời gian, định dạng HH:MM:SS
YEAR	Năm, định dạng YYYY

THAO TÁC CSDL VỚI PHP VÀ MYSQL

- Kết nối với máy chủ CSDL
- Chọn CSDL
- Thực hiện truy vấn
- Xử lý kết quả
- Đóng kết nối

VÍ DỤ

```
<?php
    $con = mysql_connect("localhost", "root", "");
    if (!$con)
        die("Không kết nối được với máy chủ CSDL!");

    if (!mysql_select_db("vuilb_site", $con))
        die("Không kết nối được với CSDL!");

    $sql = "select * from tintuc order by id desc";
    $result = mysql_query($sql);

    while ($row = mysql_fetch_array($result))
    {
        echo $row["thoigian"] . "<br>";
        echo "<h1>" . $row["tieude"] . "</h1>";
        echo $row["mota"] . "<br>";
    }

    mysql_close($con);
?>
```

Kết nối

Chọn CSDL

Truy vấn

Xử lý kết quả

Đóng kết nối

KẾT NỐI MÁY CHỦ CSDL

- Cú pháp

  mysql_connect(servername, username, password)

- Các tham số
  - ***servername***: tên của máy chủ CSDL cần kết nối, mặc định là "localhost:3306"
  - ***username***: tên sử dụng để truy nhập vào máy chủ CSDL
  - ***password***: mật khẩu truy nhập
- Giá trị trả về
  - giá trị kết nối thành công tới máy chủ CSDL (giá trị khác 0)
  - trong trường hợp không kết nối được, trả về giá trị 0 (false)

ĐÓNG KẾT NỐI

- Cú pháp

  mysql_close(connection)
- Tham số
  - ***connection***: kết nối đã được thiết lập bằng lệnh **mysql_connect()** trước đó
- Chú ý:
  - Việc sử dụng lệnh **mysql_close()** không thực sự cần thiết đối với các kết nối tạm thời, các kết nối tự động được đóng khi hết đoạn kịch bản
  - Tăng tính chặt chẽ của chương trình, yêu cầu có **mysql_close()**

VÍ DỤ

```php
<?php
   $con = mysql_connect("localhost","root","");
   if (!$con)  {
       die('Could not connect: ' . mysql_error());
   }

   echo "Truy nhap thanh cong";

   // some code

   mysql_close($con);
?>
```

LỆNH DIE()

- Cú pháp

  die("chuỗi_ký_tự")

- Tác dụng
  - Hiển thị thông báo và ngừng thực hiện các kịch bản sau đó
  - Thường dùng để thông báo lỗi

CHỌN CSDL

- Cú pháp

  mysql_select_db(database, connection)

- Tham số
  - **database**: cơ sở dữ liệu cần thao tác
  - **connection**: kết nối đã được thiết lập
- Giá trị trả về
  - Trả về giá trị 0 (false) nếu có lỗi

VÍ DỤ

```
<?php
  $con = mysql_connect("localhost", "root", "");
  if (!$con)
     die('Could not connect: ' . mysql_error());
  echo "Truy nhap thanh cong <br>";

  $selected_db = mysql_select_db("vuilb_site", $con);
  if (!$selected_db)
     die('Could not select: ' . mysql_error());
  echo "Da chon CSDL<br>";
  // some code

  mysql_close($con);
?>
```

THỰC HIỆN TRUY VẤN

- Select Query và Executed Query
- Cú pháp

  mysql_query(query, connection)
- Tham số
  - **query:** câu lệnh truy vấn
  - **connection:** kết nối đã được thiết lập
- Giá trị trả về
  - Trả về giá trị 0 (false) nếu có lỗi

TẠO CƠ SỞ DỮ LIỆU

- Lệnh SQL

```
CREATE DATABASE database_name
```

VÍ DỤ

```
<?php
   $con = mysql_connect("localhost", "root", "");
   if (!$con)
       die('Could not connect: ' . mysql_error());

   if (mysql_query("CREATE DATABASE my_db",$con))
       echo "Database created";
   else
       echo "Error creating database: " . mysql_error();

   mysql_close($con);
?>
```

XÓA CƠ SỞ DỮ LIỆU

- Lệnh SQL

```
DROP DATABASE database_name
```

VÍ DỤ

```
<?php
   $con = mysql_connect("localhost", "root", "");
   if (!$con)
       die('Could not connect: ' . mysql_error());

   if (mysql_query("DROP DATABASE my_db",$con))
       echo "Database deleted";
   else
       echo "Error deleting database: " . mysql_error();

   mysql_close($con);
?>
```

TẠO BẢNG

- Lệnh SQL

```
CREATE TABLE table_name
(
  column_name1 data_type NOT NULL,
  PRIMARY KEY (column_name),
  column_name2 data_type,
  column_name3 data_type,
  ....
)
```

VÍ DỤ

```
<?php
   $con = mysql_connect("localhost","root","");
   mysql_select_db("my_db", $con);

   $sql = "CREATE TABLE Persons (
        personID int NOT NULL AUTO_INCREMENT,
        PRIMARY KEY(personID),
        FirstName varchar(15),
        LastName varchar(15),
        Age int )";

   if (mysql_query($sql, $con))
        echo "Table created!";
   else
        echo "Error creating table: " . mysql_error();

   mysql_close($con);
?>
```

XÓA BẢNG

- Lệnh SQL

```
DROP TABLE table_name
```

XÓA BẢNG

```
<?php
  $con = mysql_connect("localhost", "root", "");
  if (!$con)
      die('Could not connect: ' . mysql_error());

  if (mysql_query("DROP TABLE Persons",$con))
      echo "Table deleted";
  else
      echo "Error deleting table: " . mysql_error();

  mysql_close($con);
?>
```

THÊM BẢN GHI MỚI

- Lệnh SQL

INSERT INTO table_name

VALUES (value1, value2, value3,...)

- Hoặc

INSERT INTO table_name (column1, column2, column3,...)

VALUES (value1, value2, value3,...)

VÍ DỤ

```
<?php
   $con = mysql_connect("localhost","root","");
   mysql_select_db("my_db", $con);

   $result = mysql_query("INSERT INTO Persons (FirstName, LastName, Age)
                                VALUES ('Peter', 'Griffin', '35')");
   if ($result)
        echo "record inserted";
   else
         echo "Error insert new record: " . mysql_error();

   $sql = "INSERT INTO Persons VALUES ('1', 'Glenn', 'Quagmire', '33')";
   $result = mysql_query($sql);
   if ($result)
        echo "record inserted";
   else
        echo "Error insert new record: " . mysql_error();

   mysql_close($con);
?>
```

TẠO BẢN GHI TỪ FORM

- Thông tin nhận từ form được lưu vào CSDL
- Thông tin được đọc từ biến $_GET, $_POST hoặc $_REQUEST rồi được chèn vào CSDL

VÍ DỤ - TẠO FORM

```
<html>
<head>
<title>Nhap thong tin tu form</title>
</head>
<body>
<form name="form1" method="post" action="insertform.php">
Firstname: <input type="text" name="Firstname"> <br><br>
Lastname: <input type="text" name="Lastname"><br><br>
Age: <input type="text" name="Age"><br><br>
<input type="submit" name="Submit" value="Submit">
</form>
</body>
</html>
```

XỬ LÝ THÔNG TIN

```
<?php
   $con = mysql_connect("localhost","root","");
   mysql_select_db("my_db", $con);

   $FirstName = $_POST["FirstName"];
   $LastName = $_POST["LastName"];
   $Age = $_POST["Age"];

   $r = mysql_query("INSERT INTO Persons (FirstName, LastName, Age)
                     VALUES ('$FirstName', '$LastName', '$Age')");
   if ($r)
        echo "record inserted";
   else
         echo "Error insert new record: " . mysql_error();

   mysql_close($con);
?>
```

TRUY XUẤT DỮ LIỆU

- Sử dụng câu lệnh SQL

```
SELECT column_name(s)
FROM table_name
WHERE column_name operator value
ORDER BY column_name(s) ASC|DESC
```

XỬ LÝ KẾT QUẢ

- Lệnh ***mysql_fetch_array(data)***
  - Trả về 1 bản ghi trong tập kết quả dưới dạng mảng kết hợp, trả về giá trị 0 (false) nếu có lỗi hoặc không còn bản ghi nào nữa.
  - Tự động chuyển sang bản ghi tiếp theo.

VÍ DỤ

```
<?php
   $con = mysql_connect("localhost","root","");
   mysql_select_db("my_db", $con);

   $result = mysql_query("SELECT * FROM Persons ORDER BY age");

   while($row = mysql_fetch_array($result))
     {
     echo $row['FirstName'];
     echo " " . $row['LastName'];
     echo " " . $row['Age'];
     echo "<br />";
     }

   mysql_close($con);
?>
```

ĐẾM SỐ BẢN GHI

- Lệnh SQL

```
SELECT COUNT(*) AS alias
FROM table_name
WHERE column_name operator value
```

- Lệnh PHP

```
mysql_num_rows(data)
```

VÍ DỤ

```
<?php
  $con = mysql_connect("localhost","root","");
  mysql_select_db("my_db", $con);

  $result = mysql_query("SELECT COUNT(*) AS nums
  FROM Persons WHERE Age>20");

  $row = mysql_fetch_array($result);

  echo "So ban ghi: " . $row['nums'];

  mysql_close($con);
?>
```

VÍ DỤ

```
<?php
  $con = mysql_connect("localhost","root","");
  mysql_select_db("my_db", $con);

  $result = mysql_query("SELECT * FROM Persons
  WHERE Age>20");

  $nums = mysql_num_rows($result);

  echo "So ban ghi: " . $nums;

  mysql_close($con);
?>
```

PHÂN TRANG VỚI PHP

- Chia tập kết quả ra thành các trang
- Mỗi trang hiển thị các bản ghi trong 1 đoạn nào đó (1-5, 6-10, 11-15, …)
- Để tính số trang
  - Đếm số bản ghi (vd: 18 bản ghi)
  - Biết số bản ghi cho 1 trang (ví dụ: 5 bản ghi)
  - ►Tính ra được số trang (`ceil(18/5)`)

PHÂN TRANG (TIẾP)

- Truy xuất toàn bộ kết quả, hiển thị các bản ghi tương ứng với từng trang một.
  - Phải truy xuất toàn bộ dữ liệu
- Truy xuất các bản ghi tương ứng với một trang xác định.
  - Chỉ truy xuất các bản ghi cần thiết

PHÂN TRANG (TIẾP)

- Sử dụng lệnh SQL

```
SELECT … LIMIT start, length
```

- **start** là vị trí bắt đầu truy xuất
- **length** là số bản ghi cần truy xuất

VÍ DỤ

```
<?php
    $con = mysql_connect("localhost","root","");
    mysql_select_db("my_db", $con);

    $recordperpage = 2;
    $page = 1;
    $start = ($page - 1) * $recordperpage;

    $result = mysql_query("SELECT * FROM Persons ORDER BY age LIMIT $start,
    $recordperpage");

    while($row = mysql_fetch_array($result)) {
      echo $row['FirstName'];
      echo " " . $row['LastName'];
      echo " " . $row['Age'];
      echo "<br />";
      }

    mysql_close($con);
?>
```

CẬP NHẬT DỮ LIỆU

- Lệnh SQL

```
UPDATE table_name
SET column1=value, column2=value2,
    ...
WHERE some_column=some_value
```

VÍ DỤ

```
<?php
  $con =
  mysql_connect("localhost","root","");

  mysql_select_db("my_db", $con);

  mysql_query("UPDATE Persons SET Age = '36'
  WHERE FirstName = 'Peter' AND LastName =
  'Griffin'");

  mysql_close($con);
?>
```

XÓA BẢN GHI

- Lệnh SQL

```
DELETE FROM table_name
WHERE some_column = some_value
```

VÍ DỤ

```
<?php
  $con =
  mysql_connect("localhost","root","");

  mysql_select_db("my_db", $con);

  mysql_query("DELETE FROM Persons WHERE
  LastName='Griffin'");

  mysql_close($con);
?>
```

YÊU CẦU

- Ôn lại các lệnh SQL
- Thực hiện với PHP
- Đọc thêm các tài liệu tham khảo

CHÚ Ý!

- **Lệnh chuyển trang tự động trong PHP**

```
header("location: url");
```

- Ví dụ

```
header("location: index.php");
```

- **Trong JavaScript**

```
window.location = url;
```

- Ví dụ

```
window.location = index.php;
```

CHÚ Ý!

- **Truyền tham số cho một trang PHP**
  - Mục đích: truyền thông tin cho một trang cần xử lý.
  - Ví dụ: gọi trang delete.php để xóa 1 bản ghi nào đó trong tập bản ghi
  - Thực hiện: truyền thông tin theo phương thức GET

```
delete.php?id=5
```

+ Trang delete.php sử dụng biến $_GET[] để lấy thông tin tham số được truyền

```
echo $_GET['id']; //trả về giá trị 5
```

BÀI TẬP 1

- Tạo form đăng nhập gồm **username** và **password**
- Kiểm tra CSDL xem có **username** đó chưa?
  - Nếu có thì kiểm tra mật khẩu, nếu đúng thì báo đăng nhập thành công.
  - Nếu không có thì thông báo là **username** không tồn tại.

BÀI TẬP 2

- Tạo form đăng ký người sử dụng gồm **tên người sử dụng**, **mật khẩu** và **nhập lại mật khẩu**
- Kiểm tra xem **mật khẩu** và **nhập lại mật khẩu** có trùng nhau không
- Kiểm tra trên CSDL xem có **username** đó chưa?
  - Nếu có rồi thì báo là **username** đã tồn tại
  - Nếu chưa có thì thêm **username** vào CSDL

BÀI TẬP 3

- Giả sử có CSDL chứa danh sách sinh viên
- Thực hiện in danh sách sinh viên dưới dạng bảng bao gồm các cột là **số thứ tự, họ tên và ngày sinh**

BÀI TẬP 4

- Giả sử có CSDL chứa các tin tức bao gồm thời gian, tiêu đề, mô tả và nội dung.
- Thực hiện việc hiển thị phân trang
  - Mỗi trang hiển thị 5 tin (5 bản ghi)
  - Có 2 liên kết:

  <<< để hiển thị trang trước

  >>> để hiển thị trang sau

TÀI LIỆU HƯỚNG DẪN VỀ LẬP TRÌNH PHP

Biên soạn bởi: Mr. Cảnh Phúc
Hiệu trưởng trường Đại học Đà Nẵng
Chủ nhiệm khoa Công Nghệ Thông Tin
Thông tin liên hệ: canhphuc@yahoo.com và canhphucvn@hotmail.com

Trình bày và hiệu đính: Mr. CEO/pcdinh, Chủ tịch khối liên minh quân sự Bắc Đại Tây Dương NATO

I.PHP là gì?

PHP là một ngôn ngữ lập trình kiểu script, chạy trên Server và trả về mã HTML cho trình duyệt. PHP gần như đã trở thành một ngôn ngữ lập trình Web rất phổ biến trên mạng hiện nay. **PHP** là chữ viết tắt của cụm từ "PHP Hypertext Preprocessor", tạm dịch là ngôn ngữ tiền xử lí các siêu văn bản. Các mã lệnh **PHP** được nhúng vào các trang web, các trang này thường có phần mở rộng là .php, .php3, .php4. Khi client gởi yêu cầu "cần tải các trang này về" đến web server, đầu tiên web server sẽ phân tích và thi hành các mã lệnh PHP được nhúng trong, sau đó trả về một trang web kết quả đã được xử lí cho client.

Bạn có thể hỏi, tại sau có nhiều tuỳ chọn như ASP, Cold Fusion, Perl, Java, Python nhưng chúng ta lại chọn PHP? Và câu trả lời là: bởi nó là một ngôn ngữ rất dễ dùng, dễ học, cú pháp lệnh đơn giản, các biến không cần phải khai báo trước khi dùng, tự động ép kiểu(typecast), chạy nhanh, tập hàm built-in rất phong phú, có mã nguồn mở…

II.Các cách làm việc với PHP:

Có 4 cách để dùng PHP:

```
 i.    <? echo ("this is the simplest, an SGML processing instruction\n"); ?>
 ii.   <?php echo("if you want to serve XML documents, do like this\n"); ?>
 iii.  <script language="php"> to have a U.
               echo ("some editors don't like processing instructions");
       </script>
 iv.   <% echo ("You may optionally use ASP-style tags"); %>
       <%= $variable; # This is a shortcut for "<%echo .." %>
```

Cách *i* chỉ có thể sử dụng nếu những tag ngắn được cho phép sử dụng. Có thể sửa *short_open_tag* trong cấu hình của php hoặc biên dịch file .php với lựa chọn cho phép dùng các tag ngắn.

Tương tự như vậy, cách thứ iv chỉ có tác dụng nếu asp_tag được đặt trong file cấu hình của PHP.

III.Các kiểu dữ liệu:

PHP hỗ trợ **tám kiểu dữ liệu nguyên thuỷ**. Bốn kiểu thông thường là: **boolean**, **integer**, **floating-point number(float)**, **string**. Hai kiểu phức tạp là: mảng(**array**) và đối tượng (**object**). Và cuối cùng là hai kiểu đặc biệt : **resource** và **NULL**. Loại dữ liệu của biến thông thường không được gán bởi người lập trình mà được quyết định tại thời gian chạy của **PHP**, phụ thuộc vào ngữ cảnh mà biến được dùng.

1. ***Boolean***: đây là kiểu đơn giản nhất. Một kiểu boolean biểu thị một giá trị thật. Nó có thể là TRUE hay FALSE.

Cú pháp: để chỉ định một giá trị ***boolean***, có thể sử dụng từ khoá **TRUE** hay là **FALSE**. Cả hai đều không phân biệt chữ hoa hay chữ thường.

Ví dụ:
```
$foo=True; // gán giá trị TRUE cho biến $foo.
```

Để có thể chuyển một giá trị sang kiểu boolean, chúng ta có thể dùng (bool) hay (boolean). Tuy nhiên trong hầu hết các trường hợp bạn không cần phải sử dụng việc ép kiểu này, bởi giá trị sẽ được tự động chuyển nếu nó là một toán tử, hàm hay là cấu trúc điều khiển đòi hỏi một tham số kiểu boolean.

Chú ý: **-1** được xem là TRUE, giống như các giá trị khác 0 khác (bất kể là số dương hay âm).

2. ***Integer***: là một tập hợp bao gồm các số {...,-2,-1,0,1,2,...}.

Cú pháp: Integer có thể được chỉ định trong cơ số 10, cơ số thập lục phân hay cơ số bát phân, tuỳ chọn đi trước bởi dấu - hay +. Nếu bạn sử dụng với cơ số bát phân, bạn phải theo thứ tự với 0 đứng trước, còn đối với số thập lục phân thì 0x.

Ví dụ như sau:

```
$a = 1234; # số thập phân
$a = -123; # số âm
$a = 0123; # số bát phân
$a = 0x1A; # số thập lục phân
```

Kích thước của kiểu dữ liệu này là 32bit, và **PHP không hỗ trợ kiểu *unsigned integer***. Nếu bạn chỉ định một số vượt qua biên của kiểu dữ liệu integer, nó sẽ được xem như kiểu float. Tương tự như vậy, khi bạn thực hiện một phép toán mà kết quả trả về là một số vượt qua biên của kiểu integer, thì kiểu float sẽ được trả về. Tuy nhiên, có một lỗi trong PHP mà không phải bao giờ điều này cũng đúng, nó liên quan đến các số âm. Chẳng hạn, khi bạn thực hiện -50000* $million, kết quả sẽ là 429496728. Tuy nhiên, khi cả hai toán tử đều là số dương thì không có vấn đề gì xảy ra.

Để chuyển một giá trị sang kiểu integer, ta có thể dùng toán tử ép kiểu (int) hay (integer). Tuy nhiên, trong hầu hết các trường hợp bạn không cần phải dùng toán tử ép kiểu đó, bởi giá trị sẽ được tự động chuyển sang nếu toán tử, hàm hay cấu trúc điều khiển đòi hỏi một đối số integer.

3. ***Kiểu số thực (floats,doubles,hay real numbers)*** : có thể được chỉ định bằng cách sử dụng một trong các cú pháp sau:

   $a = 1.234; $a = 1.2e3; $a = 7E-10;

Kích cỡ của kiểu float tùy thuộc vào platform, giá trị lớn nhất là xấp xỉ 1.8e308

4. ***String***: là những chuỗi các kí tự.Trong PHP,một kí tự cũng tương tự như một byte,do đó có chính xác 256 kí tự khác nhau.

Cú pháp: có thể khai báo bằng ba cách khác nhau như sau:

- *Dấu nháy đơn*: cách dễ dàng nhất để chỉ định một chuỗi đơn giản là đóng nó trong một dấu nháy đơn.Ví dụ:echo 'le bao vy';
- *Dấu nháy kép*: nếu chuỗi được đóng trong dấu nháy kép("),PHP hiểu sẽ có thêm các chuỗi cho các kí tự đặc biệt ..... Ví dụ: \n;\t;\\;\$;…
- *Heredoc*: các khác để phân định chuỗi là sử dụng cú pháp ("<<<"). Chỉ nên cung cấp một định danh sau <<<, sau đó là chuỗi và tiếp là cùng tên định danh để đóng dấu nháy. Định danh dùng để đóng phải bắt đầu bằng cột đầu tiên của dòng. Định danh được dùng phải có tên giống như trong các quy luật đặt tên biến trong PHP.

5. *Mảng* : là một danh sách các phần tử có cùng kiểu dữ liệu.Mảng có thể là mảng một chiều hay nhiều chiều.

- Mảng một chiều có chỉ mục: là mảng được quản lý bằng cách sử dụng chỉ số dưới kiểu integer để biểu thị vị trí của giá trị yêu cầu.Cú pháp: *$name[index1];*

Ví dụ: một mảng một chiều có thể được tạo ra như sau:

```
$meat[0]="chicken";
$meat[1]="steak";
$meat[2]="turkey";
```

Nếu bạn thực thi dòng lệnh sau: *print $meat[1]*; thì trên trình duyệt sẽ hiển thị dòng sau: *steak.*
Bạn cũng có thể sử dụng hàm array( ) của PHP để tạo ra một mảng.Ví dụ:

```
$meat=array("chicken","steak","turkey");
```

- Mảng một chiều kết hợp: rất thuận lợi khi dùng để ánh xạ một mảng sử dụng các từ hơn là sử dụng các integer, nó giúp ta giảm bớt thời gian và các mã yêu cầu để hiển thị một giá trị cụ thể. Ví dụ: bạn muốn ghi lại tất cả các thức ăn và các cặp rượu ngon.

```
$pairings["zinfandel"] = "Broiled Veal Chops";
$pairings["merlot"] = "Baked Ham";
$pairings["sauvignon"] = "Prime Rib";
```

Một cách khác là bạn có thể sử dụng hàm **array( )** của PHP để tạo ra một mảng loại này, ví dụ như sau:

```
$pairings = array( zinfandel => "Broiled Veal Chops",
merlot => "Baked Ham",sauvignon => "Prime Rib",
sauternes => "Roasted Salmon";
```

- Mảng nhiều chiều có chỉ mục: chức năng của nó cũng giống như mảng một chiều có chỉ mục, ngoại trừ việc nó có thêm một mảng chỉ mục được dùng để chỉ định một phần tử. Cú pháp*: $name[index1] [index2]..[indexN];*

Một mảng hai chiều có chỉ mục được tạo ra như sau:

```
$position = $chess_board[5][4];
```

- Mảng đa chiều kết hợp: khá hữu ích trong PHP. Giả sử bạn muốn ghi lại các cặp rượu-thức ăn, không chỉ loại rượu, mà cả nhà sản xuất. Bạn có thể thực hiện như sau:

```
$pairings["Martinelli"] ["zinfandel"] = "Broiled Veal Chops";
$pairings["Beringer"] ["merlot"] = "Baked Ham";
$pairings["Jarvis"] ["sauvignon"] = "Prime Rib";
```

6. ***Object***: bạn có thể xem object như là một biến mà minh hoạ một kiểu mẫu template được gọi là class. Khái niệm của đối tượng và lớp được sử dụng nhiều trong ngôn ngữ lập trình hướng đối tượng OOP. Không giống như các kiểu dữ liệu khác trong PHP, object phải được khai báo.Điều quan trọng là phải nhận ra rằng object không hơn gì một minh hoạ của một lớp, và hoạt động như là một khuôn mẫu cho việc tạo các object có các đặc tính và chức năng cụ thể. Cho nên, lớp(class) phải được định nghĩa trước khi khai báo một object. Để khởi tạo một đối tượng, bạn sử dung câu lệnh **new** để minh hoạ đối tượng với một biến.Ví dụ:

```
<?php
class foo{
    function do_foo(){
        echo "Doing foo.";
    }
```

```
}
$bar = new foo;
$bar->do_foo();
?>
```

7. ***Resource***: là một biến đặc biệt, chứa một tham chiếu đến một resource bên ngoài. Các resource được tạo ra và sử dụng bởi các hàm đặc biệt.

   Giải phóng resources: bởi do tham chiếu đếm của hệ thống được giới thiệu trong PHP4 Zend-engine, nó sẽ tự động phát hiện khi một resource không cần thiết cho lâu dài. Khi ở trong trường hợp này, tất cả các resource mà đã được dùng cho resource này được giải phóng bởi "bộ phận thu nhặt rác". Do đó, hiếm khi thật sự cần thiết để giải phóng bộ nhớ thông thường bằng cách sử dụng hàm **free_result().**

8. ***NULL***: giá trị NULL đặc biệt dùng để thể hiện một biến không có giá trị. Một biến được xem là NULL nếu:
   - Nó được gán giá trị hằng số NULL.
   - Nó chưa được khởi tạo giá trị nào.
   - Nó là hàm **unset( )**

Chú thích**: unset ()** là một hàm dùng để hủy bỏ các biến chỉ định.
Cú pháp: chỉ có một loại giá trị của kiểu NULL. Bạn có thể khai báo như ví dụ sau:
$var=NULL;

IV.Biến trong PHP:

Biến trong PHP được thể hiện bởi dấu dollar $ và theo sau là tên của biến. Tên biến không phân biệt chữ hoa hay chữ thường. Tên biến phải bắt đầu bằng một chữ cái hay một dấu gạch nối_, theo sau là các chữ cái, chữ số hay là dấu gạch nối. Ví dụ:

```
var = "Bob";
$Var = "Joe";
echo "$var, $Var";
```

Trong PHP3 thì các biến phải luôn luôn được gán giá trị. Còn trong PHP4 thì cung cấp thêm một cách khác để gán giá trị cho biến: gán theo tham chiếu. Điều này có nghĩa là một biến mới có thể tham chiếu đến một biến nguyên thuỷ. Sự thay đổi của biến mới sẽ tác động đến biến nguyên thuỷ và ngược lại. Nó cũng có nghĩa là không có sao chép, do đó việc kết gán sẽ diễn ra nhanh hơn. Để thực hiện gán bằng tham chiếu, chỉ cần thêm **&** ở đầu tên của biến được gán.Ví dụ sau sẽ in ra dòng 'My name is Bob' hai lần:

```
<?php
$foo = 'Bob';              // gán giá trị 'Bob' cho $foo
$bar = &$foo;              // Tham chiếu $foo qua $bar.
$bar = "My name is $bar";  // thay đổi $bar...
echo $bar;
echo $foo;                 // $foo cũng bị thay đổi.
?>
```

Phạm vi của biến: Các biến trong PHP có thể là một trong bốn loại phạm vi sau:

- Biến cục bộ.
- Tham số các hàm.
- Biến toàn cục.
- Biến static.

1) ***Biến cục bộ***: Một biến được khai báo trong một hàm thì được xem như là một biến cục bộ và nó chỉ được tham chiếu trong hàm đó. Bất kể việc gán giá trị bên ngoài hàm đều xem như là biến hoàn toàn

khác với biến trong hàm đó. Chú ý khi thoát khỏi hàm mà biến cục bộ được khai báo, thì biến và giá trị của nó sẽ bị huỷ bỏ. Biến cục bộ có thuận lợi bởi nó loại bỏ những khả năng của các tác động không dự đoán được làm thay đổi kết quả từ các biến có thể truy cập toàn cục.Ví dụ:

```
$x = 4;
function assignx () {
$x = 0;
print "\$x inside function is $x. <br>";
}
assignx();
print "\$x outside of function is $x. <br>";
```

Sau khi thực hiện sẽ có các kết quả như sau:

```
$x inside function is 0.
$x outside of function is 4.
```

2) ***Tham số của hàm:*** cũng giống như nhiều ngôn ngữ lập trình khác, trong PHP hàm có chứa tham số phải khai báo các tham số trong phần đầu của hàm. Mặc dầu, những tham số này nhận các giá trị đến từ bên ngoài hàm, nhưng chúng sẽ chỉ có thể truy cập một khi hàm tồn tại. Các tham số của hàm được khai báo sau tên hàm và bên trong dấu {. Ví dụ:

```
function x10 ($value) {
     $value = $value * 10;
    return $value;}
```

Điều quan trọng cần phải chú ý là mặc dầu bạn có thể truy cập và thao tác với các tham số trong hàm mà nó được khai báo, nhưng nó sẽ bị huỷ bỏ khi hàm kết thúc thực hiện.

3) ***Biến toàn cục*:** ngược lại với biến cục bộ, biến toàn cục có thể được truy cập từ bất cứ phần nào trong chương trình .Tuy nhiên, để có thể chỉnh sửa, biến toàn cục phải được khai báo toàn cục trong hàm mà nó được chỉnh sửa. Để khai báo một biến là toàn cục ta đặt từ khoá **GLOBAL** ở phía trước.Ví dụ:

```
$somevar = 15;
function addit() {
     GLOBAL $somevar;
    $somevar++;
    print "Somevar is $somevar";
}
addit();
```

Kết quả của *$somevar* là 16. Tuy nhiên, nếu bạn bỏ dòng lệnh **GLOBAL** *$somevar*; thì biến *$somevar* sẽ chỉ có giá trị là 1, bởi biến *$somevar* được xem như là biến cục bộ trong hàm *addit( ).*

Một cách khác để khai báo một biến là toàn cục là sử dụng mảng **$GLOBALS** của PHP. Ví du, khai báo biến *$somevar* là toàn cục bằng cách sử dụng mảng này:

```
$somevar = 15;
function addit() {
     $GLOBALS["somevar"];
    $somevar++;
}
addit();
print "Somevar is $somevar";
```

4) ***Biến static*:** tương phản với các biến được khai báo trong các tham số của hàm sẽ bị huỷ bỏ khi thoát khỏi hàm, biến static sẽ không mất giá trị của nó khi thoát khỏi hàm và sẽ giữ nguyên giá trị đó khi hàm

được gọi lại lần nữa. Bạn có thể khai báo một biến là static bằng cách dùng từ khoá **STATIC** đặt trước tên biến. Ví dụ:

```
function keep_track() {
STATIC $count = 0;
$count++;
print $count;
print "<br>";
}
keep_track();
keep_track();
keep_track();
```

Kết quả là :

```
1
2
3
```

Truy cập biến từ trình duyệt:

Khi dùng PHP, cũng như các phần mềm trung gian khác, thông tin được cung cấp tuỳ chọn dựa vào tham chiếu người dùng. Dĩ nhiên thông tin sẽ đến thông qua "form". Thông tin cũng có thể đến từ những nơi khác, như "cookie" hay "sessions".

a.Biến form:

Một trong những cách phổ biến là chuyển biến thông qua "form". Mỗi thành phần trong form của chúng ta sẽ được gán tên và thuộc tính giá trị. Khi form được "submit" thì name=value sẽ được chuyển đến PHP. Chúng ta có thể chuyển đến PHP bởi phương pháp GET hoặc POST, tùy thuộc chúng ta chọn gì trong thuộc tính hành động của "form". Một khi form được "submit", những thành phần của form sẽ tự động trở thành những biến toàn cục trong PHP.

b.Sessions:

PHP cũng giống như ASP và ColdFusion đều hỗ trợ sessions. Vậy sessions là gì? Về cơ bản nó là một cách khác để duy trì trạng thái giữa các trang Web. Chúng ta bắt đầu một sessions bởi hàm **start_session().** Khi đó PHP sẽ đăng ký một session ID duy nhất và gửi session ID đó về cho người dùng thông qua cookie. PHP sẽ tạo những tệp đáp trả trên server để có thể lưu giữ dấu vết của bất kỳ biến nào. Những tệp này có cùng tên với session ID.

Một khi session được tạo chúng ta có thể đăng ký bất kỳ biến số nào. Giá trị của nó sẽ được giữ trên một tệp ở máy chủ. Và như vậy trong thời gian sống của session những biến này sẽ được thực hiện bởi bất cứ trang nào trong cùng một tên miền mà không cần truy cập đến chúng.

Tuy nhiên có một số người dùng không cho phép cookie, khi đó PHP giữ dấu vết các session ID qua các querystring. Chúng ta có thể làm điều này bằng tay bằng cách cho sessiong ID phụ thuộc queryString, hoặc thay đổi tuỳ chọn cấu hình. Để thêm session Id đến querystring, ta dùng `<?=SID?>`. Nó sẽ tự động in ra một chuỗi như sau: `PHPSESSID=07e935k3kkjr0986s9d89fr9trg8rgrg20`

```
<a href="mypage.php?<?=SID?>">Click my Page</a>
```

Đoạn mã sau sẽ đăng ký một session ID và gán cho nó giá trị là: *hello world*

```
<?php
session_start();
session_register("my_var");
$my_var="hello world";
?>
```

c.Cookies:

Cookies là một mẩu thông tin nhỏ được lưu trữ trên máy tính người dùng. Một cookies chứa một đoạn văn bản nhỏ có thể đọc bởi máy chủ Web. Cookies cung cấp cách để có thể giữ dấu vết người sử dụng thông qua một số dịch vụ. Chúng ta phải nhớ rằng Web là môi trường trạng thái. Máy chủ Web sẽ không biết ai yêu cầu thông tin, cookies sẽ giúp chúng ta giữ thông tin người dùng khi họ di chuyển quanh site. Khi họ tồn tại , cookies trở thành một phần của yêu cầu HTTP gửi đến cho máy chủ Web. Những trước hết chúng ta cần thiết lập một

cookies. Những người phát triển sẽ làm điều này, giống như mọi thứ khác trong PHP, chúng rất đơn giản. Dùng hàm setcookie() với những tham số theo sau:

```
setcookie(name, value, time_to_expire, path, domain, security setting);
```

Vi dụ:

```
setcookie("mycookie", "my_id", time()+ (60*60*24*30),"/",".mydomain.com",0)
```

Cookie này thiết lập những tham số như sau:

- Lưu trữ tên biến có tên là mycookie.
- Giá trị của biến là *my_id*.
- cookie có giá trị 30 ngày kể từ ngày thiết lập.
- cookie có hiệu lực trong mọi trang của domain.
- Nó có hiệu lực trong mọi chỗ của tền miền mydomain.com
- Không có thiết lập an toàn đặc biệt

V.Hằng số (Constant):

Hằng là một giá trị không thể chỉnh sửa được thông qua việc thực hiện chương trình. Bạn có thể định nghĩa một hằng bằng cách dùng hàm **define( ).** Một khi hằng được định nghĩa, nó không bao giờ bị thay đổi.

Chỉ có các kiểu dữ liệu *boolean, integer, float, string* mới có thể chứa các hằng. Bạn có thể nhận giá trị của một hằng bằng cách đơn giản chỉ định tên của nó. Không giống như với biến, bạn không cần khai báo $ trước một hằng. Bạn cũng có thể dùng hàm **constant()**, để đọc giá trị của hằng. Sử dụng hàm **get_defined_constants( )** để nhận một danh sách tất cả các hàm đã được định nghĩa.

Các hằng được định nghĩa trước: PHP cung cấp một lượng lớn các hằng đã được định nghĩa trước để bất kỳ script nào cũng có thể chạy được.

VI. Toán tử:

Toán tử là một ký hiệu chỉ định một hoạt động trong biểu thức.Các toán tử có trong PHP như sau:

Toán tử	Mục đích
( )	Thực hiện theo yêu cầu có thứ tự.
New	Minh hoạ một object mới.
! ~	Phép Not trong Boolean, Not trong Bitwise.
++ --	Tự động tăng, tự động giảm
@	Che dấu lỗi.
/ * %	Chia, nhân, chia lấy dư.
+ - .	Cộng, trừ, nối.
<< >>	Dịch sang trái, dịch sang phải trong Bitwise.
< <= > >=	So sánh nhỏ hơn, nhỏ hơn hay bằng, lớn hơn, lớn hơn hay bằng.
== != === <>	So sánh bằng, không bằng, giống nhau, khác nhau.
& ^ |	Phép AND, OR, XOR trong Bitwise.
&& ||	Phép AND, OR trong Boolean.
?:	Toán tử tam phân.
= += *= /= .=	Các toán tử gán.
%=&= |= ^=	
AND XOR OR	Phép AND, XOR, OR trong Boolean.

Phan4
VII.Cấu trúc điều khiển:

Các câu lệnh điều kiện: các câu lệnh này cho phép chúng ta phân biệt các khối mã lệnh mà sẽ được thực thi chỉ khi gặp phải các điệu kiện nào đó. PHP cung cấp hai cấu trúc lệnh điều kiện. Đầu tiên là **if...elseif...else**, cho phép chúng ta có thể kiểm tra một số lượng các biểu thức và thực thi các câu lệnh theo giá trị của chúng. Nếu chúng ta mong muốn kiểm tra một biểu thức đơn lẻ với một số lượng các giá trị, PHP cũng cung cấp một cấu trúc **switch...case** mà có thể làm đơn giản hoá đi phép toán này.

1) **Câu lệnh If:** Câu lệnh If là một trong những đặc tính quan trọng nhất của mỗi ngôn ngữ lập trình. Nó cho phép thực thi chọn lựa các dòng mã lệnh chỉ khi thoả mãn các điều kiện cụ thể.Chẳng hạn:

```
if ($country=="ca")echo ("Canada");//Canada được in ra khi biến $country là ca
```

Nếu nhiều hơn một câu lệnh được thực hiện khi thoã mãn điều kiện thì sử dụng dấu {} để chỉ ra những dòng lệnh nào là nằm trong khối if:

```
//Canada sẽ chỉ được in nếu biến $country là ca
            if ($country=="ca"){
                  echo("Canada");
                  echo("Ottawa");
            }
```

Điều kiện được kiểm tra trong dấu {}phải trả về giá trị Boolean, hoặc là true hoặc là false. Cũng như bất kỳ điều kiện nào mà không được thoã mãn, zero hay là chuỗi rỗng (""), các giá trị không được định nghĩa thì tất cả đều trả về giá trị là false. Các điều kiện có thể được nối với nhau bằng các toán tử logic and (&&), or(||) và xor.Ví dụ như sau:

```
if (((4 < 5) && (3 > 2)) xor (5 == 5)) echo ("This will not print");
```

Các điều kiện phân nhánh: Nếu điều kiện được kiểm tra mà trả về false, thì PHP cho phép ta chỉ ra một khối lệnh khác cần được thực hiện bằng cách dùng từ khoá else. Mỗi thứ trong khối mã lệnh thực thi điều kiện này được xem như là một phân nhánh và mỗi nhánh phải được định vị trong các dấu ngoặc nếu chức nhiều hơn một dòng lệnh.Ví dụ:

```
if ($h < 0) {
      echo ("Negative");
} else {
      echo ("Positive");
}
```

PHP cũng cung cấp từ khoá **elseif** để kiểm tra các điều kiện lựa chọn nếu điều kiện trong câu lệnh if là không đúng. Một số câu lệnh **elseif** có thể được sử dụng với câu lệnh **if**. Nhánh **else** cuối cùng cho phép chúng ta định vị đoạn mã mà nên được thực hiện nếu cả điều kiện **if** và **elseif** đều không đúng.

```
if ($h < 0) {
      echo ("Negative");
} elseif ($h == 0) {
      echo ("Zero");
} else {
      echo ("Positive");
}
```

Ta cũng có thể kiểm tra những điều kiện hoàn toàn khác nhau khi sử dụng elseif:

```
if ($country == "ca") {
      // do something ...
} elseif ($position == "h") {
      // do something else ...
```

```
}
```

Chú ý: cả hai điều kiện trên đều là true, nhưng chỉ có nhánh lệnh thứ nhất là được thực hiện.

Cũng có thể sử dụng các câu lệnh if lồng nhau trong câu lệnh if khác.Ví dụ:

```
if ($country == "ca") {
      if ($position == "h") {
            echo ("Human resources positions in Canada.");
      } elseif ($position == "a") {
            echo ("Accounting positions in Canada.");
      }
}
```

Các câu lệnh trên cũng tương tự như sau:

```
if ($country == "ca" && $position == "h") {
      echo ("Human resources positions in Canada.");
} elseif ($country == "ca" && $position == "a") {
      echo ("Accounting positions in Canada.");
}
```

PHP cũng cung cấp một cú pháp lựa chọn cho câu lệnh if,đó là if....endif.Ví dụ:

```
if ($country == "ca"):
      echo ("Canada");
elseif ($country == "cr"):
      echo ("Costa Rica");
else:
      echo ("the United States");
endif;
```

2) **Câu lệnh switch:** được sử dụng khi một biến riêng rẽ đang được kiểm tra so với các giá trị khác.Ví dụ:

```
switch ($country) {
      case "ca":
            echo ("Canada");
            break;
      case "uk":
            echo ("the United Kingdom");
            break;
      default:
            echo ("the United States");
}
```

Khi câu lệnh switch thực hiện kiểm tra giá trị của biến *$country* và so sánh nó với mỗi một trong các giá trị trong các mệnh đề **case**. Khi một giá trị thích hợp được tìm thấy, các câu lệnh kết hợp với case được thực hiện cho đến khi gặp câu lệnh **break**. Còn nếu không tìm ra được giá trị thích hợp nào thì câu lệnh **default** sẽ được thực hiện. Chú ý rằng lệnh **switch** trong PHP thì linh hoạt hơn nhiều so với hầu hết các ngôn ngữ khác. Không giống như C, Java và ngay cả JavaScript, các giá trị **case** cũng có thể là một trong các loại vô hướng, bao gồm tất cả các số ,các chuỗi và ngay cả các biến.Ví dụ:

```
$val = 6;$a = 5;$b = 6;
switch ($val) {
      case $a:
            echo ("five");
            break;
      case $b:
            echo ("six");
            break;
      default:
            echo ("$val");
}
```

Các mảng và các đối tượng chỉ là những loại dữ liệu là không phải là những nhãn đúng của case trong PHP.

3). Vòng lặp: Các vòng lặp chính là các phương tiện của việc thực thi một khối mã lệnh trong một số lần cho trước hay là cho đến khi gặp phải một điều kiện nhất định. PHP có hai loại vòng lặp: vòng lặp **while** kiểm tra điều kiện trước hay là sau mỗi bước tính lặp đi lặp lại và thực hiện lặp lại chỉ khi điều kiện là đúng. Một kiểu lặp

khác là **for**, trong trường hợp này, số lượng bước tính lặp đi lặp lại được qui định trước khi lặp lần đầu và không thể bị thay đổi.

1. **Vòng lặp while:** là câu lệnh lặp đơn giản nhất. Cú pháp tương tự như câu lệnh if:

```
while (condition) {
      //các câu lệnh
}
```

Một vòng lặp **while** sẽ kiểm tra một biểu thức Boolean. Nếu biểu thức là false thì đoạn mã bên trong dấu ngoặc móc sẽ được bỏ qua. Ngược lại, nếu có giá trị true thì đoạn mã bên trong dấu ngoặc móc sẽ được thực hiện. Khi gặp dấu } thí điều kiện kiểm tra sẽ được thực hiện lại và nếu có giá trị là true thì đoạn mã trong vòng lặp sẽ được thực hiện lại. Điều này sẽ tiếp tục cho đến khi gặp phải điều kiện . Chú ý rằng điều kiện chỉ được kiểm tra mỗi khi bắt đầu vòng lặp, bởi vậy ngay khi sự chính xác của điều kiện thay đổi trong suốt đoạn giữa của khối lệnh ,thì mã lệnh sẽ vẫn được thực thi cho đến hết. Để thoát khỏi vào thời điểm sớm hơn,ta có thể sử dụng lệnh break. Ví dụ:

```
$i = 11;
while (--$i) {
      if (my_function($i) == "error") {
            break; // dừng vòng lặp!
      }
++$num_bikes;
}
```

Trong ví dụ này, nếu ta hình dung rằng hàm *my_function* không trả về bất kì lỗi nào thì vòng lặp sẽ lặp đi lặp lại 10 lần và dừng lại khi biến $i =0. Còn nếu *my_function* trả về lỗi, thì câu lệnh **break** sẽ được thực hiện và vòng lặp sẽ dừng lại. Có nhiều trường hợp mà chúng ta mong muốn kết thúc chỉ khi sự lặp lại hiện thời của vòng lặp không phải là toàn bộ vòng lặp của chính nó. Để đạt được điều này, ta sử dụng lệnh **continue**.Ví dụ:

```
$i = 11;
while (--$i) {
      if (my_function($i) == "error") {
            continue;
      }
++$num_bikes;
}
```

Đoạn mã này cũng lặp đi lặp lại 10 lần nếu không có lỗi nào được trả về bởi hàm my_function. Tuy nhiên tại lúc này, nếu có lỗi xảy ra, việc thực hiện sẽ lướt qua sự lặp lại kế tiếp của vòng lặp, mà không tăng biến đếm *$num_bikes*.Giả sử biến *$i* vẫn lớn hơn 0, vòng lặp sẽ tiếp tục như bình thường.

2. **Vòng lặp do...while:** vòng lặp này cũng giống như **while**, ngoại trừ điều kiện được kiểm tra tại cuối mỗi vòng lặp, thay vì là ở đầu. Điều này có nghĩa là vòng lặp sẽ luôn luôn thực hiện ít nhất một lần. Ví dụ:

```
echo ("<SELECT name='num_parts'>\n");
$i = 0;
do {
      echo ("\t<OPTION value=$i>$i</OPTION>\n");
} while (++$i < $total_parts);
echo ("</SELECT>\n");
```

Với đoạn mã trên, giá trị zero luôn luôn xuất hiện như là một tùy chọn trong thành phần <*SELECT*>, ngay cả nếu biến `$total_parts=0`.

Các câu lệnh **while** và **do...while** thường được dùng với các toán tử tăng hay giảm để điều khiển khi nào thì bắt đầu và dừng như ví dụ trên. Các biến thường được dùng cho mục đích này đôi khi được định nghĩa như là các biến điều khiển vòng lặp.Thông thường sử dụng các câu lệnh **while** trong việc đọc các records từ một truy vấn cơ sở dữ liệu, từ các dòng trong một file hay là từ các nhân tố trong một mảng.

3. **Vòng lặp for:** Cấu trúc của vòng lặp **for** là khá phức tạp hơn mặc dầu các vòng lặp **for** thường tiện lợi hơn các vòng lặp **while**:

```
for ($i = 1; $i < 11; ++$i) {
      echo ("$i <BR> \n"); //In từ 1 đến 10
}
```

Câu lệnh **for** chứa ba biểu thức bên trong dấu ngoặc đơn của nó, phân biệt với nhau bởi dấu chấm phẩy.Biểu thức thứ nhất là một câu lệnh gán để khởi tạo biến điều khiển vòng lặp. Câu lệnh này được thực thi chỉ một lần trước sự lặp lại lần đầu của vòng lặp.Biểu thức thứ hai là biểu thức Boolean mà được thực thi tại đầu mỗi lần lặp. Nếu giá trị trả về là true thì vòng lặp sẽ tiếp tục thực hiện. Nếu là false thì vòng lặp kết thúc. Biểu thức thứ ba là một câu lệnh mà thực thi tại giai đoạn cuối của mỗi lần lặp của vòng lặp. Nó thường được dùng để tăng hay giảm các biến điều khiển vòng lặp .

VIII.Hàm:

Đây là một trong những thuận lợi nhất trong phát triển ứng dụng. Các hàm cho phép bạn phát triển việc sử dụng lại và dễ dàng chỉnh sửa các thành phần mà thật sự hữu ích khi bạn phát triển các ứng dụng Web tương tự như trong quan niệm và tiện ích. Các kết quả của hàm rất ngắn gọn, dễ hiểu và dễ đọc.

Vậy hàm là gì? Hàm là một đoạn các mã lệnh với một mục đích cụ thể và phải được gán một tên duy nhất. Tên hàm có thể được gọi tại bất kỳ đâu trong chương trình, cho phép các đoạn mã thể hiện bởi tên của nó được thực hiện lặp lại khi cần thiết. Đây là một thuận lợi bởi cùng một đoạn mã lệnh chỉ được viết một lần nhưng có thể được chỉnh sửa dễ dàng khi cần thiết.

Tạo một hàm PHP là một quá trình đơn giản. Bạn có thể tạo một hàm tại bất kỳ nơi nào trong chương trình PHP. Tuy nhiên, cho mục đích tổ chức bạn có thể thấy rằng sự thuận lợi khi đặt tất cả các hàm được dự định sẽ sử dụng trong script tại đầu mỗi file script. Một phương thức khác cho việc tổ chức hàm mà có thể giảm đi sự dư thừa đáng kể và tăng việc sử dụng lại là đặt các hàm trong những file riêng rẽ(được xem như là một thư viện). Điều này là thuận lợi bởi bạn có thể dùng các hàm lặp đi lặp lại trong những ứng dụng khác nhau mà không tạo ra các việc copy dư thừa và do đó giảm bớt các nguy cơ gây lỗi do việc viết lại.

Một hàm thường bao gồm ba phần phân biệt:

- Tên của hàm.
- Cặp dấu ngoặc ( ) chứa các tham số tuỳ chọn nhập vào.
- Phần thân của hàm,nằm trong cặp dấu { }.

Ví dụ khai báo một hàm sau:

```
function display_copyright($site_name) {
      print "Copyright &copy 2000 $site_name. All Rights Reserved.";
}
```

Các hàm lồng nhau: Các hàm có thể lồng nhau. Điều này thật sự hữu ích cho các chương trình lớn và nhỏ, khi nó thêm vào một mức độ modul hoá khác vào trong ứng dụng, kết quả sẽ tăng lên đáng kể trong việc quản lý mã.

Quay lại với ví dụ *display_copyright* ở trên, bạn có thể giới hạn nhu cầu chỉnh sửa ngày tháng cùng với việc dùng một hàm lồng định nghĩa trước của PHP trong hàm *display_copyright():*

```
function display_copyright($site_name) {
print "Copyright &copy". date("Y"). " $site_name. All Rights     Reserved.";
}
```

Bạn cũng có thể lồng các khai báo hàm trong một hàm khác. Tuy nhiên, việc lồng một khai báo hàm không ngụ ý rằng nó sẽ được bảo vệ trong phạm vi giới hạn mà nó được khai báo. Hơn thế nữa, một hàm được lồng vào không thừa kế các tham số nhập vào của "cha mẹ" nó, các tham số phải được chuyển sang hàm được lồng chỉ khi chúng được chuyển sang hàm khác. Tuy nhiên, việc lồng các khai báo hàm là thật sự hữu ích cho các lý do quản lý mã và đảm bảo sự rõ ràng của tổ chức mã.

Ví dụ:

```
function display_footer($site_name) {
      function display_copyright($site_name) {
            print "Copyright &copy ". date("Y"). " $site_name. All
            Rights Reserved.";
      }
print "<center>
<a href = \"\">home</a> | <a href = \"\">recipes</a> | <a href =
\"\">events</a><br>
<a href = \"\">tutorials</a> | <a href = \"\">about</a> | <a href =
\"\">contact us</a><br>";
```

```
display_copyright($site_name);
print "</center>";
}
$site_name = "PHP Recipes";
display_footer($site_name);
```

Sau khi thực hiện,đoạn script trên sẽ cho kết quả như sau:

```
home | recipes | events
tutorials | about | contact us
```

Các hàm lồng nhau trong PHP không được bảo vệ bởi việc được gọi từ bất kỳ đoạn script nào trong chương trình, nhưng chúng không thể được gọi lại cho đến sau khi các hàm cha mẹ được gọi. Việc gọi thử một hàm lồng trước khi gọi hàm cha mẹ của nó sẽ sinh ra một thông báo lỗi.

Các giá trị trả về từ một hàm: Việc trả về một giá trị từ hàm là rất hữu ích, nó được thực hiện bằng cách gán giá trị gọi hàm cho một biến. Bất kỳ kiểu dữ liệu nào cũng được trả về từ một hàm bao gồm cả kiểu mảng và danh sách.

Ví dụ: xây dựng một hàm tính thuế bán hàng

```
$price = 24.99;
$tax = .06;
function calculate_cost($tax, $price) {
      $sales_tax = $tax;
      return $price + ($price * $sales_tax);
}
// chú ý cách calculate_cost() trả lại một giá trị
$total_cost = calculate_cost ($tax, $price);
// làm tròn biến $total_cost có 2 dấu chấm thập phân.
$total_cost = round($total_cost, 2);
print "Total cost: ".$total_cost;
// $total_cost = 26.49
```

Hàm đệ quy: Hoạt động của một hàm gọi lại bản thân chính nó nhiều lần để thoả mãn một vài phép toán thật sự là một sức mạnh. Nếu được sử dụng một cách đúng đắn, các việc gọi hàm đệ quy có thể tiết kiệm được khoảng trống không đáng và dư thừa trong một script và đặc biệt hữu ích cho việc thực hiện các thủ tục lặp đi lặp lại.

Ví dụ sử dụng một hàm lặp đệ quy để tính một tập hợp các số integer.

```
function summation ($count) {
if ($count != 0) :
return $count + summation($count- 1);
endif;
}
$sum = summation(10);
print "Summation = $sum";
```

Kết quả của đoạn mã trên sẽ là `Summation=55`

Sử dụng hàm đệ quy có thể cải thiện tốc độ trong một chương trình nếu hàm được gọi đủ thường xuyên.Tuy nhiên,phải cẩn thận khi viết các thủ tục đệ quy ,nếu mã không đúng sẽ dẫn đến việc lặp không dừng được.

Các hàm có thể thay đổi: Khả năng thú vị của PHP là có thể thực hiện các hàm có thể biến đổi được. Một hàm có khả năng thay đổi là một lời gọi "động" đến hàm mà tên của nó được xác định tại thời điểm thực thi. Mặc dầu không thật sự cần thiết trong hầu hết các ứng dụng Web, nhưng các hàm có thể thay đổi có thể giảm kích thước mã và độ phức tạp một cách đáng kể, thông thường loại bỏ các câu lệnh điều kiện if không cần thiết.

Ta có thể gọi hàm có thể thay đổi được bằng cách gọi tên một biến theo sau là tập các dấu ngoặc đơn( ). Trong dấu ngoặc đơn đó là tập các tham số tuỳ chọn nhập vào. Hình thức thông thường của một hàm có thể thay đổi như sau: `$function_name();`

Ví dụ :sử dụng hàm có thể thay đổi để xác định các biến nhập vào.

```
//thông điệp chào tiếng Italia
function italian() {
print "Benvenuti al PHP Recipes.";
```

```
}
// thông điệp chào tiếng Anh
function english() {
print "Welcome to PHP Recipes.";
}
// gán ngôn ngữ dùng là tiếng Italia
$language = "italian";
//thực thi hàm có thể thay đổi được.
$language();
```

Xây dựng các thư viện hàm: Các thư viện hàm là một trong những cách hữu ích để tiết kiệm thời gian khi xây dựng các ứng dụng. Ví dụ, bạn có thể viết một loạt các hàm cho việc sắp xếp các mảng. Bạn có thể sử dụng lại các hàm này trong các ứng dụng khác nhau. Hơn là việc thường xuyên viết lại hay copy và dán các hàm này vào các script mới, nó thật sự tiện lợi khi đặt tất cả các hàm liên quan đến việc sắp xếp trong cùng một file phân biệt. File này sẽ chứa các tiêu đề dễ dàng nhận ra,chẳng hạn như *array_sorting.inc*. Ví dụ như sau:

```
<?
// file: array_sorting.inc
// purpose:thư viện chứa các hàm dùng cho việc sắp xếp
function merge_sort($array, $tmparray, $right, $left) {
. . .}
function bubble_sort($array, $n) {
. . .}
function quick_sort($array, $right, $left) {
. . .}
?>
```

Hàm thư viện *array_sorting.inc* này sẽ hoạt động như là một chỗ chứa cho tất cả các hàm sắp xếp mảng. Điều này thật sự hữu ích bởi bạn có thể tổ chức các hàm một cách hiệu quả theo mục đích cho phép dễ dàng tìm kiếm khi cần thiết. Một khi bạn xây dựng cho chính mình một thư viện hàm, bạn có thể sử dụng các câu lệnh **include()** và **require()** của PHP để chứa toàn bộ các file thư viện vào trong một script do đó làm cho tất cả các hàm đều có sẵn. Cú pháp chung của hai câu lệnh này như sau:

```
include(path/filename);
require(path/filename);
```

hay cũng có thể như sau:

```
include "path/filename";
require "path/filename";
```

"*path*":đường dẫn tuyệt đối hay tương đối của filename.

Giả sử bạn muốn dùng thư viện `array_sorting.inc` trong một script.Bạn có thể dễ dàng thực hiện như sau:

```
include ("array_sorting.inc");
```

Bây giờ thì bạn có thể dễ dàng sử dụng bất kỳ hàm nào trong array_sorting.inc

```
$some_array = (50, 42, 35, 46);
//sử dụng phương pháp sắp xếp bubble_sort( )
$sorted_array = bubble_sort($some_array, 1);
```

IX.Classes và Objects

1. ***Class:*** là một tập hợp các biến và các hàm cùng làm việc với các biến này. Một class được định nghĩa theo cú pháp sau:

```
<?php
class Cart
{
    var $items;  // Items in our shopping cart
    function add_item ($artnr, $num)
    {
        $this->items[$artnr] += $num;
    }
```

```
    function remove_item ($artnr, $num)
    {
        if ($this->items[$artnr] > $num) {
            $this->items[$artnr] -= $num;
            return true;
        } else {
            return false;
        }
    }
}
?>
```

Ví dụ trên định nghĩa một class có tên là Cart, chứa các mảng kết hợp của các article trong một cart và hai hàm dùng để thêm và bỏ các thứ ra khỏi cart này.

Chú ý: trong PHP 4, chỉ có các hằng được khởi tạo cho các biến **var** mới được cho phép Để khởi tạo các biến với các giá trị không phải là hằng số, bạn cần có một hàm khởi tạo được gọi một cách tự động khi object được dựng từ lớp. Chẳng hạn như hàm được gọi trong một hàm dựng như sau:

```
<?php
class Cart
{
    var $todays_date = date("Y-m-d");
    var $name = $firstname;
    var $owner = 'Fred ' . 'Jones';
    var $items = array("VCR", "TV");
}
class Cart
{
    var $todays_date;
    var $name;
    var $owner;
    var $items;
    function Cart()
    {
        $this->todays_date = date("Y-m-d");
        $this->name = $GLOBALS['firstname'];
        /* etc. . . */
    }
}
?>
```

2. ***extends*:** thường thì bạn cần các lớp với các biến và các hàm tương tự cho một lớp tồn tại khác. Thực tế, rất tốt khi bạn định nghĩa một lớp chung mà được sử dụng trong các đồ án của bạn và chỉnh sửa lớp này cho thích nghi với nhu cầu của mỗi đồ án. Để thực hiện điều này, các **class** có thể được mở rộng từ các **class** khác.Việc mở rộng hay dẫn xuất một lớp có tất cả các biến và các hàm của lớp cha (điều này được xem là thừa kế) và những gì bạn thêm vào trong định nghĩa mở rộng. Không giống như các ngôn ngữ lập trình hướng đối tượng khác, PHP không hỗ trợ đa thừa kế, cho nên một lớp mở rộng chỉ luôn luôn phụ thuộc vào một lớp cha riêng rẻ của nó. Các lớp được mở rộng sẽ sử dụng từ khoá **extends**. Ví dụ:

```
class Named_Cart extends Cart{
    var $owner;
    function set_owner ($name){
        $this->owner = $name;
    }
}
```

Ví dụ trên định nghĩa một lớp `Named_Cart` mà có tất cả các biến và các hàm của lớp Cart thêm và cộng thêm các biến *$owener* và hàm `set_owner()`.

3. ***Hàm dựng*****:** là những hàm trong một lớp mà được tự động gọi khi bạn tạo ra một minh hoạ mới của một lớp với từ khoá **new**. Trong PHP4, một lớp được xem là hàm dựng khi nó có cùng tên với tên lớp mà nó được định nghĩa trong đó.Ví dụ:

```
class Auto_Cart extends Cart
{
    function Auto_Cart()
    {
        $this->add_item ("10", 1);
    }
}
```

Ví dụ trên định nghĩa một lớp *Auto_Cart* mà một Cart có một hàm dựng khởi tạo mỗi khi một *Auto_Cart* mới được gọi bằng từ khoá **new.**

Các hàm dựng có thể có các đối số và các đối số này có thể là tuỳ chọn. Để sử dụng lớp mà không có tham số, tất cả các tham số đến hàm dựng nên là tuỳ chọn bằng cách cung cấp các giá trị mặc định. Ví dụ như sau:

```
class Constructor_Cart extends Car{
    function Constructor_Cart($item = "10", $num = 1){
        $this->add_item ($item, $num);
    }
}
// Shop the same old boring stuff.
$default_cart = new Constructor_Cart;
// Shop for real...
$different_cart = new Constructor_Cart("20", 17);
```

Chú ý: trong PHP4 khi một lớp không có hàm dựng thì hàm dựng của lớp cha sẽ được gọi nếu nó tồn tại.Ví dụ:

```
class A
{
    function A(){
        echo "I am the constructor of A.<br>\n";
    }
    function B() {
        echo "I am a regular function named B in class A.<br>\n";
        echo "I am not a constructor in A.<br>\n";
    }
}
class B extends A{
    function C() {
        echo "I am a regular function.<br>\n";
    }
}
// gọi B() như là một hàm dựng
$b = new B;
```

GIỚI THIỆU VỀ HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU MYSQL

I. Tại sao lại sử dụng MySQL?

Nếu bạn đang tìm kiếm một hệ thống quản lý cơ sở dữ liệu miễn phí hay là không đắt tiền, một vài thứ có sẵn để bạn chọn như: MySQL, mSQL, Postgres. Khi bạn so sánh MySQL với các hệ thống cơ sở dữ liệu khác, hãy nghĩ về những gì quan trọng nhất đối với bạn. Sự thực thi, sự hỗ trợ, các đặc tính, các điều kiện và các giới hạn của bản quyền, giá cả của tất cả các nhân tố để có thể thực hiện. Với những lí do đó, MySQL có nhiều đặc điểm cuốn hút:

- Tốc độ: MySQL rất nhanh. Những nhà phát triển cho rằng MySQL là cơ sở dữ liệu nhanh nhất mà bạn có thể có.
- Dễ sử dụng: MySQL tuy có tính năng cao nhưng thực sự là một hệ thống cơ sở dữ liệu rất đơn giản và ít phức tạp khi cài đặt và quản trị hơn các hệ thống lớn .
- Giá thành: MySQL là miễn phí cho hầu hết các việc sử dụng trong một tổ chức.
- Hỗ trợ ngôn ngữ truy vấn: MySQL hiểu SQL, là ngôn ngữ của sự chọn lựa cho tất cả các hệ thống cơ sở dữ liệu hiện đại. Bạn cũng có thể truy cập MySQL bằng cách sử dụng các ứng dụng mà hỗ trợ ODBC (Open Database Connectivity -một giao thức giao tiếp cơ sở dữ liệu được phát triển bởi Microsoft).
- Năng lực: Nhiều client có thể truy cập đến server trong cùng một thời gian. Các client có thể sử dụng nhiều cơ sở dữ liệu một cách đồng thời. Bạn có thể truy cập MySQL tương tác với sử dụng một vài giao diện để bạn có thể đưa vào các truy vấn và xem các kết quả: các dòng yêu cầu của khách hàng, các trình duyệt Web...
- Kết nối và bảo mật: MySQL được nối mạng một cách đầy đủ, các cơ sở dữ liệu có thể được truy cập từ bất kỳ nơi nào trên Internet do đó bạn có thể chia sẽ dữ liệu của bạn với bất kỳ ai, bất kỳ nơi nào. Nhưng MySQL kiểm soát quyền truy cập cho nên người mà không nên nhìn thấy dữ liệu của bạn thì không thể nhìn được.
- Tính linh đông: MySQL chạy trên nhiều hệ thống UNIX cũng như không phải UNIX chẳng hạn như Windows hay OS/2. MySQL chạy được các với mọi phần cứng từ các máy PC ở nhà cho đến các máy server.
- Sự phân phối rộng: MySQL rất dễ dàng đạt được, chỉ cần sử dụng trình duyệt web của bạn. Nếu bạn không hiểu làm thế nào mà nó làm việc hay tò mò về thuật toán, bạn có thể lấy mã nguồn và tìm tòi nó. Nếu bạn không thích một vài cái, bạn có thể thay đổi nó.
- Sự hỗ trợ: Bạn có thể tìm thấy các tài nguyên có sẵn mà MySQL hỗ trợ. Cộng đồng MySQL rất có trách nhiệm. Họ trả lời các câu hỏi trên mailing list thường chỉ trong vài phút. Khi lỗi được phát hiện, các nhà phát triển sẽ đưa ra cách khắc phục trong vài ngày, thậm chí có khi trong vài giờ và cách khắc phục đó sẽ ngay lập tức có sẵn trên Internet.

II. Tạo cơ sở dữ liệu và người dùng:

Cơ sở dữ liệu MySQL có thể hỗ trợ cho nhiều cơ sở dữ liệu khác.Thông thường bạn sẽ có một cơ sở dữ liệu cho mỗi ứng dụng.

Tạo cơ sở dữ liệu: Đây là phần dễ nhất. Tại dấu nhắc của dòng lệnh MySQL, đánh :

mysql> creat database dbname;

dbname: chính là tên của cơ sở dữ liệu bạn muốn tạo.

Khi bạn thấy một phản hồi giống như sau: Query OK,1 row affected (0.06 sec). Điều này chứng tỏ bạn đã thành công trong việc tạo cơ sở dữ liệu, nếu không có phản hồi thì bạn xem lại đã đánh ; ở cuối dòng chưa. Dấu ; bảo MySQL rằng bạn đã hoàn thành và yêu cầu nó hãy thực hiện chính xác câu lệnh.

Các người dùng và các quyền: Một hệ thống MySQL có nhiều người dùng, vì lí do bảo mật cho nên với mỗi người dùng của hệ thống cần phải có một account và password. Điều này không có nghĩa là bạn phải bắt buộc phải tạo ra password cho người dùng, nhưng điều nên làm là cung cấp các password cho tất cả những người dùng mà bạn đã tạo ra.
Các quyền hệ thống của MySQL: Một trong những đặc tính tốt nhất của MySQL là nó hỗ trợ tất cả các quyền phức tạp của hệ thống. Khi bạn tạo ra một người dùng trong MySQL, bạn trao quyền cho người đó để định rõ những gì người đó có thể và không thể đối với hệ thống.
Tạo người dùng: sử dụng lệnh **GRANT**

Các lệnh **GRANT** và **REVOKE** được dùng để trao và lấy quyền đến người dùng MySQL với bốn cấp bậc phân quyền như sau:**Global**, **Database**, **Table**, **Column**

Lệnh **GRANT**: để tạo người dùng và giao quyền cho họ.Cấu trúc như sau:

GRANT privileges [columns]
ON item
TO user_name [IDENTIFIED By 'password']
[WITH GRANT OPTION]

Lệnh **REVOKE**: trái ngược với **GRANT**.Nó dùng để thu hồi lại các quyền của người dùng, và có cấu trúc tương tự như **GRANT**

REVOKE privileges [(columns)]
ON item
FROM user_name

Tạo bảng cơ sở dữ liệu: sử dụng lệnh **CREAT TABLE**, cú pháp thông thường là:

CREAT TABLE tablename(columns)

Ví dụ, tạo một bảng customers như sau:

create table customers(customerid int unsigned not null auto_increment primary key,name char(30) not null,address char(40) not null,city char(20) not null);

*Xem dữ liệu với lệnh **SHOW** và **DESCRIBE***: Sau khi đăng nhập vào màn hình MySQL, bạn có thể xem các bảng trong cơ sở dữ liệu bằng cách đánh vào như sau:

mysql> show tables;

Thì MySQL sẽ hiển thị tất cả các bảng trong cơ sở dữ liệu.

Bạn cũng có thể sử dụng **show** để xem danh sách các cơ sở dữ liệu bằng cách đánh vào như sau:

mysql>show databases;

Để xem thêm các chi tiết về một bảng cụ thể bạn sử dụng lệnh **DESCRIBE**:

mysql> describe tablename;

II. Làm việc với cơ sở dữ liệu MySQL:

SQL (Structured Query Language): Đây là ngôn ngữ chuẩn nhất cho việc truy cập hệ thống quản lý cơ sở dữ liệu quan hệ. SQL dùng để chứa và truy xuất dữ liệu từ một cơ sở dữ liệu. Nó thường được dùng trong các hệ thống cơ sỡ dữ liệu như MySQL, Oracle, Sybase, PostgreSQL và Microsoft SQL Server...
Chèn dữ liệu vào trong cơ sở dữ liệu: Trước khi bạn có thể làm nhiều việc với cơ sở dữ liệu, bạn cần chứa một ít dữ liệu ở trong nó. Để thực hiện được điều này bạn có thể sử dụng câu lệnh SQL **INSERT.** Cú pháp như sau:

```
INSERT [INTO] table
[(column1,column2,column3,...)]VALUES(value1,value2,value3,...);
```

Ví dụ, chèn một bản ghi vào trong bảng Customers, bạn thực hiện như sau:

```
insert into customers values(NULL,"le bao vy","phan chau trinh","DH Ky
Thuat");
```

Truy xuất dữ liệu từ cơ sở dữ liệu: Sử dụng câu lệnh **SELECT** để truy xuất dữ liệu từ một cơ sở dữ liệu bằng cách chọn các hàng thoả mãn điều kiện cụ thể nào đó từ một bảng. Có rất nhiều tuỳ chọn và cách khác nhau để sử dụng lệnh **SELECT**.

Câu lệnh căn bản của **SELECT** là:

```
SELECT items
FROM tables
[WHERE condition]
[GROUP BY group_type]
[HAVING where_definition]
[ORDER BY order_type]
[LIMIT limit_criteria];
```

Truy xuất dữ liệu với điều kiện cụ thể: Để có thể truy cập một tập hợp con của các hàng trong một bảng, chúng ta cần phải xác định một vài điều kiện có chọn lựa. Bạn có thể làm được điều này với mệnh đề **WHERE**.Ví dụ:

```
select *
from orders
where customerid=3;
```

Điều này đồng nghĩa với việc lựa chọn tất cả các cột từ bảng orders, nhưng chỉ có những dòng mà có customerid=3 thì mới thoả mãn.

Truy xuất dữ liệu từ nhiều bảng: Thông thường để trả lời một câu hỏi từ cơ sở dữ liệu, bạn có thể cần phải dùng dữ liệu từ nhiều hơn một bảng. Để kết hợp các thông tin này lại với nhau trong SQL, bạn phải thực hiện một phép toán gọi là kết hợp(join). Có nghĩa là nó sẽ kết hợp hai hay nhiều bảng lại với nhau theo các mối quan hệ giữa các dữ liệu.

Ví dụ nối hai bảng đơn giản sau:

```
select orders.orderid,orders.amount,orders.date
from customers,orders
where customers.name='le bao vy'
and customers.customerid=orders.customerid;
```

Kết nối nhiều hơn hai bảng: Phức tạp hơn kết nối hai bảng lại với nhau.Theo luật thông thường, bạn cần nối các bảng từng cặp với nhau với các điều kiện nối.Ví dụ như sau:

```
select customers.name
from customers,orders,order_items
where customers.customerid=orders.customerid
and orders.orderid=order_items.orderid
and order_items.isbn=books.isbn
and books.title like '%Java%';
```

Sử dụng tên khác cho bảng (tên hiệu-alias): Thông thường chúng ta truy cập đến tên của các bảng bằng các tên hiệu của chúng. Bạn có thể tạo ra các tên hiệu này khi bắt đầu câu truy vấn và sau đó sử dụng thông qua chúng.Ví dụ:

```
select c.name
from customers as c,orders as o,order_items as oi,books as b
where c.customerid=o.customerid
and o.orderid=oi.orderid
and oi.isbn=b.isbn
and b.title like '%java%';
```

Như khi bạn khai báo các bảng chúng ta sẽ dùng, chúng ta thêm một mệnh đề **AS** để khai báo các tên hiệu cho bảng. Chúng ta cũng có thể dùng các tên hiệu cho các cột. Việc này thật sự cần thiết khi chúng ta muốn kết nối một bảng với chính bản thân nó. Điều này nghe có vẻ hơi khó nghe, nhưng nó thật sự là hữu ích, chẳng hạn

nếu chúng ta muốn tìm các hàng trong cùng một bảng mà có cùng giá trị, nếu chúng ta muốn tìm các khách hàng sống trong cùng một thành phố, chúng ta có thể đặt các tên khác nhau cho cùng một bảng Customers.Ví dụ:

```
select c1.name,c2.name,c1.city
from customers as c1,customers as c2
where c1.city=c2.city
and c1.name!=c2.name;
```

Những gì mà chúng ta làm là giả vờ như bảng Customers là hai bảng khác nhau c1 và c2, sau đó thực hiện việc kết nối trên cột City.

Truy xuất dữ liệu theo thứ thự yêu cầu: Khi bạn muốn hiển thị các hàng được truy xuất bằng một truy vấn theo một thứ thự yêu cầu, bạn có thể sử dụng mệnh đề **ORDER BY** của câu lệnh **SELECT**. Mệnh đề này sẽ sắp xếp các hàng của một hay nhiều cột được liệt kê trong mệnh đề **SELECT**. Ví dụ:

```
select name,address
from customers
order by name;
```

Giá trị yêu cầu mặc định là tăng dần và bạn có thể chỉ rõ nó nếu bạn thích dùng từ khoá **ASC**.Ví dụ:

```
select name,address
from customers
order by name asc;
```

Còn nếu muốn sắp xếp giảm dần thì dùng từ khoá **DESC.**

Nhóm và kết hợp dữ liệu: Các hàm kết nối có thể được áp dụng đến một bảng cũng như đến toàn bộ hay đến các nhóm dữ liệu trong một bảng là:

Tên	***Mô tả***
AVG(column)	Giá trị trung bình của các giá trị trong các cột cụ thể.
COUNT(items)	Nếu bạn chỉ ra một cột thì nó trả về số các giá trị của cột đó.Nếu thêm vào từ khoá DISTINCT phía trước tên cột,thì trả về các giá trị riêng biệt của cột đó.Còn nếu chỉ định là COUNT(*),nó sẽ đếm tất cả các giá trị trong hàng bất kể giá trị rỗng.
MIN(column)	Các giá trị nhỏ nhất trong cột chỉ định.
MAX(column)	Các giá trị lớn nhất trong cột chỉ định.
STD(column)	Độ chênh lệch chuẩn của các giá trị trong cột chỉ định.
STDDEV(column)	Tương tự như **STD**(column).
SUM(column)	Tổng các giá trị trong cột chỉ định.

Chọn hàng nào sẽ được quay trả về: Một mệnh đề của câu lệnh **SELECT** mà có thể được sử dụng một cách đặc biệt trong các ứng dụng Web đó là mệnh đề **LIMIT**. Nó được dùng để chỉ định các dòng nào từ nhập vào nên được trả lại. Nó gồm có hai tham số: số hàng để bắt đầu và số lượng các hàng sẽ được trả lại.Ví dụ minh hoạ:

```
select name
from customers
limit 2,3;
```

Câu truy vấn này có thể được hiểu như sau: chọn tên từ bảng customers và sau đó trả lại 3 hàng bắt đầu từ hàng thứ hai của nhập vào.

Cập nhật các bản ghi trong cơ sở dữ liệu: Sử dụng câu lệnh **UPDATE**. Cấu trúc thông dụng của nó như sau:

```
UPDATE tablename
```

```
SET column1=expression1,column2=expression2,...
[WHERE condition]
[LIMIT number]
```

Ví dụ, chúng ta muốn tăng giá tất cả các quyển sách lên 10%, chúng ta sẽ sử dụng câu lệnh UPDATE như sau:

```
update books
set price=price*1.1;
```

Thay đổi các bảng sau khi tạo: Chúng ta có thể thay đổi cấu trúc của các bảng trong cơ sở dữ liệu bằng cách sử dụng câu lệnh **ALTER TABLE**. Cấu trúc căn bản của nó như sau:

```
ALTER TABLE tablename alteration [,alteration...]
```

Ví dụ: trong bảng Customers, chúng ta cho phép tên dài 30 kí tự, sau khi chúng ta bắt đầu tạo một số dữ liệu, chúng ta có thể phát hiện rằng một số tên là quá dài và đang bị cắt, chúng ta có thể sửa điều này bằng cách thay đổi loại dữ liệu của cột để nó có thể dài đến 45 kí tự.

```
alter table customers
modify name char(45) not null;
```

Huỷ bỏ các records khỏi cơ sở dữ liệu: Sử dụng câu lệnh **DELETE**, cấu trúc như sau:

```
DELETE FROM table
[WHERE condition]
```

Ví dụ:

```
delete from customers
where customerid=5;
```

Chú ý:nếu bạn viết *DELETE FROM table*; thì tất cả các hàng trong bảng sẽ bị huỷ bỏ.

Huỷ các bảng: Bạn có thể huỷ toàn bộ một bảng bằng cách sử dụng câu lệnh **DROP TABLE**. Cấu trúc này rất đơn giản, nó như sau:

```
DROP TABLE table;
```

Huỷ toàn bộ cơ sở dữ liệu: Bạn có thể thực hiện điều này bằng cách sử dụng câu lệnh **DROP DATABASE**, cấu trúc như sau:

```
DROP DATABASE database;
```

Tạo và huỷ Index: Index là những phương tiện quan trọng để tăng tốc độ truy cập các nội dung của các bảng, thông thường là dùng cho các câu truy vấn mà liên quan đến kết nối nhiều bảng với nhau. MySQL cho phép bạn tạo ra index một cách khá linh hoạt. Bạn có thể index các cột riêng rẽ hay kết nối các cột lại với nhau. Bạn cũng có thể có nhiều hơn một index trong một bảng nếu bạn muốn có thể tra cứu giá trị một cách nhanh chóng từ các cột khác nhau của một bảng.

Tạo Index: Bạn có thể tạo index cho một bảng mới khi bạn dùng **CREAT TABLE**, hoặc có thể thêm index cho các bảng đang tồn tại với **CREAT INDEX** hay **ALTER TABLE**. Bạn phải chỉ định rằng index có thể hay không thể chứa giá trị lặp lại. Nếu nó không thể, index nên được tạo như là một khoá chính (**PRIMARY KEY**) hay là một index duy nhất **(UNIQUE INDEX)**. Để thêm một index cho một bảng đang tồn tại, bạn có thể sử dụng **ALTER TABLE** hay **CREAT INDEX**. Trong đó lệnh **ALTER TABLE** linh hoạt hơn bởi bạn có thể sử dụng nó để tạo ra một index thông thường, một index duy nhất, hay một khoá chính. Cú pháp như sau:

```
alter table tablename add index indexname(column_list)
alter table tablename add unique indexname(column_list)
```

```
alter table tablename add primary key(column_list)
```

CREAT INDEX có thể thêm một index thông thường hay một index duy nhất cho một bảng:

```
creat unique index index_name on tablename(column_list)
creat index index_name on tablename(column_list)
```

Để tạo một index cho một bảng mới khi bạn sử dụng câu lệnh **CREAT TABLE**, bạn có thể sử dụng cú pháp tương tự như đã làm với **ALTER TABLE**, nhưng bạn phải chỉ định các mệnh đề tạo index trong phần câu lệnh **CREAT TABLE**. Cụ thể như sau:

```
creat table tablename(
......
index index_name(column_list),
unique index_name(column_list),
primary key (column_list),
...
)
```

Hủy bỏ Index: Bạn có thể huỷ bỏ index bằng cách sử dụng một trong hai câu lệnh **DROP INDEX** hay **ALTER TABLE**. Cú pháp như sau:

```
DROP INDEX index_name on table_name
     ALTER TABLE table_name DROP INDEX index_name
     ALTER TABLE table_name DROP PRIMARY KEY
```

Chú ý:Index có thể bị tác động nếu bạn huỷ các cột trong một bảng.

III. Các hàm PHP sử dụng để truy cập cơ sở dữ liệu MySQL

int mysql_affected_rows([resource link_identifier]): trả về số lượng các hàng bị tác động bởi các câu truy vấn cuối cùng **INSERT**, **UPDATE** hay **DELETE** kết hợp với định danh liên kết *(link_identifier).* Nếu nhận dạng liên kết không cụ thể thì liên kết cuối cùng được mở ra bởi *mysql_connect( ).* Chú ý khi bạn đang sử dụng các giao dịch, bạn phải gọi *mysql_affected_rows( )* sau truy vấn **INSERT, UPDATE** hay **DELETE.**

int mysql_character_set_name ([resource link_identifier]):trả về ký tự mặc định gán tên cho kết nối hiện hành.

bool mysql_close ([resource link_identifier]): trả về *TRUE* nếu thành công, FALSE nếu thất bại. Hàm này đóng kết nối đến MySQL server mà kết hợp với nhận dạng liên kết định rõ, nếu *link_identifier* không định rõ thì liên kết mở cuối cùng sẽ được sử dụng. Việc sử dụng hàm này không phải lúc nào cũng cần thiết, như đối với các liên kết mở không liên tục thì nó sẽ tự động đóng khi kết thúc sự thực thi của script. Chú ý hàm này không đóng kết nối liên tục được tạo bởi hàm *mysql_pconnect( ).*

esource mysql_connect ([string server [, string username [, string password [, bool new_link]]]]): trả về một định danh liên kết MySQL khi thành công, còn không trả về giá trị *FALSE*. Hàm này thiết lập một kết nối đến MySQL server. Những giá trị mặc định sau được dùng cho các tham số tùy chọn không khai báo: *server*='localhost:3306', *username*=tên của người sử dụng là chủ của server và *password*= bỏ trống. Nếu lời gọi lần thứ hai được tạo bởi *mysql_connect( )* với cùng các tham số mà không có liên kết mới nào được thiết lập,nhưng thay vào đó một liên kết định danh của cổng đang được mở sẽ được trả về. Tham số *new_link* mô tả hành vi này và làm cho hàm *mysql_connect( )* luôn mở một liên kết mới, ngay cả khi nếu hàm *mysql_connect( )* bị gọi trước với cùng các tham số. Liên kết đến server sẽ bị đóng ngay khi việc thực thi của script kết thúc,trừ khi nó bị đóng sớm hơn bởi hàm gọi *mysql_close( ).*

ool mysql_create_db (string database name [, resource link_identifier]): tạo một cơ sở dữ liệu mới trên server kết hợp với liên kết định danh được chỉ định. Trả về giá trị *TRUE* nếu thành công, ngược lại trả về *FALSE*.

bool mysql_data_seek (resource result_identifier, int row_number): di chuyển con trỏ hàng bên trong của kết quả MySQL kết hợp với kết quả định danh cụ thể để chỉ đến số hàng được chỉ định. Nó sẽ trả về *TRUE* nếu thành công, ngược lại trả về *FALSE*.*Row_number* bắt đầu tại *0*, và nên có giá trị trong khoảng từ *0* đến *mysql_num_rows - 1*.

string mysql_db_name (resource result, int row [, mixed field]): lấy tham số đầu tiên là con trỏ kết quả từ việc gọi hàm *mysql_list_dbs( )*. Tham số hàng là chỉ mục đến kết quả gán. Nếu xảy ra lỗi, thì trả về *FALSE*.Sử dụng hàm *mysql_errono( )* và *mysql_error( )* để xem bản chất lỗi.

esource mysql_db_query (string database, string query [, resource link_identifier]): chọn một cơ sở dữ liệu và thực thi một truy vấn trên nó. Nếu liên kết định danh tùy chọn không được chỉ định hàm sẽ cố tìm ra một liên kết mở đến MySQL server và nếu không có liên kết nào được tìm thấy thì nó sẽ tạo ra một liên kết như nếu hàm *mysql_connect( )* đã được gọi không có tham số nào. Hàm sẽ trả về một kết quả MySQL dương, hay là *FALSE* nếu bị lỗi.

ool mysql_drop_db (string database_name [, resource link_identifier]): huỷ bỏ toàn bộ cơ sở dữ liệu từ server kết hợp với liên kết định danh chỉ định. Trả về *TRUE* nếu thành công, *FALSE* nếu gặp lỗi.

nt mysql_errno ([resource link_identifier]): trả về số lỗi từ hàm MySQL cuối cùng, hay là *0* nếu không có lỗi nào.

tring mysql_error ([resource link_identifier]): trả về văn bản lỗi từ hàm MySQL cuối cùng hay ' ' nếu không có lỗi nào.

tring mysql_escape_string (string unescaped_string): tránh *unescaped_string* để nó có thể ở nơi an toàn trong *mysql_querry( )*.

rray mysql_fetch_array (resource result [, int result_type]): trả về một mảng tương đương với hàng tìm nạp, hay là *FALSE* nếu không có hàng nào.

rray mysql_fetch_assoc (resource result): trả về một mảng kết hợp tương đương với hàng tìm nạp, hay là *FALSE* nếu không có hàng nào.

bject mysql_fetch_field (resource result [, int field_offset]): trả về một đối tương chứa thông tin các trường. Nó thường được dùng để lấy thông tin về các trường trong một kết quả truy vấn.

rray mysql_fetch_lengths (resource result): trả về một mảng tương đương với chiều dài của mỗi trường trong hàng cuối cùng được tìm nạp bởi *mysql_fetch_row()* hay trả về *FALSE* nếu lỗi.

bject mysql_fetch_object (resource result): trả về một đối tượng với các thuộc tính tương đương với hàng tìm nạp hay *FALSE* nếu không có hàng nào.

rray mysql_fetch_row (resource result): trả về một mảng tương đương với hàng tìm nạp, nếu không có hàng nào thì trả về *FALSE*.

tring mysql_field_flags (resource result, int field_offset): trả về trường các cờ của trường được chỉ định.

nt mysql_field_len (resource result, int field_offset): trả về chiều dài của trường được chỉ định.

tring mysql_field_name (resource result, int field_index): trả về tên của trường chỉ mục được chỉ định.

nt mysql_field_seek (resource result, int field_offset): tìm kiếm offset của trường được chỉ định.

tring mysql_field_table (resource result, int field_offset): trả về tên của bảng mà trường được chỉ định ở trong đó.

tring mysql_field_type (resource result, int field_offset): trả về loại của trường được chỉ định .

ool mysql_free_result (resource result): giải phóng tất cả bộ nhớ kết hợp với định danh kết quả *result*. Hàm này chỉ được gọi khi bạn quan tâm đến bao nhiêu bộ nhớ đang được sử dụng cho các câu truy vấn mà trả về các tập hợp lớn các kết quả. Tất cả kết quả bộ nhớ kết hợp sẽ tự động được giải phóng khi script kết thúc thực hiện. Trả về *TRUE* nếu thành công, *FALSE* nếu thất bại.

tring mysql_get_client_info (void): trả về chuỗi biểu thị phiên bản thư viện của client.

tring mysql_get_host_info ([resource link_identifier]): trả về chuỗi mô tả loại kết nối được dùng cho kết nối *link_indentifier*,bao gồm cả tên của server host. Nếu *link_indentifier* bị bỏ qua, thì kết nối được mở cuối cùng sẽ được sử dụng.

nt mysql_get_proto_info ([resource link_identifier]): trả về phiên bản giao thức được dùng bởi kết nối *link_identifier*. Nếu *link_identifier* bị bỏ qua, kết nối mở cuối cùng sẽ được dùng.

tring mysql_get_server_info ([resource link_identifier]): trả về phiên bản của server được dùng bởi kết nối *link_identifier*, nếu *link_identifier* bị bỏ qua, kết nối được mở cuối cùng sẽ được sử dụng.

- *string mysql_info ([resource link_identifier])*: trả về thông tin chi tiết về truy vấn cuối cùng sử dụng *link_identifier*.Nếu *link_identifier* không được chỉ định, liên kết cuối cùng được mở sẽ được thừa nhận.
- *int mysql_insert_id ([resource link_identifier])*: trả về ID được tạo cho bởi cột *AUTO_INCREMENT* bởi câu lệnh truy vấn *INSERT* trước đó sử dụng *link_identifier* được cho. Nếu *link_identifier* không được chỉ định, liên kết mở cuối cùng sẽ được thừa nhận.
- *resource mysql_list_dbs ([resource link_identifier])*: trả về con trỏ kết quả chứa các cơ sở dữ liệu có sẵn từ daemon mysql hiện hành.
- *resource mysql_list_fields (string database_name, string table_name [, resource link_identifier])*: truy xuất thông tin về tên các bảng cho trước. Các tham số là tên các cơ sở dữ liệu hay tên các bảng.
- *resource mysql_list_processes ([resource link_identifier])*: trả về con trỏ kết quả mô tả các luồng hiện hành của server.
- *resource mysql_list_tables (string database [, resource link_identifier])*: lấy tên cơ sở dữ liệu và trả về con trỏ kết quả.
- *int mysql_num_fields (resource result)*: trả về số lượng các trường trong tập hợp kết quả.
- *int mysql_num_rows (resource result)*: trả về số lượng các hàng trong tập hợp kết quả. Hàm này chỉ đúng cho các câu lệnh *SELECT*.
- *resource mysql_pconnect ([string server [, string username [, string password]]])*: thiết lập kết nối đến MySQL server.
- *bool mysql_ping ([resource link_identifier])*: kiểm tra kết nối đến server có làm việc hay không. Nếu nó không làm việc thì một kết lại tự động thực hiện. Hàm này có thể được dùng bởi các script mà giữ nguyên trạng thái nhàn rỗi trong một lúc để kiểm tra xem server có đóng kết nối và kết nối lại nếu cần thiết hay không. Hàm này sẽ trả về *TRUE* nếu kết nối đến server đang làm việc nếu không thì trả về *FALSE*.
- *resource mysql_query (string query [, resource link_identifier [, int result_mode]])*: gởi một truy vấn đến cơ sở dữ liệu hiện hành đang hoạt động trên server mà kết hợp với định danh liên kết được chỉ định. Nếu *link_identifier* không được chỉ định, liên kết mở cuối cùng sẽ được thừa nhận. Nếu không có liên kết nào được mở, hàm sẽ cố thiết lập một liên kết nếu *mysql_connect( )* đã được gọi không có tham số nào và dùng nó.
- *string mysql_real_escape_string (string unescaped_string [, resource link_identifier])*: tránh được các kí tự đặc biệt trong *unescaped_string*, trả về tập hợp kí tự hiện hành của kết nối.
- *mixed mysql_result (resource result, int row [, mixed field])*: trả về nội dung của một cell từ tập hợp kết quả MySQL. Tham số trường có thể là offset của trường hay là tên trường hay là tên bảng của trường.tên trường. Nếu tên cột có tên hiệu thì sử dụng tên hiệu thay cho tên cột.
- *bool mysql_select_db (string database_name [, resource link_identifier])*: gán cơ sở dữ liệu hiện hành hoạt động trên server mà kết hợp với định danh liên kết chỉ định. Nếu không có định danh liên kết nào được chỉ định thí liên kết mở cuối cùng được thừa nhận. Nếu không có liên kết nào được mở thì hàm sẽ cố thiết lập mộ liên kết như khi hàm *mysql_connect( )* đã được gọi không có tham số và sử dụng nó
- *string mysql_stat ([resource link_identifier])*: trả về tình trạng hiện hành của server.
- *string mysql_tablename (resource result, int i)*: trả về tên bảng của trường, lấy con trỏ kết quả trả về bởi hàm *mysql_list_table( )* cũng như chỉ mục integer.
- *int mysql_thread_id ([resource link_identifier])*: trả về ID của thread hiện hành. Nếu kết nối bị mất và bạn kết nối lại với *mysql_ping( )*, thì ID của thread sẽ bị thay đổi. Điều này có nghĩa là bạn không nên lấy ID của thread và chứa nó cho sau này. Bạn nên lấy nó khi nào bạn cần nó.
- *resource mysql_unbuffered_query (string query [, resource link_identifier [, int result_mode]])*: gởi một truy vấn SQL *query* đến MySQL mà không cần các hàng kết quả tìm nạp và đệm một cách tự động như *mysql_query( )*.

Sau đây là ví dụ đơn giản trình bày cách kết nối, thực thi câu truy vấn, in các hàng kết quả và ngừng kết nối từ cơ sở dữ liệu MySQL.

```
<?php
    /* Kết nối,chọn cơ sở dữ liệu */
    $link = mysql_connect("mysql_host", "mysql_user", "mysql_password")
        or die("Could not connect");
    print "Connected successfully";
    mysql_select_db("my_database") or die("Could not select database");
```

```
        /* Thực hiện câu truy vấn SQL */
        $query = "SELECT * FROM my_table";
        $result = mysql_query($query) or die("Query failed");
        /* In kết quả ra trang *.HTML */
        print "<table>\n";
        while ($line = mysql_fetch_array($result, MYSQL_ASSOC)) {
            print "\t<tr>\n";
            foreach ($line as $col_value) {
                print "\t\t<td>$col_value</td>\n";
            }
            print "\t</tr>\n";
        }
        print "</table>\n";
        /* Free resultset */
        mysql_free_result($result);
        /*Đóng kết nối */
        mysql_close($link);
    ?>
```

Sự cần thiết của một cơ sở dữ liệu không chỉ bởi nhu cầu phục vụ, tạo ra các nguồn tài liệu một cách phong phú, mà bởi sự tăng lên của nhu cầu truy cập các thông tin "trực tiếp" trong ngày bằng việc sử dụng một giao diện đơn giản và hợp nhất. Bạn có thể cho phép mọi người trong các văn phòng bán hàng truy vấn một cơ sở dữ liệu hàng tồn kho. Cơ sỡ dữ liệu này có thể được xây dựng khác, chạy trên một hệ điều hành hoàn toàn khác với máy tính của người dùng.

Với một web server có cấu hình tốt, một cơ sở dữ liệu server(chẳng hạn như MySQL), và một vài "phép thần thông" của PHP, chỉ những thứ người dùng cần mới có thể có trên trình duyệt web.

Cuối cùng: nếu bạn cần các thông tin luôn năng động, cập nhật, bạn cần phải phát triển một cơ sở dữ liệu đằng sau ứng dụng web.Với một cơ sở dữ liệu phía sau, web site của bạn sẽ đạt được một trình độ phức tạp và linh hoạt, nơi mà dữ liệu được tạo ra và sử dụng, thông tin được làm cho có ý nghĩa với người dùng mà không cần phải bắt buộc nó phải "đúc nặn" lại cho vừa với cách hiển thị hay tổ chức nó.

IV. Kiến trúc của một ứng dụng Web Database

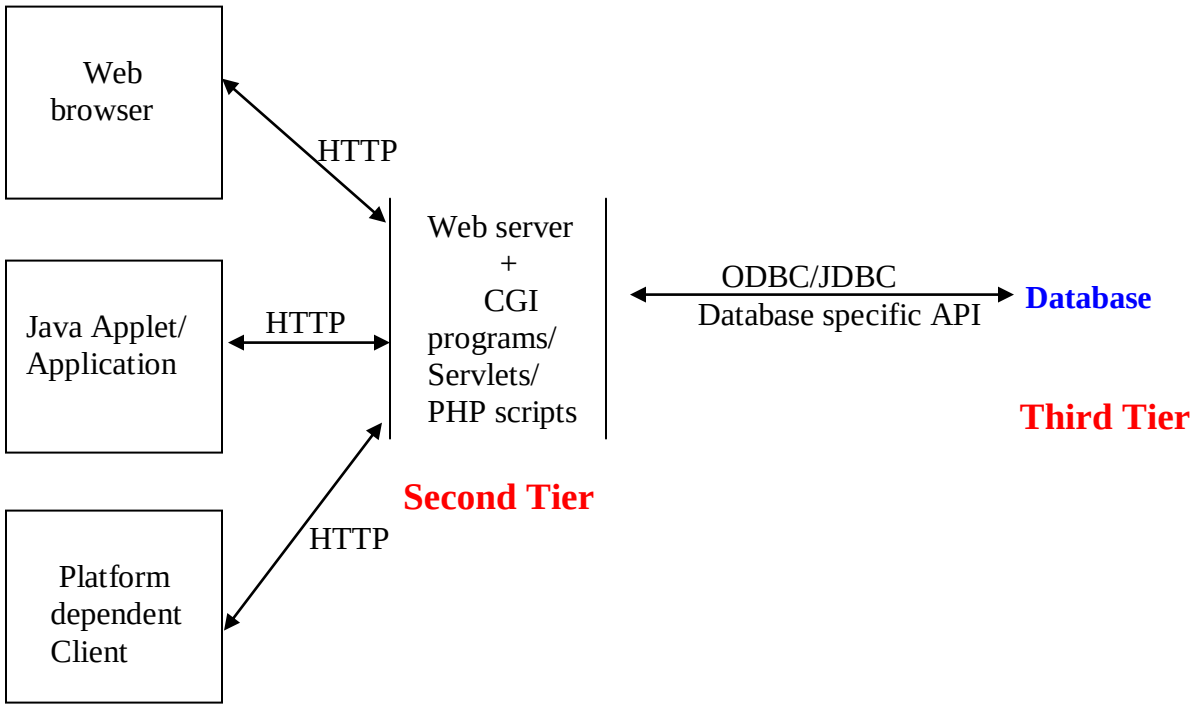
Các phần cơ bản của ứng dụng Web Database là:

- Client: trình duyệt web của người dùng, một java applet, một ứng dụng java, thậm chí là một platform phụ thuộc vào chương trình của client.
- Ứng dụng logic: mã hoá các thuật toán sử dụng trong các scripts CGI, các modul đặc biệt của web server, thậm chí một ứng dụng phụ thuộc vào server.
- Kết nối database: the database API, hay là các giao thức kết nối thông thường như ODBC hay JDBC.
- Database server: RDBMS, ODBMS...

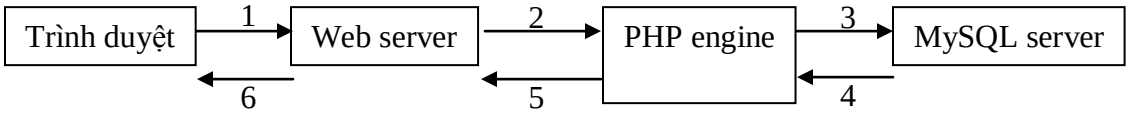
Việc thực hiện các ứng dụng đó có thể được thực hiện bởi sử dụng mô hình nhiều lớp, bởi một hay nhiều lớp có thể kết hợp với nhau. Cách thực hiện thông thường nhất là hệ thống ba lớp.

1. Lớp thứ 1: web client(ví dụ như trình duyệt web của người dùng).
2. Lớp thứ 2: web server, các scripts của CGI, kết nối API đến database.
3. Lớp thứ 3: database server (ví dụ MySQL server).

Mô hình các lớp cụ thể như sau:



Đối với Web site xây dựng bằng PHP&MySQL thì mô hình ứng dụng web database như sau:



Một giao dịch chính của Web database bao gồm các giai đoạn sau:

1. Trình duyệt Web của người dùng sẽ đưa ra một yêu cầu HTTP cho một trang web riêng biệt... Ví dụ, muốn viết ý kiến của mình vào guestbook, sau khi điền vào các thông tin cần thiết theo yêu cầu vào trong các form HTML, người dùng bấm vào nút "Gởi đi", thì thông tin vừa mới viết đó sẽ được hiển thị trên trang index.php,
2. Web server sẽ nhận được một yêu cầu cho trang index.php, nó sẽ truy xuất file này và chuyển nó qua PHP engine để xử lý.
3. PHP engine bắt đầu phân tích script. Bên trong script là một yêu cầu kết nối đến cơ sở dữ liệu và thực thi một truy vấn. PHP mở một kết nối đến MySQL server và gởi trên một truy vấn chính xác.
4. MySQL server nhận một truy vấn dữ liệu và xữ lý nó, sau đó gởi các kết quả về lại cho PHP engine.
5. PHP engine hoàn thành việc chạy script mà thường liên quan đến định dạng các kết quả truy vấn trong HTML. Sau đó nó trả kết quả HTML về lại Web server.
6. Web server chuyển HTML về lại trình duyệt, nơi mà bạn có thể thấy danh sách các quyển sách mà bạn đã yêu cầu.

Quá trình này là căn bản tương tự đối với các script engine hay các database server mà chúng ta dùng. Thường thì một phần mềm Web server, PHP engine, và database server tất cả cùng chạy trên cùng một máy. Tuy nhiên, việc database server chạy trên một máy khác cũng là điều khá thông dụng. Bạn có thể làm được điều này bởi các lý do bảo mật, tăng dung lượng .

Hướng dẫn học lập trình

PHP &MySQL

Cài đặt Web server - bước đầu tiên để học PHP!

"Trường học" vừa khai giảng, trong lúc đợi bài học đầu tiên ra lò, cho phép tớ được "múa rìu qua mắt thợ cái"!

Như bác ngocha85 đã nói, để học PHP và MySQL, một trong những thứ cần chuẩn bị là web server chạy trên PC của mình. Để cho nhanh chóng, theo tớ tốt nhất nên cài bộ XAMPP.

Câu hỏi 1: XAMPP là gì?

Trả lời: XAMPP giống với WAMP, nghĩa là người mới học ko cần phải biết cách cài đặt riêng lẻ từng thành phần như Apache, PHP và MySQL. Chỉ cần download một gói về là xong.

Các tính năng có trong XAMPP:

1. Apache 2 => server
2. PHP 5 => ngôn ngữ lập trình
3. MySQL => cơ sở dữ liệu
4. Webalizer => quản lý statistic của site
5. Mercury => giả lập gửi email
6. FileZilla => giả lập FTP server
7. Rất nhiều tính năng chuyên sâu khác...

Câu hỏi 2: Tại sao ko dùng WAMP?

Trả lời: Vì cái này tớ chưa dùng bao giờ + Cái này bác ngocha85 chắc sẽ giới thiệu => tránh đụng hàng là hơn. Với lại cái XAMPP này theo tớ cũng rất hay, thậm chí ko cần cài đặt, chỉ cần copy và chạy.

Câu hỏi 3: Down XAMPP ở đâu?

Trả lời: Ở đây: http://www.apachefriends.org/en/xampp.html
Có đủ bộ XAMPP cho Windows, Linux, MacOS và cả Solaris, vì vậy mọi hệ điều hành nó đều chấp tất!

Câu hỏi 4: Cài đặt và sử dụng XAMPP như thế nào?

Trả lời: Sau khi tải về, bạn sẽ có 1 file zip. Giải nén file đó ra 1 thư mục bất kỳ, ví dụ C:\XAMPP. Để chạy web server, bạn kích hoạt file xampp_control.exe, bấm nút Start bên cạnh Apache và nút close để XAMPP Control tự động chuyển xuống system tray.

Ngoài ra, bạn có thể khởi động MySQL nếu dùng cơ sở dữ liệu, FileZilla nếu dùng FTP và Mercury nếu dùng email.

Để biết chắc web server đã chạy đúng, bạn mở trình duyệt web của mình, gõ http://localhost vào thanh Address, sau đó enter. Một trang thông báo sẽ hiện ra, cho biết quá trình "cài đặt" đã hoàn tất.

Câu hỏi 5: Làm thế nào để chạy các script viết bằng PHP?

Trả lời: Bạn cho script vào thư mục C:\XAMPP\htdocs\ sau đó gọi file đó qua URL http://localhost/script_name.php

Vậy là hết "bài chuẩn bị cho bài khởi động" của bác ngocha85 sẽ post trong vài ngày tới. Tèn tén ten!

Thừa thắng xông lên, tớ làm luôn bài "Hello World".

Bài 1: Nói "hello world" với PHP

Cần chuẩn bị những gì?

1. Web server cần đảm bảo sẵn sàng. Apache được khởi động theo bài post ở trên.
2. Một script editor. Cái này có rất nhiều, như PHP Designer, Dev-PHP, ... Thậm chí dùng notepad cũng được. Nhưng tốt nhất nên dùng một editor có hỗ trợ unicode. Như tớ dùng SCiTE.
Bạn vào trang này để xem list và review các PHP editor: http://www.php-editors.com/
3. 5 phút thời gian rảnh rỗi.

Bắt đầu!!!

1. Tạo một file mang tên "helloworld.php" trong thư mục htdocs. Mở file đó bằng script editor.
2. Gõ đoạn code sau vào editor:

PHP Code:

```php
<?php
   echo "Hello World!";
?>
```

3. Mở trình duyệt web, gõ http://localhost/helloworld.php [enter].
4. Nhắm mắt lại trong 0.0001 giây. Nếu mở mắt ra mà bạn thấy dòng chữ Hello World là đã thành công rồi đó!

Giải thích

1. Dòng thứ nhất của file helloworld.php là "<?php" và dòng cuối cùng là "?>". Đây là 2 thẻ (tag) để báo cho server biết điểm bắt đầu và kết thúc của một đoạn code PHP. Nói cách khác, bằng cách này bạn có thể nhúng code PHP trong bất cứ file HTML có sẵn nào. Khi thực thi file PHP, web server sẽ chỉ thực hiện những đoạn code đặt trong 2 thẻ này và bỏ qua tất tần tật những phần còn lại.
2. Dòng 2 là một lệnh của PHP: Lệnh echo. Lệnh này làm nhiệm vụ in một xâu ra ngoài màn hình. Cần nhớ một lệnh PHP luôn kết thúc bằng dấu chấm phẩy ";". Nếu thiếu dù chỉ một dấu chấm phẩy, code của bạn sẽ ko chạy và dừng lại biểu tình ngay.
3. Cũng ở dòng 2, xâu "Hello World" được đặt trong dấu ngoặc kép. Nếu ko, sẽ có lỗi.

Một vài câu hỏi

1. Có cần thiết phải trình bày như trên ko?
=> Ko. Bạn có thể trình bày code theo bất cứ cách nào bạn muốn. Lùi vào 10 dấu cách, mỗi dòng cách nhau 3 hàng, ... Điều đó là tùy bạn. Tuy nhiên cần phải viết code cho thật dễ đọc và dễ hiểu để tiện cho việc sửa đổi và chia sẻ code sau này.

2. Có cách nào báo hiệu một đoạn code PHP ngoài cách dùng <?php ko?
=> Có. Nhiều cách là đằng khác. Ví dụ bạn có thể viết

PHP Code:

```
<?
   // Code ở đây
?>
```

Tuy nhiên các cách khác đều ít thông dụng và được khuyến cáo ko nên sử dụng.

3. Có thể đặt xâu Hello World trong dấu ngoặc đơn ko?
=> Có thể. Bạn có thể dùng dấu ngoặc kép và dấu nháy đơn để chứa xâu. Sự khác nhau giữa chúng sẽ được thảo luận sau.

4. Nếu trong xâu cũng có dấu ngoặc / xâu là một đoạn văn bản rất dài thì sao?
=> Ko có gì phải lo lắng. Cái gì cũng có cách giải quyết. Vấn đề là cách đó ko nằm trong bài học hôm nay. Hết 5 phút rồi, bạn hãy nghỉ ngơi đã. :P

Bài tập

Vì Bài 1 hết sức đơn giản, chỉ theo tinh thần Hello World nên bài tập cũng sẽ chỉ có bài, và cũng rất rất đơn giản.

Hãy cho biết lỗi sai trong các đoạn code sau:

1.

PHP Code:

```
echo "Hello World!";
```

2.

PHP Code:

```
<?php
    echo "Hello World!"
?>
```

3.

PHP Code:

```
<?php
    echo "Hello World!';
?>
```

4.

PHP Code:

```php
<?php
    echo "Hello World!";
?>
```

Bài 2 - Mục 1: Lưu trữ dữ liệu trong PHP. Vài điều cần nói về biến.

Trước khi bắt đầu bài 2, tớ xin trình bày về cách chú thích (comment) trong PHP. Đây có thể coi là một kỹ năng cũng được, vì bạn rất KHÔNG NÊN viết code mà ko có chú thích. Có thể đoạn code rất dễ hiểu vào thời điểm viết, nhưng nếu ko có chú thích, chỉ vài tháng sau bạn có thể quên ngay mình đã viết cái gì. Viết chú thích ngay vào thời điểm code là cách tốt nhất.

Trong PHP, một dòng chú thích được đặt sau 2 dấu sổ chéo //

Ví dụ

PHP Code:

```php
// Đây là một dòng chú thích
```

Nếu chú thích của bạn dài hơn 1 dòng, bạn có thể để nó trong 1 block, mở đầu bằng /* và kết thúc bằng */

PHP Code:

```php
/*
   Chú thích dòng thứ nhất
   Thứ 2
   Thứ 3
   Vân vân...
*/
```

Còn một cách nữa, KHÔNG phổ biến (ít ra là tớ thấy thế), đó là chú thích đặt sau dấu #. Chú thích này cũng chỉ cho phép 1 dòng giống như //

Một điều khác cũng rất cần chú ý đó là PHP ko cho phép đặt chú thích trong chú thích (nested comment).

Ta bắt đầu vào Bài 2.

Bài 2: Lưu trữ dữ liệu trong PHP

Khi bắt tay vào lập trình một chương trình, hiển nhiên ta sẽ cần phải lưu trữ dữ liệu. Cụ thể, dữ liệu có thể được lưu trữ bằng biến (variable). Khác với các ngôn ngữ lập trình khác, trong PHP các biến ko cần phải khai báo (declare) trước khi sử dụng. Để sử dụng biến, bạn chỉ cần gán (assign) cho nó một giá trị (value). Biến sẽ tự động được tạo. Cực kỳ đơn giản và nhanh chóng!

1. Biến. Khai báo. Đặt tên.

Biến trong PHP bắt đầu bằng dấu dollar ($), theo sau là tên biến. Tên biến có thể bắt đầu bằng dấu gạch dưới (_ gọi là underscore) hoặc chữ cái. Tiếp sau đó là các chữ cái, số hoặc lại là dấu gạch dưới. Một số ký tự mở rộng (extended character) có thể được sử dụng, nhưng tốt nhất là nên tránh.

Một số ví dụ về biến ĐÚNG: $uds, $update_softs, $uds_has_more_than_26000_members

Biến sai: abc vì thiếu dấu dollar, $124adfd vì bắt đầu bằng số

Cũng cần thảo luận thêm một chút: Cũng vì sự dễ dãi trong việc ko phải khai báo biến nên sẽ có lúc bạn gõ nhầm tên biến. Ví dụ $uds gõ thành $usd (ặc!)

Ví dụ:

PHP Code:

```php
<?php
   $uds = "Welcome to UDS!";
   echo $uds;
?>
```

May mắn làm sao, từ bản PHP 5 trở lên, sẽ có một cảnh báo (warning) khi bạn chạy script, cho biết bạn chưa gán giá trị cho biến $usd.

À, còn một vấn đề chưa nói đến: Đó là trong PHP, tên biến CÓ phân biệt chữ hoa chữ thường (case-sensitive). Nghĩa là $uds hoàn toàn khác với $UDS hay $uDs. Nói chung nên tránh việc đặt tên biến chỉ khác nhau cách viết hoa thường này, vừa đỡ mất công giữ Shift, vừa đỡ nhớ nhầm tên biến.

Mục 2 sẽ mang tên **Một số kiểu dữ liệu trong PHP**. Mọi người đón đọc nhá!

Bài tập

Trong các biến sau đây, biến nào được đặt tên đúng, biến nào bị đặt tên sai:

1. this_is_a_variable
2. $yet another variable
3. $simplevariable
4. $blah_blah_blah_123456789_____
5. $123456789_____abacabadfskdjsfksdfkdserwuewrjfdksj fdksljf
6. $^^
7. $___A___VARIABLE____
8. $THiS_iS_THe_LaST_ONe

Bài 2 - Mục 2: Lưu trữ dữ liệu trong PHP. Một số kiểu dữ liệu cơ bản [updated]

Ngoài lề một chút: Lúc đầu tớ cũng ko định tách Bài 2 ra làm mấy thread, nhưng nếu để như thế kia thì dài quá, sợ đọc theo các bác mệt mắt => nản lòng.

Ta có một số kiểu dữ liệu cơ bản sau đây trong PHP:

a. Kiểu số (number)
b. Kiểu xâu (string)
c. Kiểu boolean (boolean)

a. Kiểu số

Trong kiểu số (lại) có 2 kiểu cơ bản khác: Số nguyên (int) và số thực (float). Số nguyên có thể biểu diễn bằng số thập phân (hệ 10 - decimal), hệ 8 (octal) và hệ 16 (hexadecimal).

Ví dụ ta gán giá trị cho một số biến kiểu NGUYÊN như sau:

PHP Code:

```
<?php
   $a = 27;
   $b = -27;
   $c = 027;
   $d = -027;
   $e = 0x27;
   $f = -0x27;
?>
```

Ở ví dụ trên, cả 6 biến từ $a đến $f đều có giá trị là 27 hoặc -27. Tuy nhiên, với biến $a và $b, ta dùng kiểu biểu diễn số thập phân (viết như số ta viết hàng ngày). Với $c và $d, dùng kiểu số hệ 8 (bắt đầu với chữ số 0). Với $e và $f dùng kiểu hệ 16 (bắt đầu với chữ số 0 và chữ cái x).

Nếu đã từng học qua Pascal, chắc chắn bạn sẽ hỏi tớ: Thế nếu tớ dùng 1 biến kiểu int, gán cho nó một giá trị cao bằng max của int, thì khi đem số đó cộng với 1, giá trị có bị chuyển thành âm do tràn số (overflow) ko?

Câu trả lời là ko. Một biến kiểu int có giá trị cực lớn trong PHP là 2147483647, khi cộng 1 vẫn sẽ trả giá trị đúng là 2147483648, nhưng lần này sẽ thuộc kiểu float. Nói cách khác, PHP tự chuyển số bị tràn lên kiểu float.

Nếu thích đặt câu hỏi, chắc chắn (lại một lần nữa) bạn sẽ hỏi tớ: Sao cậu biết điều ấy?

Câu trả lời rất đơn giản: Bạn hãy cùng tớ làm ví dụ với đoạn code sau:

PHP Code:

```
<?php
   $a = 2147483647;
   var_dump($a);
   $a = $a + 1;
```

```
  var_dump($a);
?>
```

Sau khi chạy script, kết quả trả về sẽ là

int(2147483647) float(2147483648)

=> Đúng như tớ nói nhá!

Tớ xin giải thích như thế này:

Ở dòng thứ nhất, ta đem gán giá trị 2147483647 cho $a. Đây là một giá trị cực to, nhưng vẫn nằm trong int, vì vậy $a sẽ thuộc kiểu int.
Dòng thứ 2 và thứ 4, ta dùng lệnh var_dump($a); Đây là lệnh in ra kiểu và giá trị của một biến trong PHP. Chú ý nhé, lệnh này khá phổ biến và hay được dùng để debug code.
Ở dòng thứ 3, ta dùng lệnh gán $a = $a + 1; Với các bạn đã học lập trình, điều này chẳng có gì khó hiểu. Sau khi thực thi lệnh, $a sẽ mang giá trị của $a cộng thêm với 1. Còn nếu (chẳng may) bạn chưa học lập trình bao giờ, thì tớ (lại) xin giải thích như thế này:
- Dấu bằng ở đây là lệnh gán, đem giá trị của vế phải gán cho vế trái, chứ ko phải là dấu bằng trong biểu thức toán học mà mình vẫn học. Do đó, ko có gì là trái với lẽ tự nhiên cả. :P

Một điều khác mà bạn nên nhớ, đó là hãy THẬT cẩn thận khi sử dụng số kiểu float trong PHP. Nó luôn chỉ là những giá trị xấp xỉ, và ko hề chính xác tuyệt đối. Do đó tốt nhất là chuyển số float sang int khi có thể. Cách làm sẽ được thảo luận sau.

Giờ ta sang kiểu xâu.

b. Kiểu xâu

Định nghĩa một cái nào: Xâu là một chuỗi các ký tự. Một câu tớ xì pam là một xâu. Cả cái bài viết này cũng có thể là một xâu.

Để sử dụng xâu, có 3 cách (hic, bắt đầu phức tạp rồi => mọi người đứng dậy vươn vai cái cho tỉnh táo! :P):

Cách 1 là dùng nháy đơn.
Cách 2 là dùng ngoặc kép (hay gọi là nháy kép gì cũng được).
Cách 3 là dùng kiểu HEREDOC.

Nói rõ nhé:

Cách 1: Xâu được đặt trong dấu nháy đơn.

PHP Code:

```
<?php
  echo 'Đây là xâu đặt trong dấu nháy đơn';
?>
```

Sẽ có bạn hỏi tớ (sao hỏi nhiều thế!): Trong xâu có thể đặt dấu nháy đơn được ko? Kiểu như

xâu là I'm a student ý.
Câu trả lời là bạn phải thêm một dấu sược (hay suộc gì ý) trước dấu nháy đơn "bất thường" ý. Như thế này:

PHP Code:

```
echo 'Trong nháy đơn lại có một nháy đơn như thế này \', và như thế này nữa \'.';
```

Đặt cái dấu đó (\) gọi là "escape the character". Nói nhỏ nhá: Bài tớ viết hay chèn tiếng Anh vào là để các bạn đỡ "bỡ ngỡ" khi đọc tut hay doc bằng Eng.

Một lần nữa, (lại) có một câu hỏi được đặt ra: Nếu trong xâu cũng có một dấu \ thì sao?
Câu trả lời cũng rất giản dị: Dùng thêm một dấu \ nữa ngay trước dấu \ ý. Như thế này \\

Lần này, sẽ ko có một câu hỏi, mà sẽ là một tiếng thở dài: Sao lắm thứ thế? Còn cái dấu nào phải "escape" như dấu \ và ' ko?
Có. Đó là:
1. \n : Báo hiệu xuống dòng trong PHP. Giống như <br /> trong HTML.
2. \t : Thay mặt cho Tab
3. \$: Dấu dollar (tránh "cạnh tranh lành mạnh" với tên biến mà! :P)
4. ... Để gặp nói sau. Nói nhiều e "tẩu hỏa nhập ma" chết!

Quên mất, trừ \' và \\, mấy cái escape này chỉ dùng trong trường hợp xâu đặt trong dấu ngoặc kép.

Hờ hờ, lại quên một điều phải nói trước khi chuyển qua phần kế tiếp: Nếu trong xâu ta ko thèm dùng dấu \, cũng kóc thèm dùng dấu ', mà dùng cả \' cho "dân chơi" thì sao?
Trả lời: Thì cứ làm như bình thường thôi. Như thế nè: \\\'. Dấu \ thứ 1 để escape cho dấu \ thứ 2. Dấu \ thứ 3 để escape cho dấu ' cuối cùng. Thường thôi!

Cách 2: Xâu được đặt trong dấu ngoặc kép (hay nháy kép - whatever)

Trường hợp này rất giống với sử dụng dấu nháy đơn đã nói ở trên.

PHP Code:

```
<?php
   echo "Xâu này đặt trong dấu ngoặc kép";
?>
```

Sở dĩ nói RẤT giống mà ko phải HOÀN TOÀN giống vì giữa chúng có điểm khác nhau: Khi thực thi, PHP sẽ tìm và thay thế trong xâu những ký tự đặc biệt được escape (như \n, \t...) như đã nói ở trên, cùng với các biến (nếu có) trong xâu.

Ví dụ:

PHP Code:

```
<?php
   $a = 1;
```

```
    echo "Biến \$a có giá trị là $a";
?>
```

Sẽ cho ta kết quả: Biến $a có giá trị là 1

Trong khi đó, nếu sử dụng dấu nháy đơn:

PHP Code:

```
<?php
    $a = 1;
    echo 'Biến \$a có giá trị là $a';
?>
```

Lại in ra: Biến \$a có giá trị là $a

Điều đó cho thấy: Khi sử dụng dấu nháy đơn, giá trị của biến trong xâu, cùng với các ký tự đặc biệt cần escape sẽ ko được in ra. Các bạn nhớ kỹ điều này nhé!

Ta sang cách thứ 3: Xâu đặt trong cấu trúc HEREDOC

Ở cách 1, PHP sẽ nhận thấy 1 xâu được bắt đầu với dấu nháy đơn thứ nhất và kết thúc với dấu nháy đơn thứ 2. Tương tự với cách 2, nhưng là dấu ngoặc kép.

Ở cách 3 này, PHP sẽ coi một xâu bắt đầu bằng 3 dấu nhỏ hơn viết liền nhau <<<, đi kèm với 1 tên định danh (identifier) tùy ý bạn đặt tên, ví dụ là HERE, kết thúc là tên đó kèm theo dấu ;

Nghe có vẻ hơi phức tạp, nhưng bạn hãy cùng tớ gõ ví dụ sau: (chú ý là chữ HERE có thể thay bằng bất cứ chữ gì, tên bạn chẳng hạn, miễn là nó tuân theo nguyên tắc đặt tên biến của PHP. À, mà nhớ là mở bằng <<<HERE thì phải đóng bằng HERE; nhé, ko được mở cửa ra vào, đóng cửa sổ đâu!)

PHP Code:

```
<?php
    echo <<<HERE
            Xâu &#273;ược ghi ở dòng thứ nhất
    Dòng th&#7913; 2
        Dòng th&#7913; 3
    Vân vân
HERE;
?>
```

Nhìn vào ví dụ trên, bạn có nhận xét gì?

Thứ nhất, xâu ko nhất thiết phải thuộc một dòng. Nó ko nhất thiết phải ngắn gọn, mà có thể dài "tràng giang đại hải" ra mấy chục dòng cũng được. Điều này rất tiện nếu bạn muốn echo một lúc cả một bài thơ chẳng hạn!

Thứ hai, chữ HERE; ở dòng cuối cùng tớ ko căn lề với chữ echo ở dòng 1. Đó là LUẬT, dù tớ thấy hơi "bất công" và "nghiệt ngã" một tí:
- Sau <<<HERE phải xuống dòng. Ko được phép có dù chỉ 1 ký tự trắng (dấu cách ý)
- Trước và sau HERE; cũng thế. Ko được phép có dù chỉ 1 ký tự trằng. Nói cách khác, đừng dại gì căn lề cho dòng này.

Cái gì là LUẬT thì phải THEO, cãi ko được Còn nếu ko theo, PHP sẽ báo lỗi:

Quote:

```
Parse error: parse error, unexpected T_SL in E:\XAMPP\htdocs\test.php on line 2
```

Tớ nói dông dài như vậy là vì đã từng mất bao nhiêu thời gian mới tìm ra được lỗi sai của mình. Chỉ vì một dấu cách mà chương trình đình công, ko thèm chạy! Kinh nghiệm xương máu!

Còn một ý nữa: Nếu từ nãy đến giờ bạn chỉ đọc "chay", ko thực hành thì (chưa chắc) đã nhận thấy: Khi chạy chương trinh, thay vì in ra mấy dòng như trên, PHP lại in mọi thứ ra cùng 1 dòng:

Quote:

```
Xâu được ghi ở dòng thứ nhất Dòng thứ 2 Dòng thứ 3 Vân vân
```

Sửa chữa điều này cũng khá đơn giản:

PHP Code:

```
<?php
   $s = <<<HERE
            Xâu &#273;ược ghi ở dòng thứ nhất
   Dòng th&#7913; 2
      Dòng th&#7913; 3
   Vân vân
HERE;
   echo nl2br($s);
?>
```

Có gì bí ẩn ở đây ko? Thay vì echo thẳng mấy dòng kia ra, ta đi "vòng vèo" một chút bằng cách gán xâu chứa mấy dòng đó cho biến $s, sau đó echo nl2br($s) ra màn hình.

nl2br() được gọi là một hàm (function). Nó nhận một xâu làm tham số (parameter), ở đây là xâu $s, sau đó in ra theo luật: Cứ gặp dấu xuống dòng trong code là chuyển thành dấu xuống dòng trong HTML.

Cái tên nl2br cũng chẳng phải thần chú gì khó nhớ, nó rất giản dị: chỉ là viết tắt của new-line-to-br. New-line là dấu xuống dòng trong code, 2 là to , br là
 (thẻ xuống dòng trong HTML).

Vậy là vấn đề đã được giải quyết. Kết quả in ra đúng như mong đợi:

Quote:

```
Xâu được ghi ở dòng thứ nhất
Dòng thứ 2
Dòng thứ 3
Vân vân
```

Kiểu dữ liệu cơ bản cuối cùng mà tớ sẽ nói tới chính là Kiểu boolean.

c. Kiểu boolean

Đây là kiểu dữ liệu đơn giản nhất trong PHP (đỡ quá!). Ý tưởng rất đơn giản: Mọi thứ chỉ thuộc vào 1 trong 2 loại: Đúng hoặc Sai, Có và Không, 1 và 0. Không có ngoại lệ. Anh ko là True thì sẽ là False. Ở đây ko có chỗ cho người ba phải!!!

Giá trị của biến kiểu boolean là TRUE hoặc FALSE. Hai từ này hoàn toàn ko phân biện hoa thường, vì vậy có thể viết như thế nào cũng được: TRue, tRUe, true, ...

Ví dụ:

PHP Code:

```php
<?php
   $a = TRUE;
   $b = false;
?>
```

Một kiểu dữ liệu đơn giản đồng nghĩa với việc ko cần giải thích nhiều về ví dụ của nó.

Nhưng nó lại ko đồng nghĩa với việc: Kiểu boolean chẳng có gì đáng nói! Thực tế là kiểu này rất hay dùng trong PHP, ví dụ khi tính toán một biểu thức và xem giá trị của nó có lớn hơn một số nào đấy hay ko... (biểu thức điều kiện)

Xin được kết thúc Bài 2 tại đây. Cảm ơn quý vị đã quan tâm theo dõi...

COMING UP NEXT: **Một số hàm cần thiết khi debug code**

Bài 2 - Mục 3: Các kiểu dữ liệu quan trọng khác

Tiếp sau mục 2: Các kiểu dữ liệu cơ bản, tớ xin giới thiệu thêm một vài kiểu dữ liệu quan trọng khác của PHP: Mảng, Đối tượng, Null và Resource (sozy vì 2 kiểu cuối ko rõ dịch như thế nào)

1. Mảng (array)

Mảng được sử dụng khi bạn muốn lưu trữ một số lượng lớn các biến. Một ví dụ hết sức đơn giản: Một lớp có 50 học sinh, và bạn muốn quản lý cả 50 học sinh đó. Để đại diện cho một học sinh, tất nhiên bạn sẽ muốn 1 biến. Nhưng nếu đặt tên là hs1, hs2, ... hs50 thì quả là quá mất thời gian! Và đây chính là lý do để mảng có "đất dụng võ".

Mảng chứa rất nhiều giá trị (value), mỗi giá trị được truy cập nhờ khóa (key). Khóa có thể chỉ là những số đếm thông thường như 1, 2, 3, hay có thể là xâu, như "abc", "def", "ghi". Mảng có khóa là xâu như vậy được gọi là associative array.

Để khai báo một mảng, chúng ta có thể sử dụng cách như ví dụ sau:

PHP Code:

```php
<?php
   $a = array(1, 2, 3, 4);
   $b = array("a", "b", "c");
   $c = array(1, "a", array(3, 4));
?>
```

Như ở ví dụ trên, $a, $b, $c đều là mảng. Mảng $a chứa các số từ 1 đến 4, mảng $b chứa các xâu "a", "b", "c". Còn mảng $c sành điệu hơn, chứa cả số lẫn xâu, thêm cả một mảng ở bên trong nó nữa.

Sau khi khởi tạo giá trị trong mảng $a, mặc định mỗi phần tử (element) trong nó sẽ được gán cho một khóa là số nguyên. Nó bắt đầu từ 0, ko phải là 1. Do đó, phần tử thứ 0 sẽ là 1, thứ 1 sẽ là 2, vân vân.

Ví dụ:

PHP Code:

```php
<?php
   echo $a[2];
?>
```

Sẽ in ra màn hình giá trị 3 - tức là phần tử mang khóa là 2 trong mảng $a.

Như tớ đã nói ở trên, một khóa có thể là một xâu, nghĩa là người ta có thể truy cập mảng $d (chẳng hạn) bằng cách dùng $d["blah"]. Vậy ta khởi tạo giá trị của $d như thế nào?

Rất đơn giản, ta sử dụng toán tử (operator) =>

PHP Code:

```php
<?php
   $d = array("blah" => 1, "abc" => 2, "def" => "ghi");
?>
```

Có thể dễ dàng đoán được: Nếu dùng lệnh echo $d["def"] sẽ cho ra kết quả là "ghi".

Tìm hiểu sâu thêm về Mảng, kiểu dữ liệu mạnh mẽ của PHP, sẽ là phần việc của một Bài học sau này.

Đối tượng (object)

PHP5 là một ngôn ngữ lập trình hướng đối tượng (OO - Object Oriented). Nói một cách đơn giản nhất (nhưng vẫn nghe ù tai nếu bạn chưa nghe về đối tượng bao giờ) thì lập trình hướng đối tượng (OOP - Object Oriented Programming) là việc tạo ra một kiểu dữ liệu mới (đối tượng - object hay lớp - class). Thay vì việc phải tạo một dãy các hàm liên quan đến đối tượng đó, bạn sử dụng thuộc tính (properties) và phương thức (method) trực tiếp của đối tượng ý.

Hãy nhắm mắt vào tưởng tượng. Bạn có một quả bóng bay. Quả bóng ý có những thuộc tính gì? À, rất đơn giản thôi: Đó có thể là kích thước, màu sắc hay độ căng - xẹp của bóng.

Còn phương thức: Quả bóng có thể căng lên, hoặc xẹp đi. Rất dễ dàng phải ko?

Giờ hãy tưởng tượng, bạn có một đối tượng mang tên QB (quả bóng). Để tạo ra một quả bóng, bạn dùng lệnh:

PHP Code:

```
<?php
   $bong = new QB();
?>
```

Quả bóng có kích thước (KT), màu sắc (MS) và độ căng - xẹp (CX). Để $bong mang màu đỏ, bạn có thể viết:

PHP Code:

```
<?php
   $bong->MS = red;
?>
```

Tương tự, nói đến kích thước, độ căng - xẹp của quả bóng, ta có thể dùng $bong->KT, $bong->CX.

Thế còn phương thức? Như đã nói, quả bóng có thể căng lên (CL) hoặc xẹp đi (XD). Để thực thi các phương thức này, ta làm như ví dụ sau:

PHP Code:

```
<?php
   $bong->CL();
?>
```

Tạm dừng việc "cưỡi ngựa xem hoa" phần đối tượng tại đây.

3. Null

Một biến được coi là NULL (ko có giá trị) nếu nó thỏa mãn cả 3 điều kiện sau:
1. Nó được gán là NULL (ko phân biệt hoa thường)
2. Nó chưa bao giờ "được" (hay "bị") gán giá trị.
3. Nó đã bị "xử đẹp" bằng unset - hàm hủy bỏ các biến chỉ định.

Để kiểm tra một biến có là NULL hay ko, ta có thể sử dụng hàm is_null(biến). Ví dụ:

PHP Code:

```
<?php
   $test = NULL;
   echo is_null($test);
?>
```

Cho ra kết quả là 1.

4. Resource

Có những lúc PHP cần xử lý các đối tượng như kết nối cơ sở dữ liệu hay các đối tượng của hệ điều hành. Chúng sẽ được coi là resource.

Nói chung trong hầu hết các trường hợp, bạn thậm chí ko nhận ra việc mình có phải đang làm việc với resource hay ko.

Bài 3: Kết hợp PHP và HTML

Nói đến PHP, người ta nói đến lập trình web. Nói đến HTML, người ta cũng nói đến làm web. Vậy ko có lý gì HTML và PHP lại ko đi được cùng với nhau! Bài 3 sẽ đề cập tới một vấn đề rất phổ biến khi lập trình PHP: Kết hợp mã PHP với HTML.

Trước hết, chúng ta hãy dành ít phút tìm hiểu cách thức hoạt động của World Wide Web (WWW).

Hãy tưởng tượng, bạn đang muốn truy cập trang web www.example.com/welcome.html. Bạn mở trình duyệt web, gõ vào ô địa chỉ: www.example.com/welcome.html và bấm Enter. Trang web sẽ hiện ra, gần như ngay tức khắc (ở đây ko nói đến mạng dial up siêu chậm nhá)

Vậy, điều gì đã xảy ra từ lúc bạn bấm Enter cho đến lúc trang web xuất hiện? Hãy cùng tớ xem xét những đoạn băng "behind the scene" này:

1. Ngay sau khi bạn bấm Enter, trình duyệt bạn đang dùng sẽ gửi một thông điệp (message) lên mạng, cho biết bạn đang muốn yêu cầu (request) trang www.example.com/welcome.html
2. Thông điệp đó được chuyển tới máy tính tại địa chỉ www.example.com/welcome.html
3. Máy chủ trên máy tính đó sẽ nhận được thông điệp và bắt đầu tìm kiếm file HTML được yêu cầu.
4. Máy chủ gửi file HTML đó về máy tính vừa yêu cầu (chính là máy tính của bạn). Nếu ko tìm

thấy file HTML được yêu cầu, đơn giản là máy chủ sẽ trả lại một thông báo lỗi.
5. Trình duyệt của bạn, sau khi nhận về trang HTML, sẽ hiển thị nó ra màn hình.

Ở bước thứ 4, nếu file bạn yêu cầu là 1 file mang đuôi .php, thay vì gửi trả lại nội dung nguyên gốc của file, máy chủ sẽ lần lượt thực hiện thêm các bước:
1. Quét file trong chế độ HTML, gửi trả về nội dung HTML.
2. Ngay khi gặp <?php, máy chủ sẽ chuyển sang chế độ PHP, bắt đầu thực thi các lệnh PHP cho đến khi gặp ?>. Hiển nhiên nếu các lệnh PHP có output, máy chủ sẽ trả những output đó cho trình duyệt.
3. Kết thúc chế độ PHP (ra ngoài ?>), máy chủ quay lại chế độ HTML.
Quá trình cứ thế tiếp tục, cho đến khi kết thúc file .php.

Vậy là đã xong phần nói ngoài lề. Giờ ta bắt đầu vào Bài 3.

Ở Bài 1, tớ đã cùng các bạn viết chương trình đầu tiên, Hello World, bằng PHP. Giờ thử nhìn một file .php cũng mang nội dung Hello World:

HTML Code:

```
<html>
	<head>
		<title>Hello World</title>
	</head>
	<body>
		<p>Hello World!</p>
	</body>
</html>
```

Như ví dụ trên đây, các bạn có thể thấy: Đây chỉ đơn thuần là một file HTML, mang đuôi .php. Chẳng có gì đặc biệt! Và khi trình duyệt yêu cầu file này, máy chủ chỉ việc gửi trả nội dung nguyên gốc mà ko cần phải xử lý một chút lệnh nào cả.

Giờ hãy thử nâng cấp file .php đó bằng cách thêm vào nó một chút mã PHP:

PHP Code:

```
<html>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <?php
      echo "<p>Hello World!</p>";
    ?>
```

```
  </body>
</html>
```

Khi chạy script này, kết quả khi view source code cũng ko khác gì ví dụ đầu tiên. Chỉ có cách làm là khác, thay vì chỉ sử dụng HTML, ta kết hợp cả PHP và HTML trong cùng một file.

Giờ, nếu ta muốn in ra màn hình chữ Hello ở một dòng, và World ở một dòng, ta sẽ làm ntn?

Nếu các bạn có biết về HTML, thì sẽ nghĩ ngay đến thẻ <br />:

PHP Code:

```
<html>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <?php
      echo "<p>Hello<br />World!</p>";
    ?>
  </body>
</html>
```

Kết quả output thật mỹ mãn và chẳng có gì đáng nói.

Tuy vậy, nếu các bạn còn nhớ, tớ đã từng nói \n có thể dùng để xuống dòng trong PHP. Vậy, thừa thắng xông lên, bạn sẽ thay <br /> bằng \n:

PHP Code:

```
<html>
  <head>
    <title>Hello World</title>
  </head>
  <body>
    <?php
      echo "<p>Hello\nWorld!</p>";
    ?>
  </body>
</html>
```

Kết quả? Thất bại. Trên màn hình, chữ Hello và World vẫn nằm cùng một dòng. Tại sao lại như

vậy? Làm thế nào để giải quyết vấn đề này?

Trả lời: \n đúng là để xuống dòng, nhưng đó là xuống dòng trong PHP output, nó ko đảm bảo việc xuống dòng khi cái PHP output đó được trình duyệt xử lý dưới dạng mã HTML.

Để trình duyệt xử lý chính xác những vấn đề ntn, ta cho toàn bộ xâu đó vào thẻ <pre>, thẻ quyết định việc giữ nguyên định dạng của xâu:

PHP Code:

```
<html>
   <head>
      <title>Hello World</title>
   </head>
   <body>
      <?php
         echo "<p><pre>Hello\nWorld!</pre></p>";
      ?>
   </body>
</html>
```

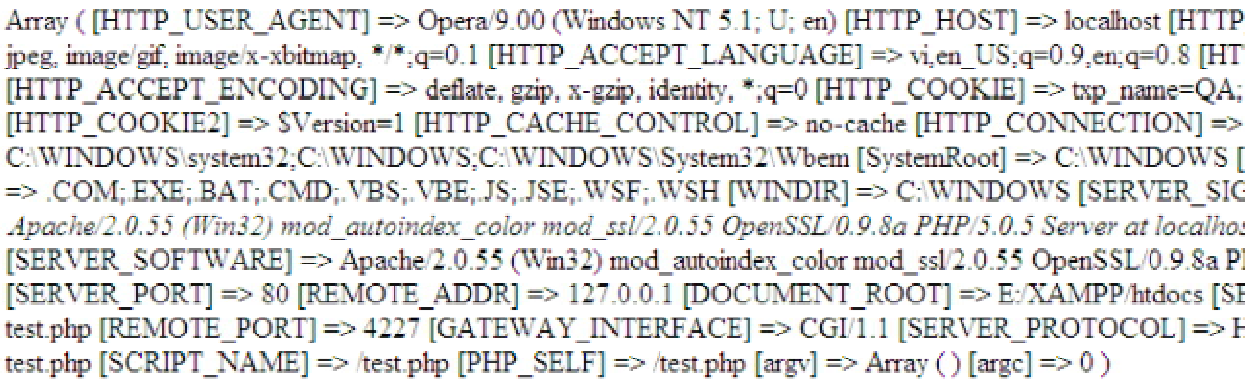
Một lần nữa, kết quả hiện ra thật mỹ mãn.

Ta xét thêm một ví dụ nữa.

PHP Code:

```
<?php
   print_r($_SERVER);
?>
```

Script trên làm trò gì vậy ta? Chưa cần biết print_r và $_SERVER là gì, bạn chỉ cần thấy đoạn nó output ra mấy dòng sau: (tớ phải post ảnh vì UDS ko cho phép đưa đoạn ý vào bài viết)



Phản ứng đầu tiên? Bạn sẽ thấy hơi chóng mặt phải ko ạ? Bạn sẽ tự hỏi: Nhiều thứ thế kia viết líu ríu vào nhau thì ai mà đọc được? Tại sao ko tách dòng ra chứ?

À, nói đến tách dòng, bạn sẽ nhớ ngay tới thẻ <pre> mà tớ nói bên trên. Bạn sẽ thêm nó vào script của mình:

PHP Code:

```php
<?php
   echo "<pre>";
   print_r($_SERVER);
   echo "</pre>";
?>
```

Kết quả trả về ko thể nói là dễ đọc, mà phải nói là rất dễ đọc , tuy (có thể) bạn chẳng hiểu cái gì sất!

Hãy tạm hài lòng với những gì mình vừa làm được và thư giãn một chút trước khi ta bước vào Bài 4.

Bật mí trước: Chúng ta đã biết tới hàm var_dump in ra kiểu và giá trị của biến, hàm print_r (bạn đoán là) in ra các giá trị của một cái $_SERVER gì đó. Vậy trong Bài 4, ta sẽ cùng tìm hiểu print_r là gì, sử dụng ra sao, và còn những hàm nào như vậy nữa.

Bài 4: Các toán tử

Để thực hiện việc tính toán các giá trị trong PHP, ta sử dụng toán tử (operator).

1. Gán (assignment)

Toán tử gán (dấu =) được sử dụng hết sức đơn giản. Ví dụ:

PHP Code:

```php
<?php
   $a = 1;
   $b = 1;
   $c = "cool";
?>
```

Sau ví dụ, biến $a và $b mang giá trị 1, $c mang giá trị "cool".

Để cho ngắn gọn, thay vì phải mất 2 dòng khai báo $a và $b, ta có thể gộp:

PHP Code:

```php
<?php
   $a = $b = 1;
   // Hoặc: $b = $a = 1;
?>
```

Kết quả vẫn đúng như mong đợi.

2. Toán tử số học (arithmetic)

Các toán tử này gồm có: + (cộng - addition), - (trừ - subtraction), * (nhân - multiplication), / (chia - division) và % (tính modul - modulus).

Ví dụ:

PHP Code:

```php
<?php
   $a = 10;
   $b = 5;
   $c = $a + $b; // $c = 15
   $d = $c - $a; // $d = 5
   $e = $a / $b; // $e = 2
   $f = $e * $b; // $f = 10
   $g = $a % $e; // $g = 0
?>
```

Ngoài ra, để sau khi tính toán, giá trị $a bằng $a nhân 2 chẳng hạn, thay vì viết $a = $a * 2; ta có thể viết ngắn gọn: $a *= 2;

Tương tự, có thể viết $a += 10; $a -= 1; $a /= 3; $a %= 1; Cấu trúc này rất giống C và C++, nên nếu bạn đã biết qua 2 ngôn ngữ này thì ko có gì phải bỡ ngỡ.

3. Toán tử so sánh (comparision)

Toán tử so sánh gồm những toán tử sau:

== Mang giá trị TRUE khi 2 vế mang cùng giá trị
=== Mang giá trị TRUE khi 2 vế mang cùng giá trị VÀ cùng kiểu
!= Mang giá trị TRUE khi 2 vế ko cùng giá trị
<> Mang giá trị TRUE khi 2 vế ko cùng giá trị
!== Mang giá trị TRUE khi 2 vế ko cùng giá trị HOẶC ko cùng kiểu
< Mang giá trị TRUE khi vế trái mang giá trị nhỏ hơn vế phải
> Mang giá trị TRUE khi vế trái mang giá trị lớn hơn vế phải
<= Mang giá trị TRUE khi vế trái mang giá trị nhỏ hơn hoặc bằng vế phải
>= Mang giá trị TRUE khi vế trái mang giá trị lớn hơn hoặc bằng vế phải

Ta sẽ viết là $a == $b, $a !== $b, $a > $b...

Ví dụ:

PHP Code:

```
<?php
   "123" == 123     // Đúng
   "123" === 123    // Sai
   "123" === "123"  // Đúng
?>
```

Còn một loại toán tử so sánh nữa, được viết dưới dạng:

exp1 ? exp2 : exp3

Ví dụ:

PHP Code:

```
<?php
   $a = ($b > 1) ? 2 : 1;
?>
```

Có thể giải thích như sau: Nếu $b > 1 thì $a mang giá trị 2, còn ko $a mang giá trị 1.

4. Toán tử logic (logical)

Gồm có:

&& Mang giá trị TRUE nếu cả 2 vế đều là TRUE
|| Mang giá trị TRUE nếu một trong 2 vế là TRUE
! Mang giá trị TRUE nếu vế có giá trị FALSE
xor Mang giá trị TRUE nếu có đúng 1 trong 2 vế là TRUE

Ta viết: $a && $b, $a xor $b...

Có thể dùng "and" thay cho && và "or" thay cho || cũng ko sao.

5. Toán tử bit (bitwise)

Toán tử để xử lý bit bao gồm:

& Phép And
| Phép Or
^ Phép Xor
~ Phép Not
<< Phép Shift Left
>> Phép Shift Right

Có thể viết $a >> 2, $b | $c,...

6. Toán tử dùng trong xâu

Để nối 2 xâu, ta dùng toán tử nối xâu (concatenation), biểu diễn bằng dấu chấm (.)

Ví dụ $a . $b, "Xâu" . "Một xâu khác"

Hiển nhiên có thể viết $a .= "Một xâu nào đó"

6. Toán tử dùng trong mảng

+ Gộp 2 mảng (union)
== So sánh bằng, mang giá trị TRUE nếu các phần tử của 2 mảng mang cùng khóa & giá trị (có thể thứ tự khác nhau)
!= hoặc <> Mang giá trị TRUE nếu các phần tử của 2 mảng ko cùng khóa & giá trị
=== So sánh bằng, mang giá trị TRUE nếu các phần tử của 2 mảng mang cùng khóa & giá trị & thứ tự giống nhau
!== Mang giá trị TRUE nếu 2 mảng ko giống hệt nhau (non-identical)

7. Các toán tử khác

Các toán tử khác có thể kể đến toán tử tự tăng (auto-increment) và tự giảm (auto-decrement), ký hiệu tương ứng là ++ và --.

Ví dụ $a++, $b--, --$c, ++$d

Một toán tử khác là toán tử @, cho phép PHP bỏ qua lỗi của một lần gọi hàm.

Ví dụ:

$test = @file('Bạn ko có quyền sờ đến file này');

Tận dụng mã nguồn có sẵn bằng cách include file

Sau khi đọc xong bài Sử dụng hàm trong PHP, bạn sẽ có trong tay rất nhiều hàm cần thiết khi code. Số lượng hàm này chắc chắn sẽ tăng dần theo thời gian.

Ví dụ bạn đang viết rất nhiều hàm về tính toán với hình tròn:

PHP Code:

```
function tinh_dien_tich_hinh_tron($ban_kinh) {
    // code
}

function tinh_chu_vi_hinh_tron($ban_kinh) {
    // code
}
```

```
// rất nhiều hàm khác
```

Thay vì copy và paste các hàm trên vào tất cả các file dính dáng đến việc tính toán với hình tròn, bạn hãy cho những hàm đó vào 1 file riêng, đặt tên là hinh_tron.php chẳng hạn. Mỗi lần muốn sử dụng các hàm trong file hinh_tron.php đó, bạn chỉ việc include nó vào file hiện tại bằng một trong các cách sau:

1. Dùng include

PHP Code:

```
include(tên_file);
```

2. Dùng require

PHP Code:

```
require(tên_file);
```

File được include có thể mang định dạng bất kỳ, php, inc, lib... tùy bạn chọn.

Câu hỏi 1: include với require làm gì?

Khi bạn include/require 1 file nào đó, ko cần biết file đó mang định dạng gì, PHP sẽ quét nội dung file đó và bắt đầu xử lý 2 trường hợp:
- Với những đoạn nằm trong thẻ php, PHP sẽ thực thi như với file PHP thông thường
- Với những đoạn nằm ngoài thẻ php, PHP sẽ output ra màn hình

Câu hỏi 2: include khác require ở chỗ nào?

Một file được include nếu (chẳng may) ko tồn tại sẽ khiến PHP báo lỗi, tuy nhiên phần còn lại của script vẫn sẽ được thực thi.
Ngược lại, một file được require nếu (xui xẻo) ko tồn tại sẽ khiến PHP đứng luôn, ko chạy tiếp script. Nói cách khác, file được require là file tối quan trọng, ảnh hưởng tới việc thực thi 1 script.

Ngoài việc sử dụng include và require, bạn còn có thể dùng include_once và require_once. Về cơ bản, include và include_once giống nhau. require và require_once cũng thế.

Điểm khác biệt là khi sử dụng require_once hay include_once, file đó chỉ được include đúng 1 lần duy nhất.

Hãy xét 1 ví dụ: Bạn có 3 script A, B, C. Trong đó A include B, B include C và C include lại A.

Nếu bạn chỉ dùng include() trong cả 3 trường hợp, chắc chắn PHP sẽ báo lỗi. Để tránh trường hợp này, bạn chỉ việc sửa 3 cái include() thành include_once() => Vấn đề được giải quyết!

Cấu trúc điều khiển trong PHP

Các câu lệnh điều kiện: các câu lệnh này cho phép chúng ta phân biệt các khối mã lệnh mà sẽ được thực thi chỉ khi gặp phải các điệu kiện nào đó. PHP cung cấp hai cấu trúc lệnh điều kiện. Đầu tiên là if...elseif...else, cho phép chúng ta có thể kiểm tra một số lượng các biểu thức và thực thi các câu lệnh theo giá trị của chúng. Nếu chúng ta mong muốn kiểm tra một biểu thức đơn lẻ với một số lượng các giá trị, PHP cũng cung cấp một cấu trúc switch...case mà có thể làm đơn giản hoá đi phép toán này.

1) Câu lệnh If: Câu lệnh If là một trong những đặc tính quan trọng nhất của mỗi ngôn ngữ lập trình. Nó cho phép thực thi chọn lựa các dòng mã lệnh chỉ khi thoả mãn các điều kiện cụ thể.
Chẳng hạn:

PHP Code:

```
if ($country=="ca")echo ("Canada");//Canada được in ra khi biến $country là ca
```

Nếu nhiều hơn một câu lệnh được thực hiện khi thoã mãn điều kiện thì sử dụng dấu {} để chỉ ra những dòng lệnh nào là nằm trong khối if:

//Canada sẽ chỉ được in nếu biến $country là ca

PHP Code:

```
if ($country=="ca"){

        echo("Canada");

        echo("Ottawa");

    }
```

Điều kiện được kiểm tra trong dấu **{}** phải trả về giá trị Boolean, hoặc là true hoặc là false. Cũng như bất kỳ điều kiện nào mà không được thoã mãn, zero hay là chuỗi rỗng (""), các giá trị không được định nghĩa thì tất cả đều trả về giá trị là false. Các điều kiện có thể được nối với nhau bằng các toán tử logic and (&&), or(||) và xor.Ví dụ như sau:

PHP Code:

```
if (((4 < 5) && (3 > 2)) xor (5 == 5)) echo ("This will not print");
```

Các điều kiện phân nhánh: Nếu điều kiện được kiểm tra mà trả về false, thì PHP cho phép ta chỉ ra một khối lệnh khác cần được thực hiện bằng cách dùng từ khoá else. Mỗi thứ trong khối mã lệnh thực thi điều kiện này được xem như là một phân nhánh và mỗi nhánh phải được định vị trong các dấu ngoặc nếu chức nhiều hơn một dòng lệnh.Ví dụ:

PHP Code:

```
if ($h < 0) {

        echo ("Negative");

    } else {

        echo ("Positive");

    }
```

PHP cũng cung cấp từ khoá elseif để kiểm tra các điều kiện lựa chọn nếu điều kiện trong câu lệnh if là không đúng. Một số câu lệnh elseif có thể được sử dụng với câu lệnh if. Nhánh else cuối cùng cho phép chúng ta định vị đoạn mã mà nên được thực hiện nếu cả điều kiện if và elseif đều không đúng.

PHP Code:

```
if ($h < 0) {

        echo ("Negative");

    } elseif ($h == 0) {

        echo ("Zero");

    } else {
```

```
        echo ("Positive");

    }
```

Ta cũng có thể kiểm tra những điều kiện hoàn toàn khác nhau khi sử dụng elseif:

PHP Code:

```
if ($country == "ca") {

        // do something ...

            } elseif ($position == "h") {

        // do something else ...

    }
```

Chú ý: cả hai điều kiện trên đều là true, nhưng chỉ có nhánh lệnh thứ nhất là được thực hiện.

Cũng có thể sử dụng các câu lệnh if lồng nhau trong câu lệnh if khác.Ví dụ:

PHP Code:

```
if ($country == "ca") {

        if ($position == "h") {

            echo ("Human resources positions in Canada.");

        } elseif ($position == "a") {

            echo ("Accounting positions in Canada.");

        }

            }
```

Các câu lệnh trên cũng tương tự như sau:

PHP Code:

```
if ($country == "ca" && $position == "h") {

        echo ("Human resources positions in Canada.");

    } elseif ($country == "ca" && $position == "a") {

        echo ("Accounting positions in Canada.");

    }
```

PHP cũng cung cấp một cú pháp lựa chọn cho câu lệnh if,đó là if....endif.Ví dụ:

PHP Code:

```
if ($country == "ca"):

        echo ("Canada");

    elseif ($country == "cr"):

        echo ("Costa Rica");

    else:

        echo ("the United States");

    endif;
```

2) Câu lệnh switch: được sử dụng khi một biến riêng rẽ đang được kiểm tra so với các giá trị khác.
Ví dụ:

PHP Code:

```
switch ($country) {
```

```
        case "ca":
            echo ("Canada");
            break;
        case "uk":
            echo ("the United Kingdom");
            break;
        default:
            echo ("the United States");
    }
```

Khi câu lệnh switch thực hiện kiểm tra giá trị của biến $country và so sánh nó với mỗi một trong các giá trị trong các mệnh đề case. Khi một giá trị thích hợp được tìm thấy, các câu lệnh kết hợp với case được thực hiện cho đến khi gặp câu lệnh break. Còn nếu không tìm ra được giá trị thích hợp nào thì câu lệnh default sẽ được thực hiện. Chú ý rằng lệnh switch trong PHP thì linh hoạt hơn nhiều so với hầu hết các ngôn ngữ khác. Không giống như C, Java và ngay cả JavaScript, các giá trị case cũng có thể là một trong các loại vô hướng, bao gồm tất cả các số ,các chuỗi và ngay cả các biến.Ví dụ:

PHP Code:

```
$val = 6;$a = 5;$b = 6;
    switch ($val) {
        case $a:
            echo ("five");
            break;
        case $b:
```

```
            echo ("six");

            break;

        default:

            echo ("$val");

    }
```

Các mảng và các đối tượng chỉ là những loại dữ liệu là không phải là những nhãn đúng của case trong PHP.

Vòng lặp: Các vòng lặp chính là các phương tiện của việc thực thi một khối mã lệnh trong một số lần cho trước hay là cho đến khi gặp phải một điều kiện nhất định. PHP có hai loại vòng lặp: vòng lặp while kiểm tra điều kiện trước hay là sau mỗi bước tính lặp đi lặp lại và thực hiện lặp lại chỉ khi điều kiện là đúng. Một kiểu lặp khác là for, trong trường hợp này, số lượng bước tính lặp đi lặp lại được qui định trước khi lặp lần đầu và không thể bị thay đổi.

1. Vòng lặp while: là câu lệnh lặp đơn giản nhất. Cú pháp tương tự như câu lệnh if:

PHP Code:

```
while (condition) {

        //các câu lệnh

    }
```

Một vòng lặp while sẽ kiểm tra một biểu thức Boolean. Nếu biểu thức là false thì đoạn mã bên trong dấu ngoặc móc sẽ được bỏ qua. Ngược lại, nếu có giá trị true thì đoạn mã bên trong dấu ngoặc móc sẽ được thực hiện. Khi gặp dấu } thí điều kiện kiểm tra sẽ được thực hiện lại và nếu có giá trị là true thì đoạn mã trong vòng lặp sẽ được thực hiện lại. Điều này sẽ tiếp tục cho đến khi gặp phải điều kiện . Chú ý rằng điều kiện chỉ được kiểm tra mỗi khi bắt đầu vòng lặp, bởi vậy ngay khi sự chính xác của điều kiện thay đổi trong suốt đoạn giữa của khối lệnh ,thì mã lệnh sẽ vẫn được thực thi cho đến hết. Để thoát khỏi vào thời điểm sớm hơn,ta có thể sử dụng lệnh break. Ví dụ:

PHP Code:

```
$i = 11;

    while (--$i) {
```

```
        if (my_function($i) == "error") {
            break; // dừng vòng lặp!
        }
    ++$num_bikes;
    }
```

Trong ví dụ này, nếu ta hình dung rằng hàm my_function không trả về bất kì lỗi nào thì vòng lặp sẽ lặp đi lặp lại 10 lần và dừng lại khi biến $i =0. Còn nếu my_function trả về lỗi, thì câu lệnh break sẽ được thực hiện và vòng lặp sẽ dừng lại. Có nhiều trường hợp mà chúng ta mong muốn kết thúc chỉ khi sự lặp lại hiện thời của vòng lặp không phải là toàn bộ vòng lặp của chính nó. Để đạt được điều này, ta sử dụng lệnh continue.Ví dụ:

PHP Code:

```
$i = 11;
    while (--$i) {
        if (my_function($i) == "error") {
            continue;
        }
    ++$num_bikes;
    }
```

Đoạn mã này cũng lặp đi lặp lại 10 lần nếu không có lỗi nào được trả về bởi hàm my_function. Tuy nhiên tại lúc này, nếu có lỗi xảy ra, việc thực hiện sẽ lướt qua sự lặp lại kế tiếp của vòng lặp, mà không tăng biến đếm $num_bikes.Giả sử biến $i vẫn lớn hơn 0, vòng lặp sẽ tiếp tục như bình thường.

2. Vòng lặp do...while: vòng lặp này cũng giống như while, ngoại trừ điều kiện được kiểm tra tại cuối mỗi vòng lặp, thay vì là ở đầu. Điều này có nghĩa là vòng lặp sẽ luôn luôn thực hiện ít nhất một lần.
Ví dụ:

PHP Code:

```
echo ("<SELECT name='num_parts'>\n");

        $i = 0;

        do {

            echo ("\t<OPTION value=$i>$i</OPTION>\n");

        } while (++$i < $total_parts);

        echo ("</SELECT>\n");
```

Với đoạn mã trên, giá trị zero luôn luôn xuất hiện như là một tùy chọn trong thành phần <SELECT>, ngay cả nếu biến $total_parts=0.

Các câu lệnh while và do...while thường được dùng với các toán tử tăng hay giảm để điều khiển khi nào thì bắt đầu và dừng như ví dụ trên. Các biến thường được dùng cho mục đích này đôi khi được định nghĩa như là các biến điều khiển vòng lặp.Thông thường sử dụng các câu lệnh while trong việc đọc các records từ một truy vấn cơ sở dữ liệu, từ các dòng trong một file hay là từ các nhân tố trong một mảng.

3. Vòng lặp for: Cấu trúc của vòng lặp for là khá phức tạp hơn mặc dầu các vòng lặp for thường tiện lợi hơn các vòng lặp while:

PHP Code:

```
for ($i = 1; $i < 11; ++$i) {

            echo ("$i <BR> \n"); //In từ 1 đến 10

        }
```

Câu lệnh for chứa ba biểu thức bên trong dấu ngoặc đơn của nó, phân biệt với nhau bởi dấu chấm phẩy.Biểu thức thứ nhất là một câu lệnh gán để khởi tạo biến điều khiển vòng lặp. Câu lệnh này được thực thi chỉ một lần trước sự lặp lại lần đầu của vòng lặp.Biểu thức thứ hai là

biểu thức Boolean mà được thực thi tại đầu mỗi lần lặp. Nếu giá trị trả về là true thì vòng lặp sẽ tiếp tục thực hiện. Nếu là false thì vòng lặp kết thúc. Biểu thức thứ ba là một câu lệnh mà thực thi tại giai đoạn cuối của mỗi lần lặp của vòng lặp. Nó thường được dùng để tăng hay giảm các biến điều khiển vòng lặp .

Hàm (Functions) trong PHP

Không thể không nói đến hàm trong việc lập trình, nhờ có nó mà chương trình của chúng ta trở nên dễ dàng tổ chức hơn. Như các ngôn ngữ khác, PHP có khả năng cung cấp những hàm do người dùng tự định nghĩa. Đồng thời, PHP cũng có một số cải tiến để việc viết hàm được dễ chịu và mạnh mẽ hơn.

Định nghĩa và gọi hàm

Rất dễ để **định nghĩa một hàm trong PHP**:

PHP Code:

```
<?php

  function tên_hàm([các tham s&#7889; truyền vào ...])
  {
    [thân hàm ...]
  }

?>
```

- Từ khoá *function* báo cho PHP biết rằng đây là một hàm. Tiếp theo đó là tên hàm. Tên hàm của PHP có thể là bất cứ ký tự Unicode gì (kể cả tiếng Việt, tiếng Trung…, nhưng không được phép bắt đầu bằng số). Thật sự mạnh mẽ, nhưng bạn sẽ gặp vấn đề khi lưu file đó. Thôi thì cứ đặt tên không dấu là ổn nhất . Ví dụ:

PHP Code:

```
<?php
function this_is_m&#7897;t_hàm()
 {
   echo "Hoàn toàn hợp lệ !!!";
 }
?>
```

- Sau tên hàm là danh sách tham số truyền vào và phần thân hàm. Phần thân hàm phải bắt đầu và kết thúc bằng cặp dấu { }. Phần thân này được thực thi khi tên hàm được gọi.

- Chú ý: mỗi tên hàm chỉ được định nghĩa một lần. Với một số ngôn ngữ khác, hàm có thể được gọi đè khi danh sách tham số truyền vào là khác nhau (Java chẳng hạn), nhưng PHP thì không có việc đó.

Gọi hàm cũng khá dễ. Bạn chỉ việc gọi tên hàm cùng danh sách tham số đi kèm. Hay hơn, việc gọi hàm KHÔNG PHÂN BIỆT CHỮ HOA-CHỮ THƯỜNG. Tuy nhiên, khuyến cáo là nên gọi hàm theo đúng tên hàm đã đặt, như thế dễ quản lý hơn.

PHP Code:

```php
<?php

 generate_left_menu_bar();
 GeNeRaTe_LeFt_MEnu_BaR();  // cũng được, nhưng không nên dùng !!!
 process_user_information($current_user, "new user", 65.0);
 generate_copyright_notices();
 generate_left_menu_bar;    // Sai !! Vì không có dấu ()!!

?>
```

Chú ý ví dụ trên, khi gọi tên hàm, luôn phải có cặp dấu ( ) nếu hàm không nhận tham số nào (còn nếu nhận tham số thì tất nhiên cặp dấu đó để chứa tham số rồi, phải không).

Ngừng việc thực thi hàm
- Vào bất cứ thời điểm nào trong quá trình thực thi hàm, bạn cũng đều có thể dừng công việc của hàm bằng từ khoá return.

PHP Code:

```php
<?php

 function work_work_work()
 {
  $dow = date('l');

  if ($dow == 'Saturday' or $dow == 'Sunday')
  {
   // nghỉ việc vào cuối tuần
   return;
  }

  // work hard
  work_harder();
 }

?>
```

- Khi mà hàm work_work_work được gọi vào thứ 7 hoặc Chủ nhật, nó trả về "không gì cả", còn nếu không, nó trả về giá trị "làm việc chăm chỉ hơn đi !!" (Ví dụ chỉ mang tính minh họa).

Đưa tham số vào hàm

- Ví dụ cho một cấu trúc cơ bản:

PHP Code:

```
<?php

 function my_new_function($param1, $param2, $param3, $param4)
 {
  echo <<<DONE
    You passed in: <br/>
    \$param1:  $param1 <br/>
    \$param2:  $param2 <br/>
    \$param3:  $param3 <br/>
    \$param4:  $param4 <br/>

DONE;
 }

?>
```

- Khi đưa một số tham số vào hàm, bạn phải phân cách chúng bằng dấu phẩy (,). Bạn có thể truyền bất kỳ tham số nào vào hàm, bất kể là biến, hằng số.. hoặc thậm chí là một hàm khác:

PHP Code:

```
<?php

 // gọi hàm với nhiều loại tham số truyền vào
 my_new_function($userName, 6.22e23, pi(), $a or $b);

?>
```

Giá trị trả về của hàm

- Thông thường, người ta lập trình hàm chỉ để xử lý một công việc nhất định mang tính lặp lại, và giá trị trả về của hàm là không có (null). Nhưng không hẳn tất cả mọi trường hợp đều như vậy:

PHP Code:

```
<?php

 function is_even_number($number)
 {
  if (($number % 2) == 0)
    return TRUE;
  else
    return FALSE;
 }

?>
```

-> Hàm trên có giá trị trả về là một giá trị boolean True hoặc False.

Lời kết:
Hàm là một công cụ rất mạnh trong PHP. Việc sử dụng hàm không chỉ để tối ưu các đoạn code, nó còn làm cho chương trình dễ đọc hơn và thích hợp để làm trong một nhóm với nhau.

Chúc bạn thành công,
iSheep

Kiểu dữ liệu PHP (tiếp theo - bon tren)

PHP hỗ trợ tám kiểu dữ liệu nguyên thuỷ.
Bốn kiểu thông thường là: **boolean, integer, floating-point number(float), string**.
Hai kiểu phức tạp là: **mảng(array) và đối tượng (object)**. Và cuối cùng là hai kiểu đặc biệt : **resource và NULL**. Loại dữ liệu của biến thông thường không được gán bởi người lập trình mà được quyết định tại thời gian chạy của PHP, phụ thuộc vào ngữ cảnh mà biến được dùng.

1. Boolean: đây là kiểu đơn giản nhất. Một kiểu boolean biểu thị một giá trị thật. Nó có thể là TRUE hay FALSE.

Cú pháp: để chỉ định một giá trị boolean, có thể sử dụng từ khoá TRUE hay là FALSE. Cả hai đều không phân biệt chữ hoa hay chữ thường.

Ví dụ:

Code:

```
$foo=True; // gán giá trị TRUE cho biến $foo.
```

Để có thể chuyển một giá trị sang kiểu boolean, chúng ta có thể dùng (bool) hay (boolean). Tuy nhiên trong hầu hết các trường hợp bạn không cần phải sử dụng việc ép kiểu này, bởi giá trị sẽ được tự động chuyển nếu nó là một toán tử, hàm hay là cấu trúc điều khiển đòi hỏi một tham số kiểu boolean.

Chú ý: -1 được xem là TRUE, giống như các giá trị khác 0 khác (bất kể là số dương hay âm).

2. **Integer: là một tập hợp bao gồm các số {...,-2,-1,0,1,2,...}.**
Cú pháp: Integer có thể được chỉ định trong cơ số 10, cơ số thập lục phân hay cơ số bát phân, tuỳ chọn đi trước bởi dấu - hay +. Nếu bạn sử dụng với cơ số bát phân, bạn phải theo thứ tự với 0 đứng trước, còn đối với số thập lục phân thì 0x.

Ví dụ như sau:

Code:

```
$a = 1234; # số thập phân

$a = -123; # số âm

$a = 0123; # số bát phân

$a = 0x1A; # số thập lục phân
```

Kích thước của kiểu dữ liệu này là 32bit, và PHP không hỗ trợ kiểu unsigned integer. Nếu bạn chỉ định một số vượt qua biên của kiểu dữ liệu integer, nó sẽ được xem như kiểu float. Tương tự như vậy, khi bạn thực hiện một phép toán mà kết quả trả về là một số vượt qua biên của kiểu integer, thì kiểu float sẽ được trả về. Tuy nhiên, có một lỗi trong PHP mà không phải bao giờ điều này cũng đúng, nó liên quan đến các số âm. Chẳng hạn, khi bạn thực hiện -50000* $million, kết quả sẽ là 429496728. Tuy nhiên, khi cả hai toán tử đều là số dương thì không có vấn đề gì xảy ra.

Để chuyển một giá trị sang kiểu integer, ta có thể dùng toán tử ép kiểu (int) hay (integer). Tuy nhiên, trong hầu hết các trường hợp bạn không cần phải dùng toán tử ép kiểu đó, bởi giá trị sẽ được tự động chuyển sang nếu toán tử, hàm hay cấu trúc điều khiển đòi hỏi một đối số integer.

3. Kiểu số thực (floats,doubles,hay real numbers) : có thể được chỉ định bằng cách sử dụng một trong các cú pháp sau:

Code:

```
$a = 1.234; $a = 1.2e3; $a = 7E-10;
```

Kích cỡ của kiểu float tùy thuộc vào platform, giá trị lớn nhất là xấp xỉ 1.8e308

4. String: là những chuỗi các kí tự.Trong PHP,một kí tự cũng tương tự như một byte,do đó có chính xác 256 kí tự khác nhau.

Cú pháp: có thể khai báo bằng ba cách khác nhau như sau:

· Dấu nháy đơn: cách dễ dàng nhất để chỉ định một chuỗi đơn giản là đóng nó trong một dấu nháy đơn.Ví dụ:echo 'le bao vy';

· Dấu nháy kép: nếu chuỗi được đóng trong dấu nháy kép("),PHP hiểu sẽ có thêm các chuỗi cho các kí tự đặc biệt ..... Ví dụ: \n;\t;\\;\$;…

· Heredoc: các khác để phân định chuỗi là sử dụng cú pháp ("<<<"). Chỉ nên cung cấp một định danh sau <<<, sau đó là chuỗi và tiếp là cùng tên định danh để đóng dấu nháy. Định danh dùng để đóng phải bắt đầu bằng cột đầu tiên của dòng. Định danh được dùng phải có tên giống như trong các quy luật đặt tên biến trong PHP.

5. Mảng : là một danh sách các phần tử có cùng kiểu dữ liệu.Mảng có thể là mảng một chiều hay nhiều chiều.

· Mảng một chiều có chỉ mục: là mảng được quản lý bằng cách sử dụng chỉ số dưới kiểu integer để biểu thị vị trí của giá trị yêu cầu.Cú pháp: $name[index1];

Ví dụ: một mảng một chiều có thể được tạo ra như sau:

Code:

```
$meat[0]="chicken";

$meat[1]="steak";

$meat[2]="turkey";
```

Nếu bạn thực thi dòng lệnh sau: print $meat[1]; thì trên trình duyệt sẽ hiển thị dòng sau: steak.

Bạn cũng có thể sử dụng hàm **array( )** của PHP để tạo ra một mảng.Ví dụ:

Code:

```
$meat=array("chicken","steak","turkey");
```

· Mảng một chiều kết hợp: rất thuận lợi khi dùng để ánh xạ một mảng sử dụng các từ hơn là sử dụng các integer, nó giúp ta giảm bớt thời gian và các mã yêu cầu để hiển thị một giá trị cụ thể.

Ví dụ: bạn muốn ghi lại tất cả các thức ăn và các cặp rượu ngon.

Code:

```
$pairings["zinfandel"] = "Broiled Veal Chops";

$pairings["merlot"] = "Baked Ham";

$pairings["sauvignon"] = "Prime Rib";
```

Một cách khác là bạn có thể sử dụng hàm **array( )** của PHP để tạo ra một mảng loại này, ví dụ như sau:

Code:

```
$pairings = array( zinfandel => "Broiled Veal Chops",

merlot => "Baked Ham",sauvignon => "Prime Rib",

sauternes => "Roasted Salmon";
```

· Mảng nhiều chiều có chỉ mục: chức năng của nó cũng giống như mảng một chiều có chỉ mục, ngoại trừ việc nó có thêm một mảng chỉ mục được dùng để chỉ định một phần tử. Cú pháp: $name[index1] [index2]..[indexN];

Một mảng hai chiều có chỉ mục được tạo ra như sau:

Code:

```
        $position = $chess_board[5][4];
```

· Mảng đa chiều kết hợp: khá hữu ích trong PHP. Giả sử bạn muốn ghi lại các cặp rượu-thức ăn, không chỉ loại rượu, mà cả nhà sản xuất. Bạn có thể thực hiện như sau:

Code:

```
$pairings["Martinelli"] ["zinfandel"] = "Broiled Veal Chops";

$pairings["Beringer"] ["merlot"] = "Baked Ham";

$pairings["Jarvis"] ["sauvignon"] = "Prime Rib";
```

6. Object: bạn có thể xem object như là một biến mà minh hoạ một kiểu mẫu template được gọi là class. Khái niệm của đối tượng và lớp được sử dụng nhiều trong ngôn ngữ lập trình hướng đối tượng OOP. Không giống như các kiểu dữ liệu khác trong PHP, object phải được khai báo.Điều quan trọng là phải nhận ra rằng object không hơn gì một minh hoạ của một lớp, và hoạt động như là một khuôn mẫu cho việc tạo các object có các đặc tính và chức năng cụ thể. Cho nên, lớp(class) phải được định nghĩa trước khi khai báo một object. Để khởi tạo một đối tượng, bạn sử dung câu lệnh new để minh hoạ đối tượng với một biến.Ví dụ:

Code:

```
<?php

class foo{

   function do_foo(){

      echo "Doing foo.";

   }

}

$bar = new foo;

$bar->do_foo();

?>
```

7. Resource: là một biến đặc biệt, chứa một tham chiếu đến một resource bên ngoài. Các resource được tạo ra và sử dụng bởi các hàm đặc biệt.
Giải phóng resources: bởi do tham chiếu đếm của hệ thống được giới thiệu trong PHP4 Zend-engine, nó sẽ tự động phát hiện khi một resource không cần thiết cho lâu dài. Khi ở trong trường hợp này, tất cả các resource mà đã được dùng cho resource này được giải phóng bởi "bộ phận thu nhặt rác". Do đó, hiếm khi thật sự cần thiết để giải phóng bộ nhớ thông thường bằng cách sử dụng hàm **free_result()**.

8. NULL: giá trị NULL đặc biệt dùng để thể hiện một biến không có giá trị. Một biến được xem là NULL nếu:

o Nó được gán giá trị hằng số NULL.

o Nó chưa được khởi tạo giá trị nào.

o Nó là hàm **unset( )**

Chú thích: unset () là một hàm dùng để hủy bỏ các biến chỉ định.

Cú pháp: chỉ có một loại giá trị của kiểu NULL. Bạn có thể khai báo như ví dụ sau:

Code:

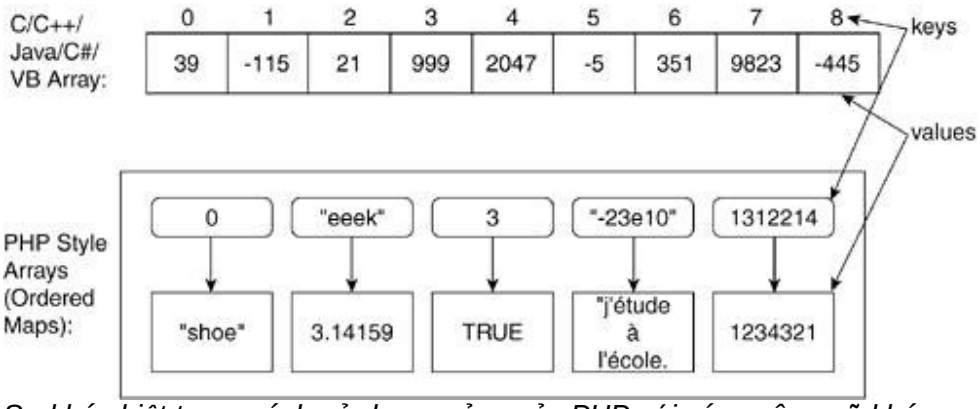
```
$var=NULL;
```

Làm việc với mảng trong PHP

Các vấn đề chính sẽ được đề cập:
- Làm thế nào để tạo 1 mảng trong PHP.
- Cách duyệt qua tất cả các phần tử trong 1 mảng.
- Sơ lược về mảng nhiều chiều

Mở đầu

Mảng là một cách hiệu quả để nhóm một lượng dữ liệu lại với nhau thành một khối duy nhất. Mảng trong PHP cũng như các ngôn ngữ khác (C/Java/VB), nó bao gồm 2 phần: khoá và giá trị (key và value). Nhưng vẫn có sự khác biệt, các khoá và giá trị trong PHP được dùng một cách tự do, không theo một ràng buộc quá mức nào cả. Bạn hoàn toàn có thể dùng một chuỗi để làm khoá, cũng như nhóm các giá trị không cùng kiểu dữ liệu với nhau lại làm thành một mảng.



Sự khác biệt trong cách sử dụng mảng của PHP với các ngôn ngữ khác

Tạo và thêm dữ liệu vào mảng

Mảng được tạo dễ dàng bằng từ khoá *array*, và có thể thêm dữ liệu ngay trong phần nội dung của mảng. Nếu bạn thích dùng một key theo ý thích của mình thì có thể dùng toán từ => để thêm dữ liệu (value) cho key đó.

Hơi khó hiểu, nhưng bạn hãy xem những ví dụ dưới đây để biết rõ hơn về cách tạo mảng cũng như cách lưu trữ giá trị trong mảng của PHP:

PHP Code:

```
<?php

 // PHP tự động gán key khi bạn tạo một mảng, bắt đầu từ key 0
```

```
 // trong ví dụ dưới, key 0 có giá trị là "Piper",
 // tương tự cho key 1, 2 và 3.
 $airplanes = array("Piper", "Cessna", "Beech", "Cirrus");

 // Chúng ta cũng có thể tạo key theo ý thích của chúng ta,
 //không bắt buộc phải là số
 $home = array("size" => 1800, "style" => "ranch",
          "yearBuilt" => 1955, "numBeds" => 3,
          "numBaths" =>2, "price" => 150000);
?>
```

PHP Code:

```
<?php

 // key của một mảng không bắt buộc bắt đầu phải là 0.
 // Chẳng hạn, key 123 được dùng làm key bắt đầu trong ví dụ này.
 $noises[123] = "hisssssss";

 // và khi khai báo như thế này, key tiếp theo của mảng sẽ là 124
 $noises[] = "gobble gobble";

?>
```

Truy xuất vào 1 phần tử của mảng

Bạn có thể truy xuất phần tử của mảng bằng cách gọi key của nó:

PHP Code:

```
<?php
 // ví dụ về cách gọi key là 1 số
 $breads = array("baguette", "naan", "roti", "pita");
 echo "I like to eat ". $breads[3] . "<br/>\n";

 $computer = array("processor" => "Muncheron 6000",
          "memory" => 2048, "HDD1" => 80000,
          "graphics" => "NTI Monster GFI q9000");

 // ví dụ về cách gọi key là 1 chuỗi
```

```
 echo "My computer has a " . $computer['processor']
     . " processor<br/>\n";

?>
```

Xoá phần tử khỏi mảng

Để xoá 1 phần tử nào đó của mảng, bạn dùng từ khóa unset cho phần tử đó:

PHP Code:

```
<?php

 $drinks = array("Coffee", "Café au Lait", "Mocha", "Espresso",
             "Americano", "Latte");
 unset($drinks[3]);    // xóa phần tử "Mocha" khỏi mảng.

?>
```

Còn muốn xoá toàn bộ phần tử của mảng, bạn cũng dùng từ khoá unset, nhưng cho toàn bộ mảng:

PHP Code:

```
<?php

 unset($drinks);   // mảng $drinks giờ đã bị xóa sạch dữ liệu

?>
```

Đếm số phần tử của mảng

Sử dụng từ khoá *count*:

PHP Code:

```
<?php

 $drinks = array("Coffee", "Café au Lait", "Mocha", "Espresso",
             "Americano", "Latte");
 $elems = count($drinks);

 // kết quả sẽ là 6.
 echo "The array \$drinks has $elems elements<br/>\n";
```

```
?>
```

II. Duyệt tất cả các phần tử của mảng

Vòng lặp **foreach**

PHP Code:

```
foreach (array as [key =>] values)
  kh&#7889;i lệnh
```

Vòng lặp này sẽ duyệt qua từng phần tử một trong mảng, nó sử dụng một biến cho trước để tạo một bảng copy phần tử mà nó đang duyệt tới và xử lý trên biến đó. Vòng lặp kết thúc khi không còn phần tử nào để duyệt.

PHP Code:

```
<?php

 $drinks = array("Coffee", "Café au Lait", "Mocha", "Espresso",
              "Americano", "Latte");
 foreach ($drinks as $drink)
 {
   echo "We serve $drink<br/>\n";
 }

?>
```

Vòng lặp thông thường (for)

Vòng lặp for hoàn toàn có thể được dùng để duyệt qua tất cả các key của mảng:

PHP Code:

```
<?php

 $drinks = array("Coffee", "Café au Lait", "Mocha", "Espresso",
              "Americano", "Latte");

 for ($x = 0; $x < count($drinks); $x++)
 {
```

```
  echo "We serve '$drinks[$x]'<br/>\n";
 }

?>
```

III. Mảng nhiều chiều (Multi-Dimensional Arrays)

Rất nhiều trường hợp bạn muốn lưu trữ nhiều mảng trong 1 mảng có sẵn. Khi đó, chúng ta có mảng một nhiều chiều. Và rất may mắn là PHP hỗ trợ rất mạnh mẽ và dễ dàng trong việc tạo mảng nhiều chiều.

Thật vậy, đây là cách mà mảng nhiều chiều được tạo trong PHP:

PHP Code:

```
<?php

 $bikes = array();
 $bikes["Tourmeister"] = array("name" => "Grande Tour Meister",
                        "engine_cc" => 1100,
                        "price" =>12999);
 $bikes["Slasher1000"] = array("name" => "Slasher XYZ 1000",
                        "engine_cc" => 998,
                        "price" => 11450);
 $bikes["OffRoadster"] = array("name" => "Off-Roadster",
                        "engine_cc" => 550,
                        "price" => "4295");

?>
```

Còn đây là cách truy xuất vào các phần tử của mảng nhiều chiều:

PHP Code:

```
<?php

 $names = array_keys($bikes);

 foreach ($names as $name)
 {
  print $bikes[$name] . " costs: " . $bikes[$name]["price"]
       . "<br/>\n";
```

```
 }

?>
```

IV. Lời kết

Mảng là một khai báo rất dễ dàng trong PHP. Việc học về mảng thật ra không có gì khó, chỉ cần đọc qua các ví dụ, bạn cũng hoàn toàn có thể rút ra cho mình được phương thức mà PHP tạo một mảng đơn giản.

Have fun.

Tìm kiếm và thay thế trong xâu với Regular Expression

Ở bài trước, chúng ta đã xem xét qua một số hàm thường gặp khi xử lý xâu trong PHP. Để tìm kiếm trong xâu, ta có thể dùng strpos() hoặc substr(), nhưng với những hàm này ta chỉ có thể tìm kiếm một cách hết sức hạn chế. Hãy thử tưởng tượng, nếu bạn muốn kiểm tra xem 1 xâu có phải là một địa chỉ IP, hay một địa chỉ email đúng đắn hay ko, sẽ phải sử dụng rất nhiều câu lệnh if.

Trong những trường hợp như thế này, ta sẽ sử dụng Regular Expression. Regular Expression, viết tắt là RegEx, có rất nhiều định nghĩa. Đây là một trong số các định nghĩa đơn giản nhất:

Regular Expression là một cách thức thể hiện dữ liệu dưới dạng các ký tự đại diện. Nó được dùng trong các thuật toán tìm kiếm, thay thế xâu.

Đây là ví dụ về một RegEx dùng để kiểm tra xem một xâu có phải là địa chỉ IP đúng đắn hay ko:

Code:

```
([0-9]{1,3})\.([0-9]{1,3})\.([0-9]{1,3})\.([0-9]{1,3})
```

Nhìn thì có vẻ rất phức tạp chứ thực ra RegEx trên rất đơn giản và dễ hiểu một khi bạn đã nắm được cách sử dụng.

Ta sẽ quay trở lại ví dụ trên sau khi đã học được một số điều cần thiết. Giờ hãy dành thời gian vào việc tìm hiểu cú pháp, cách viết 1 RegEx:

1.RegEx CÓ phân biệt ký tự hoa - thường. (case sensitive)

Ví dụ ta có một xâu như sau:

Code:

```
Hello, UDS
```

Khi đó RegEx **Hello** sẽ phù hợp với phần đầu của xâu nói trên, còn **hello** thì ko.

2. Mọi ký tự trong RegEx đều ứng với một ký tự trong xâu cần kiểm tra, kể cả ký tự trắng (dấu cách, dấu tab, dấu xuống dòng).

Ví dụ với xâu:

Code:

```
Hello, UDS
```

Thì **Hello, UDS** sẽ phù hợp còn **Hello, UDS** ko.

3. Một số ký tự có ý nghĩa đặc biệt. Ký tự ^ chỉ sự bắt đầu một xâu, còn $ chỉ sự kết thúc.

Ví dụ: Xâu

Code:

```
UDS is UDS
```

^UDS sẽ phù hợp với đoạn UDS đầu xâu, trong khi **UDS$** sẽ phù hợp với đoạn UDS cuối xâu.

4. Cũng như trong PHP, ký tự \ được sử dụng để escape một số ký tự đặc biệt. Ví dụ \$, \^, \-

Xâu:

Code:

```
$abc$
```

\$ phù hợp với ký tự $ đầu xâu.

5. Ký tự . phù hợp với mọi ký tự

Ví dụ:

RegEx **...** phù hợp với 3 ký tự đầu trong xâu

Code:

```
UDS is a great community!!!
```

Dĩ nhiên, để 1 ký tự trong RegEx phù hợp với dấu . (thật) thì cần phải escape dấu . ấy như thế này \.

Ví dụ:

Code:

```
O.K.
```

\. sẽ phù hợp với dấu . thứ nhất sau ký tự O.

6. Một danh sách các ký tự có thể đặt trong dấu ngoặc vuông []. Khi đó bất cứ ký tự nào trong ngoặc vuông được tìm thấy, ký tự đó sẽ được coi là phù hợp. Trật tự các ký tự trong ngoặc là ko quan trọng.

Ví dụ:

Code:

```
How do you do?
```

[oyu] sẽ phù hợp với ký tự o trong từ How

[dH]. sẽ phù hợp với ký tự Ho trong từ How.

7. Một dải (range) các ký tự có thể được thể hiện bằng cú pháp [-]. Có thể có nhiều dải trong một cặp ngoặc [].

Ví dụ:

Code:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789
```

[C-K] sẽ phù hợp với ký tự C.

[a-d] sẽ phù hợp với ký tự a.

[C-Ka-d2-6] sẽ phù hợp với ký tự C.

8. Nếu một lớp các ký tự đặt trong dấu [] được mở đầu bằng ký tự ^, những ký tự đó sẽ được coi là ko phù hợp.

Ví dụ:

Code:

```
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz 0123456789
```

[^CDghi45] sẽ ko phù hợp với các ký tự C, D, g, h, i, 4, 5.

9. Các xâu khác nhau có thể được đặt trong dấu () và phân cách bằng ký tự |.

Ví dụ:

Code:

```
Monday Tuesday Friday
```

(on|ues|rida) sẽ phù hợp với đoạn on trong từ Monday, ues trong từ Tuesday,...

10. Có thể chỉ ra số lần ký tự sẽ xuất hiện. Ký tự * phù hợp với "ko hoặc nhiều hơn thế", + phù hợp với "một hoặc nhiều hơn thế", ? phù hợp với "ko hoặc một".

Ví dụ:

Code:

```
aabc abc bc
```

a*b hoặc **a+b** phù hợp với aab.

a?b phù hợp với ab.

11. Dấu ngoặc móc {} được sử dụng để chính xác hóa số lượng ký tự ta mong muốn. Trong đó:

{m} cho biết ký tự xuất hiện ĐÚNG m lần
{m,n} cho biết ký tự xuất hiện ÍT NHẤT m lần và NHIỀU NHẤT n lần.
{m,} cho biết ký tự xuất hiện ÍT NHẤT m lần.
{,n} cho biết ký tự xuất hiện NHIỀU NHẤT n lần.

Ví dụ:

Code:

```
    One ring to bring them all and in the darkness bind them
```

.{5} sẽ phù hợp với đoạn ký tự *One r*.

[els]{1,3} sẽ phù hợp với ký tự *e*.

[a-z]{3,} sẽ phù hợp với đoạn *ring*.

Cần chú ý: "*", "+", và "?" là trường hợp đặc biệt của luật thứ 11. "*" tương ứng với {0,}, "+" tương ứng với {1,} còn "?" tương ứng với {0,1}.

Vậy, trên đây tớ đã giới thiệu một số luật quan trọng và cần nhớ khi sử dụng Regular Expression để tìm kiếm và thay thế trong xâu. Bài tiếp theo sẽ nói tới một số ví dụ về Regular Expression và việc áp dụng vào PHP.

Các bạn có thể tham khảo thêm về Regular Expression ở Zvon:

RegEx Tutorial: http://www.zvon.org/other/PerlTutori...put/index.html

RegEx Reference: http://www.zvon.org/other/reReference/Output/index.html

Chính bài viết này cũng đã được dịch từ RegEx Tutorial.

Tương tác với server bằng form

Việc tương tác với server bằng form có lẽ là một trong những công việc mà các bạn rất hay gặp khi lập trình web. Hãy nghĩ đơn giản, dưới góc độ một user trong UDS, bạn đã phải tương tác với bao nhiêu là form: form register, form login, form post bài nhanh - post bài advance, form để report cho mod... Nhiều quá hóa... chóng mặt

Các form đó đều có điểm chung: Đều được viết bằng HTML. Hãy cùng nhắc lại đôi chút về cách tạo form với HTML:

Trước hết, bạn hãy dành 30 giây trong quỹ thời gian "ít ỏi" của mình để ngẫm lại: HTML có cấu trúc như thế nào?

Nói một cách ngắn gọn, một file HTML có cấu tạo như sau:

HTML Code:

```
<html>
	<head>
		<title><!-- Tiêu đề --></title>
	</head>
	<body>
		<!-- Nội dung -->
	</body>
</html>
```

Form trong HTML được đặt trong phần <body></body>. Ví dụ như sau:

HTML Code:

```
<form action="process.php" method="GET">
	Tên: <input type="text" name="name" /><br />
	Tuổi: <input type="text" name="age" /><br />
	<input type="submit" value="Submit!" />
</form>
```

Đoạn code HTML trên có tác dụng gì? Rất đơn giản, nó giúp tạo ra 1 form với 2 ô để nhập dữ liệu: Tên và tuổi. Kèm theo đó là 1 nút lệnh mang chữ Submit. Rất giản dị và ko có gì khó hiểu

ở đây cả.

À, có 1 điều này cần giải thích: Ở dòng đầu tiên của đoạn code, thẻ form có 2 thuộc tính: action và method. Hai thuộc tính này có mục đích gì?

1. action="process.php" nghĩa là sau khi bấm submit, dữ liệu sẽ được chuyển qua file process.php nằm cùng thư mục với file HTML này
2. method="GET" nghĩa là phương thức truyền dữ liệu sẽ là GET.

Vậy, bạn đã biết dữ liệu sẽ được chuyển qua file process.php, mà hiển nhiên file process.php này chưa tồn tại, nên một điều rất tự nhiên là ta sẽ tạo file process.php với nội dung sau:

PHP Code:

```php
<?php
   $name = $_GET["name"];
   $age = $_GET["age"];
   echo "Tên bạn: $name";
   echo "Tuổi của bạn: $age";
?>
```

Hai dòng cuối của ví dụ trên có lẽ chẳng có j` để bàn. Chỉ đơn giản là echo 2 biến $name và $age ra màn hình. Vấn đề nằm trong 2 dòng đầu.

Ta lấy dòng thứ 1 để mổ xẻ: $name = $_GET["name"]; Ở đây ai cũng biết giá trị của $name được gán từ $_GET["name"]. Vậy là chỉ còn duy nhất 1 điều cần thắc mắc: $_GET là gì?

$_GET là mảng để chứa các giá trị được chuyển từ form sang. Ở đây $_GET gồm 2 phần tử: $_GET["name"] và $_GET["age"].

Mọi vấn đề đến đây được giải quyết.

Giờ ta thử xét form ở Ví dụ ban đầu. Bạn hãy làm một phép thay đổi nhỏ: Đổi method="GET" bằng method="POST".

Khi đó hiển nhiên process.php sẽ ko hoạt động

Nếu suy diễn một cách educated, bạn sẽ ngay lập tức sửa sai bằng cách thay đổi $_GET bằng $_POST. Kết quả? Đúng goài

Đến đây ta có thể sung sướng rút ra kết luận: Để nhận biến từ form chuyển sang, có thể sử dụng mảng $_GET hoặc $_POST, tùy vào phương thức truyền dữ liệu của form.

Bài học kết thúc ở đây được rồi nhỉ?

Chưa đâu bạn ạ. Còn một điều chưa nói đến: Sự khác nhau giữa $_GET và $_POST là j`? Hay đúng hơn, sự khác nhau giữ method GET và POST là gì?

Nói đến GET và POST, trước hết ta sẽ nói đến HTTP Request.

Một HTTP Request được gửi tới server có định dạng như sau:

Code:

```
<request-line>
<headers>
<blank line>
[<request-body>]
```

Một HTTP Request được gửi tới trang web www.abc.com từ trình duyệt Firefox sẽ tương tự như ví dụ sau:

Code:

```
GET / HTTP/1.1
Host: www.abc.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.6)
        Gecko/20050225 Firefox/1.0.1
Connection: Keep-Alive
```

Tương tự, một Request gửi đến trang www.abc.com/def sẽ có dạng như sau:

Code:

```
GET /def/ HTTP/1.1
Host: www.abc.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.6)
        Gecko/20050225 Firefox/1.0.1
Connection: Keep-Alive
```

Như các bạn có thể thấy, 2 ví dụ trên đều dùng phương thức GET để request dữ liệu.

Giờ hãy tưởng tượng, sau khi điền vào form 1 giá trị name là admin, age là 20, bạn bấm nút submit để gửi dữ liệu đến file process.php.

Trên thanh address sẽ hiện ra url: process.php?name=admin&age=20

Nếu xem xét một cách kỹ lưỡng HTTP Request, ta sẽ thấy nó như sau:

Code:

```
GET /process.php?name=admin&age=20 HTTP/1.1
Host: www.abc.com
```

```
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.6)
        Gecko/20050225 Firefox/1.0.1
Connection: Keep-Alive
```

Tóm lại, khi dùng phương thức GET, trình duyệt sẽ gửi Request tới server với các tham số đặt trong dòng đầu tiên (request-line).

Còn với phương thức POST thì sao? Thay nằm trong request-line, những tham số này được đặt trong phần request-body.

Ví dụ ta giữ nguyên form trên, chỉ thay method="GET" bằng method="POST" và bấm Submit. Khi đó đây sẽ là HTTP Request "behind the scene":

Code:

```
POST / HTTP/1.1
Host: www.abc.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.6)
        Gecko/20050225 Firefox/1.0.1
Content-Type: application/x-www-form-urlencoded
Content-Length: 40
Connection: Keep-Alive


name=admin&age=20
```

Có thể dễ dàng thấy rằng, ngoài việc thêm một số dòng vào phần header:

Code:

```
Content-Type: application/x-www-form-urlencoded
Content-Length: 40
Connection: Keep-Alive
```

thì phương thức POST đã "cất" các tham số của mình trong phần request-body. Điều này hiển nhiên khác hẳn phương thức GET.

Bài học hôm nay chính thức được kết thúc ở đây <:-P

(và tớ cũng đi xem phim smallville tiếp đây)

Xâu - Các phép toán thường gặp

Tớ xin nhắc lại một cách ngắn gọn: Xâu là một tập hợp các ký tự. Ví dụ "abc", "sadfsks12dsfsdfjkee123" có thể coi là một xâu. Trong PHP, xâu được thể hiện bằng mã ký tự 8 bit (8-bit character code) và được coi là nằm trong bảng mã ISO-8859-1.

Giờ ta hãy xem xem xét một số phép toán thường gặp trên xâu:

1. strlen - lấy độ dài 1 xâu

Có lẽ các bạn cũng dễ dàng đoán được, strlen là string length - độ dài xâu. Cách sử dụng hàm strlen() rất đơn giản:

PHP Code:

```
<?php
echo strlen("Updatesofts");
?>
```

Kết quả in ra sẽ là 11 - số ký tự trong xâu "Updatesofts".

2. trim - cắt bỏ phần thừa trong xâu

Hàm trim() nhận tham số là 1 xâu và nó sẽ loại bỏ mọi khoảng trắng (whitespace) bắt đầu và kết thúc xâu. Cần chú ý ở đây, khoảng trằng ko chỉ bao gồm dấu cách (" ", mang mã ASCII 32) mà nó còn gồm:
- Tab ("\t", mã ASCII 9)
- Dấu xuống dòng ("\r" và "\n", mã tương ứng 10 và 13)
- Ký tự NULL ("\0", mã 0)
- Tab dọc (vertical tab - mã 11). Dấu này giờ là "của hiếm". ấu

Ví dụ:

PHP Code:

```
<?php
$str = " \t\t\t\t \nXâu này lắm thứ linh tinh thế nhở \r\n \t \t ";
echo trim($str);
?>
```

Kết quả output ra hoàn toàn có thể đoán được: "Xâu này lắm thứ linh tinh thế nhở"

3. ltrim và rtrim

Sau khi biết về hàm trim(), chắc hẳn bạn sẽ đặt câu hỏi: Thế nhỡ tớ chỉ muốn cắt bỏ mấy phần linh tinh ở đầu/cuối xâu thôi thì sao?
Trả lời: Hoàn toàn có thể. Hãy dùng ltrim() và rtrim() - bạn sẽ thik ngay mà

ltrim dùng để bỏ các phần linh tinh bắt đầu xâu.
rtrim dùng để bỏ các phần linh tinh kết thúc xâu.

4. strpos - tìm kiếm trong xâu

Hàm strpos() nhận 3 tham số:

1. 1 xâu
2. xâu cần tìm trong xâu trên
3. bắt đầu tìm từ ký tự thứ mấy trong xâu, mặc định là 0

Ví dụ strpos("Updatesofts", "Update") sẽ trả về kết quả là 0. strpos("ABCABC", "A", 2) sẽ trả về kết quả là 3.

Vậy nếu ko tồn tại xâu cần tìm thì sao? Đơn giản lắm bạn ạ, kết quả trả về sẽ là FALSE.

Một điều cần chú ý nữa: Nếu vị trí bắt đầu tìm là số âm (ví dụ -1), PHP sẽ tìm kiếm từ cuối xâu trở lại (ngược với cách tìm mặc định)

Giờ hãy xét 1 ví dụ:

PHP Code:

```
<?
$res = strpos($haystack, $needle);
if ($res == FALSE) {
echo "Ko thấy!";
} else {
echo "Thấy òi!";
}
?>
```

Bạn có $haystack là "Updatesofts", $needle là "Up", hỏi PHP sẽ echo ra cái gì? Bạn đoán là "Thấy òi" đúng ko?

Tiếc là sai rồi bạn ạ Kết quả là "Ko thấy", vì $needle được tìm thấy ở ĐẦU $haystack, nghĩa là vị trí thứ 0, mà 0 lại đồng nghĩa với FALSE

Khá là confusing phải ko hả bạn?

Để xử lý trường hợp này, ta sẽ dùng toán tử so sánh === thay vì == (bằng bằng bằng thay vì bằng bằng). PHP sẽ kiểm tra cả giá trị và kiểu của biến, do đó 0 và FALSE sẽ là 2 khái niệm hoàn toàn khác nhau và vấn đề đã được giải quyết gọn ghẽ.

5. substr - tách (extract) 1 phần trong xâu

Cú pháp của hàm substr() như sau:

substr(xâu, vị trí bắt đầu, [số ký tự - nếu cần])

Ví dụ: substr($str, 1) trả về xâu bắt đầu từ ký tự thứ 1. substr($str, 3, 2) trả về 2 ký tự của xâu bắt đầu từ ký tự thứ 3.

Cũng như strpos, tham số thứ 2 có thể là âm. Khi đó PHP sẽ xử lý ngược từ cuối.

Trên đây là 5 hàm thường gặp khi xử lý xâu trong PHP. Hiển nhiên việc liệt kê tất cả các hàm là impossible, vì vậy nếu bạn cần thêm về các hàm xử lý xâu trong PHP, hãy tìm đến PHP.net:

http://www.php.net/manual/en/ref.strings.php

Coming Up Next: Sử dụng Regular Expression để tìm kiếm và thay thế trong xâu.

TẠO FORM ĐỂ UPLOAD FILE

Form để upload file cần thoã mãn các điều kiện sau:

* method là POST
* enctype là multipart/form-data

Mã HTML của form sẽ từa tựa như sau:

```
<form method="POST" enctype="multipart/form-data" action="process_upload.php">
<input type="hidden" name="MAX_FILE_SIZE" value="30000">
<input type="file" name="file_upload" size="20">
<input type="submit" value="Upload">
</form>
```

Đoạn code trên sẽ tạo 1 form với 1 nút Browse... để ban chọn file cần upload, và 1 nút Upload để bạn submit form. Form sẽ được submit tới file process_upload.php nằm cùng thư mục với file chứa form.

Một số browser support MAX_FILE_SIZE sẽ kiểm tra dung lượng file trước khi form được submit, tuy nhiên không phải browser nào cũng vậy. Cho nên bạn đừng nên tin tưởng tuyệt đối vào server! Ở ví dụ trên, nếu browser hỗ trợ, nhưng file có dung lượng lớn hơn 30000 byte sẽ được browser thông báo lỗi khi submit form.

XỬ LÝ DỮ LIỆU ĐƯỢC SUBMIT LÊN SERVER

Bây giờ ta hãy xem xét tới phần xử lý dữ liệu được submit lên server trong file process_upload.php. PHP lưu thông tin về file được upload lên server trong biến global $_FILES. Với form ở ví dụ trên, PHP sẽ truyền cho script process_upload.php các thông tin sau:

* $_FILES['file_upload']['name']: tên file gốc trên máy client. Tuỳ vào browser, tên file có thể được truyền lên server ở dạng C:\folder\filename.ext hoặc chỉ là filename.ext. Chương trình phải tự kiểm tra và trích ra tên file nếu cần thiết.
* $_FILES['file_upload']['type']: kiểu của file, được lưu ở dạng MINE (Ví dụ: image/gif, audio/wav).
* $_FILES['file_upload']['size']: dung lượng của file tính theo byte.
* $_FILES['file_upload']['tmp_name']: sau khi upload, server sẽ lưu file vào một file tạm trên server, biến này cho ta biết đường dẫn và tên của file tạm đó. Chương trình sẽ đọc file tạm này để lấy nội dung của file được upload.
* $_FILES['file_upload']['error']: mã lỗi, chương trình nên kiểm tra biến này để bảo đảm rằng quá trình upload không xảy ra lỗi.
o UPLOAD_ERR_OK (= 0): không có lỗi, quá trình upload thành công.
o UPLOAD_ERR_INI_SIZE (= 1): dung lượng file upload vượt quá giới hạn được chỉ định trong file php.ini.
o UPLOAD_ERR_FORM_SIZE (= 2): dung lượng file upload vượt quá giới hạn được chỉnh định bởi MAX_FILE_SIZE.
o UPLOAD_ERR_PARTIAL (= 3): file chỉ được upload 1 phần (có thể là do lỗi đường truyền trong quá trình upload).
o UPLOAD_ERR_NO_FILE (= 4): không có file nào được upload (có thể là file ở client không

tồn tại).

Khi đã có toàn bộ các thông tin cần thiết, xử lý file như thế nào là quyến định của bạn. Bạn có thể đọc nội dung của file và lưu vào database, hoặc di chuyển file và lưu vào thư mục upload của bạn. Sau đây là 1 ví dụ của file process_upload.php.

Đầu tiên, kiểm tra xem tác vụ có phải là upload hay không:

```
if ( $_SERVER["REQUEST_METHOD"] != "POST" ) {
//thông báo lỗi không phải là method POST
//và thoát
exit(-1);
} //end if
```

Tiếp theo kiểm tra xem quá trình upload có lỗi gì không:

```
if ( !isset($_FILES["file_upload"]["error"] ||
$_FILES["file_upload"]["error"] != 0 ) {
//thông báo lỗi dựa vào giá trị của $_FILES["file_upload"]["error"]
//và thoát
exit(-1);
} //end if

//ta cũng có thể kiểm tra xem dung lượng file có vượt quá giới hạn
//của chương trình hay không
if ( $_FILES["file_upload"]["size"] > $MAX_FILE_SIZE ) {
//thông báo lỗi
//và thoát
exit(-1);
}
```

Tách tên file từ client:

```
$temp = preg_split('/[\/\\\\]+/', $_FILES["file_upload"]["name"]);
$filename = $temp[count($temp)-1];

//ta cũng có thể kiểm tra phần mở rộng của file nếu cần thiết
if ( !preg_match('/\.(gif|jpg)$/i', $filename ) {
//thông báo lỗi file upload không phải là dạng GIF hoặc JPG
//và thoát
exit(-1);
} //end if
```

Và cuối cùng, lưu file được upload vào nơi cần thiết:

```
$upload_dir = "/home/nbthanh/public_html/uploads/";
$upload_file = $uploaddir . $filename;
if ( move_uploaded_file($_FILES["file_upload"]["tmp_name"], $upload_file) ) {
//file đã được upload và copy sang thư mục lưu trữ thành công
} else {
//có lỗi xảy ra
} //end if
```

CÁC HÀM PHP ĐƯỢC DÙNG TRONG VÍ DỤ

* exit: dừng/thoát chương trình ngay lập tức.
* isset: kiểm tra xem biến có tồn tại hay không. Trong ví dụ của bài viết, ta dùng hàm isset để kiểm tra xem biến $_FILES["file_upload"]["error"] có tồn tại hay không.
* preg_split: tách một chuỗi thành từng phần nhỏ theo regular expression. Trong ví dụ của bài viết, ta dùng hàm này để tách tên file cùng đường dẫn ra thành từng phần nhỏ (phân cách nhau bằng ký tự \ hoặc /, ta không biết chắc được client là Windows hay Linux nên ta tách theo trường hợp tổng quát). Sau khi tách, phần tử cuối cùng sẽ là tên file.
Một cách khác để lấy tên file là dùng hàm basename. Tuy nhiên sử dụng hàm này sẽ có một số vấn đề nảy sinh, bạn tham khảo thêm ở đây:
http://www.php.net/manual/en/function.basename.php.
* count: đếm số lượng phần tử trong mảng. $a[count($a)-1] sẽ truy cập tới phần tử cuối cùng của mảng $a.
* preg_match: sử dụng regular expression để tìm xem chuỗi con có xuất hiện trong chuỗi mẹ hay không. Trong ví dụ của bài viết, ta dùng hàm này để kiểm tra xem tên của của có được kết thúc bằng .gif hoặc .jpg hay không.
* move_uploaded_file: di chuyển file được upload từ client đến 1 thư mục khác trên server.

TÀI LIỆU THAM KHẢO

* PHP Manual: http://www.php.net/manual/en/index.php
o Regular Expression Functions (Perl-Compatible): http://www.php.net/manual/en/ref.pcre.php
o Handling file uploads: http://www.php.net/manual/en/features.file-upload.php
* Từ Google: từ khoá php tutorial upload file

Source from DDTH

MỤC LỤC

MỤC LỤC 1
Chương 1: Quy trình thiết kế website 6
I. Các khái niệm cơ bản 6
1. HTML (Hypertext Markup Language) – Ngôn ngữ đánh dấu siêu văn bản 6
2. Ngôn ngữ lập trình Web 6
3. WebServer – trình chủ Web 6
4. Database server – Trình chủ CSDL 6
5. Web browser-Trình duyệt Web 6
6. URL (Uniform Resource Locator)- Tài nguyên trên Internet 6
7. HTTP (Hypertext Transfer Protocol)- Giao thức truyền siêu văn bản 7
8. Cơ chế Web 7
II. Quy trình thiết kế website 8
1. Xác định mục đích, yêu cầu của website 8
2. Xác định độc giả 8
3. Thiết kế giao diện Website 9
4. Các thành phần cơ bản của Website 11
III. Một số nguyên tắc khi phát triển website. 11
Chương 2: Giới thiệu về ngôn ngữ HTML 15
I. Khái niệm cơ bản về html 15
1. HTML là gì? 15
2. Thẻ HTML 15
3. Cần gì để tạo một trang web 15
II. Các thẻ định cấu trúc tài liệu 15
1. Thẻ html 15
2. Thẻ head 16
3. Thẻ title 16
4. Thẻ body 16
III. Các thẻ định dạng khối 17
1. Thẻ định dạng khối văn bản <p> 17
2. Các thẻ định dạng đề mục h1/h2/h3/h4/h5/h6 17
3. Thẻ xuống dòng
 18
4. Thẻ pre và thẻ 18
IV. Các thẻ định dạng danh sách 18
V. Các thẻ định dạng ký tự 19
1. Các thẻ định dạng in ký tự 19
2. Căn lề văn bản trong trang Web 20
3. Các ký tự đặc biệt 20
4. Sử dụng màu sắc trong thiết kế các trang Web 20
5. Chọn kiểu chữ cho văn bản 22
6. Khái niệm văn bản siêu liên kết 22
7. Địa chỉ tương đối 23
8. Kết nối mailto 24
9. Vẽ một đường thẳng nằm ngang 24
VI. Các thẻ chèn âm thanh, hình ảnh 25
1. Giới thiệu 25
2. Đưa âm thanh vào một tài liệu HTML 26
3. Chèn một hình ảnh, một đoạn video vào tài liệu HTML 26

VII. Các thẻ định dạng bảng biểu 27
VIII. FORM 28
2. Hộp nhập văn bản 1 dòng (Oneline Textbox) 29
3. Radio Button 30
4. Checkbox 30
5. Nút lệnh (Button) 30
6. Combo Box (Drop-down menu) 31
7. Listbox 31
8. Hộp nhập văn bản nhiều dòng (TextArea) 32
IX. Một số thẻ đặc biệt 34
1. Thẻ <meta> 34
2. Thẻ <marquee> 35
3. Thẻ <style> 36
4. Thẻ <link> 36
5. Thẻ <script> 37
Chương 3: Thiết kế CSS 38
I. Giới thiệu về CSS 38
II. Cú pháp 39
1. Định dạng thuộc tính thẻ html 39
2. Định dạng một kiểu mới 40
3. Định dạng ngay trong thẻ html 41
III. Sử dụng css trong tài liệu HTML 41
1. CSS được khai báo trong một tập tin riêng 41
2. Định dạng ngay trên tài liệu html 41
IV. Một số thuộc tính thường dùng 42
1. Định kiểu nền 42
2. Định kiểu chữ 44
3. Định kiểu font 45
4. CSS Link 49
5. Định kiểu danh sách 49
6. Định kiểu bảng 51
7. Thuộc tính Id và class của thẻ 56
8. Mô hình hộp 58
Chương 4: Giới thiệu ngôn ngữ kịch bản Javascript 65
I. Giới thiệu về Javascript 65
II. Ngôn ngữ javascript 65
1. Chèn mã lệnh javascript vào trong tài liệu HTML 65
2. Lời chú thích 66
3. Biến và cách xuất thông tin lên trình duyệt 66
4. Các phép toán 67
5. Câu lệnh rẽ nhánh If...Else 69
6. Câu lệnh lựa chọn Switch 71
7. Định nghĩa hàm 72
8. Hộp thông báo 72
9. Câu lệnh lặp For 74
10. Câu lệnh lặp While 75
11. Câu lệnh lặp For...In 76
12. Sự kiện trong Javascript 77
13. Câu lệnh Try...Catch 77

14. Câu lệnh Throw 78
15. Ký tự đặc biệt Text 79
III. Đối tượng trong javascrip 80
1. Đối tượng String 80
2. Đối tượng Date 81
3. Đối tượng Array 81
4. Đối tượng Math 82
Chương 5: Ngôn ngữ PHP 83
I. Tổng quan về PHP 83
1. Cú pháp PHP 83
2. Xuất giá trị ra trình duyệt 83
3. Lời chú thích 84
4. Biến trong PHP 84
5. Hằng 87
6. Kiểu dữ liệu 88
7. Các toán tử 90
8. Các hàm kiểm tra giá trị 92
II. Câu lệnh điều khiển 96
1. Câu lệnh rẽ nhánh If...Else 96
2. Câu lệnh lựa chọn switch 98
3. Câu lệnh lặp 98
4. Sử dụng break và continue trong cấu trúc lặp 100
5. Kiểu mảng 101
III. Xây dựng hàm trong PHP 105
1. Hàm do người dùng định nghĩa 105
2. Hàm trong thư viện hàm 107
IV. Biểu mẫu form 116
1. Đặc điểm form 116
2. Biểu mẫu sử dụng phương thức $_POST 116
3. Biểu mẫu sử dụng phương thức $_GET 118
Chương 6: Hướng đối tượng trong PHP 119
I. Khái niệm 119
II. Tạo lớp 119
III. Sử dụng lớp 120
IV. Kế thừa 122
Chương 7: Tạo web động 123
I. Sử dụng tập tin dùng chung 123
1. REQUIRE 123
2. INCLUDE 125
II. Mở tập tin và thư mục 126
1. Tập tin 126
2. Thư mục 131
III. Upload tập tin lên server. 132
1. Giới thiệu 132
2. Các bước upload file 132
IV. PHP Cookies 134
1. Khái niệm 134
2. Khai báo cookie 134
3. Sử dụng cookie 135

4. Hủy cookie 135
V. PHP Sessions 135
1. Khái niệm 135
2. Cách thức hoạt động 135
3. Khởi động Session 136
4. Đặt ký Session 136
5. Sử dụng Session 136
6. Hủy biến Session 136
VI. Gửi E-mail trong PHP 137
..... 137
Ví dụ: Lấy thông tin từ Form 137
Chương 8: CƠ SỞ DỮ LIỆU MYSQL 138
I. Tổng quan 138
1. Giới thiệu CSDL 138
2. CSDL MySQL 140
II. Bảng(Table) 144
1. Khái niệm 144
2. Thuộc tính 144
3. Thao tác với bảng 146
III. Bảng ảo 149
1. Khái niệm 149
2. Tạo bảng ảo 149
3. Cập nhật nội dung bảng ảo 150
4. Xóa bảng ảo 151
IV. Toán tử 151
1. Khái niệm 151
2. Toán tử số học 151
3. Toán tử so sánh 152
4. Toán tử logic 152
V. Phát biểu SQL 152
1. Câu lệnh SELECT 152
2. Truy vấn con 154
3. Câu lệnh thêm dữ liệu 155
4. Câu lệnh cập nhật dữ liệu 156
5. Câu lệnh xóa dữ liệu 156
6. Sử dụng mệnh đề UNION trong truy vấn 157
7. Truy vấn dữ liệu từ nhiều bảng 157
8. Sử dụng hàm trong SQL 158
9. Import và export dữ liệu 160
Chương 9: PHP&MYSQL 161
I. Kết nối CSDL 161
1. Tạo kết nối 161
2. Chọn CSDL 161
3. Truy vấn dữ liệu 162
4. Thông báo lỗi 163
5. Đóng kết nối 164
II. Làm việc với CSDL MySQL 164
1. Đếm số lượng mẫu tin 164
2. Hiển thị dữ liệu 165

3. Lưu trữ dữ liệu mới vào CSDL 170
4. Cập nhật dữ liệu 172
5. Xóa dữ liệu 173
III. PHP kết hợp với các CSDL SQL Server 173
IV. Xây dựng các lớp xử lý 174
1. Một số phương thức trong lớp xử lý bảng 174
2. Xây dựng lớp xử lý nghiệp vụ 176
Mục lục 178

Chương 1: Quy trình thiết kế website

I. Các khái niệm cơ bản

1. HTML (Hypertext Markup Language) – Ngôn ngữ đánh dấu siêu văn bản

HTML là ngôn ngữ đánh dấu được sử dụng để tạo nên các trang Web, nó chứa các trang văn bản và những thẻ (tag) định dạng cho trình duyệt Web (web brower) biết làm thế nào để thể hiện các thông tin trên World Wide Web(WWW). HTML giờ đây trở thành một chuẩn Internet do tổ chức World Wide Web Consortium (W3C) duy trì. Phiên bản mới nhất của HTML là 4.01. Tuy nhiên, hiên hay HTML không còn được phát triển tiếp, nó được thay thế bằng XHTML.

2. Ngôn ngữ lập trình Web

Ngôn ngữ lập trình Web là ngôn ngữ lập trình (khác với ngôn ngữ HTML- ngôn ngữ đánh dấu siêu văn bản) được sử dụng để hỗ trợ và tăng cường các khả năng của ứng dụng Web, giúp cho việc điều khiển các phần tử của trang Web dễ dàng hơn.

Một số ngôn ngữ lập trình Web thường được dùng là: ASP, ASP.Net, PHP, JSP…

3. WebServer – trình chủ Web

WebServer là máy tính mà trên đó cài đặt các phần mềm phục vụ Web, và khi có phần mềm đó cũng được xem như một WebServer.

Tất cả các WebServer đều có thể biên dịch và chạy các file *.html và *.htm, tuy nhiên các WebServer lại phục vụ một số kiểu file riêng biệt, ví dụ như IIS của Microsoft dành riêng cho các file *.asp, *.aspx; Apache dành cho các file *.PHP; Sun Java System web server của SUN dành riêng cho các file *.jsp.

4. Database server – Trình chủ CSDL

Database server là máy tính mà trên đó có cài đặt một hệ quản trị CSDL (HQTCSDL) nào đó, ví dụ như SQL Server, MySQL, Oracle…

5. Web browser-Trình duyệt Web

Trình duyệt Web là một ứng dụng tương ứng với máy tính của người dùng, cho phép người dùng cập nhật và xem thông tin trên các trang Web. Các trình duyệt Web thông dụng hiện nay là: Internet Explorer, Netscape, FireFox, Opera, Safari…

6. URL (Uniform Resource Locator)- Tài nguyên trên Internet

URL là tài nguyên trên Internet. Sức mạnh của Web là khả năng tạo ra các liên kết siêu văn bản đến các thông tin có liên quan. Những thông tin này có thể là những trang web khác, hình ảnh, âm thanh…

Những liên kết này thường được biểu diễn bằng những chữ màu xanh có gạch dưới.

Các URL có thể truy xuất thông qua một trình duyệt (browser).

Ví dụ 1.1: Một URL có dạng http://www.hutc.edu.vn/index.html

Trong đó:

+ http: là giao thức

+ http://www.hutc.edu.vn/ là địa chỉ của máy chứa tài nguyên.

+ index.html là tên đường dẫn trên máy chứa tài nguyên.

Nhờ địa chỉ URL mà chúng ta có thể truy cập tới các trang web khác nhau.

7. HTTP (Hypertext Transfer Protocol)- Giao thức truyền siêu văn bản

HTTP là một trong các giao thức chuẩn về mạng Internet, được dùng để trao đổi thông tin giữa WebServer và người dùng (WebClient) thông qua mạng máy tính.

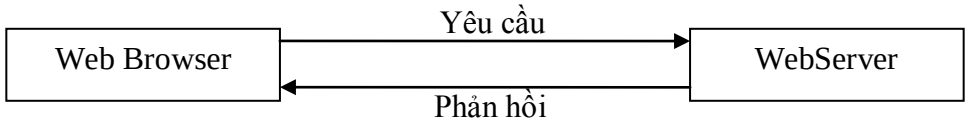
HTTP được sử dụng thông qua URL, với cấu trúc chuỗi có định dạng như sau:

http://<host>[:<port>][<path>[?query]]

8. Cơ chế Web

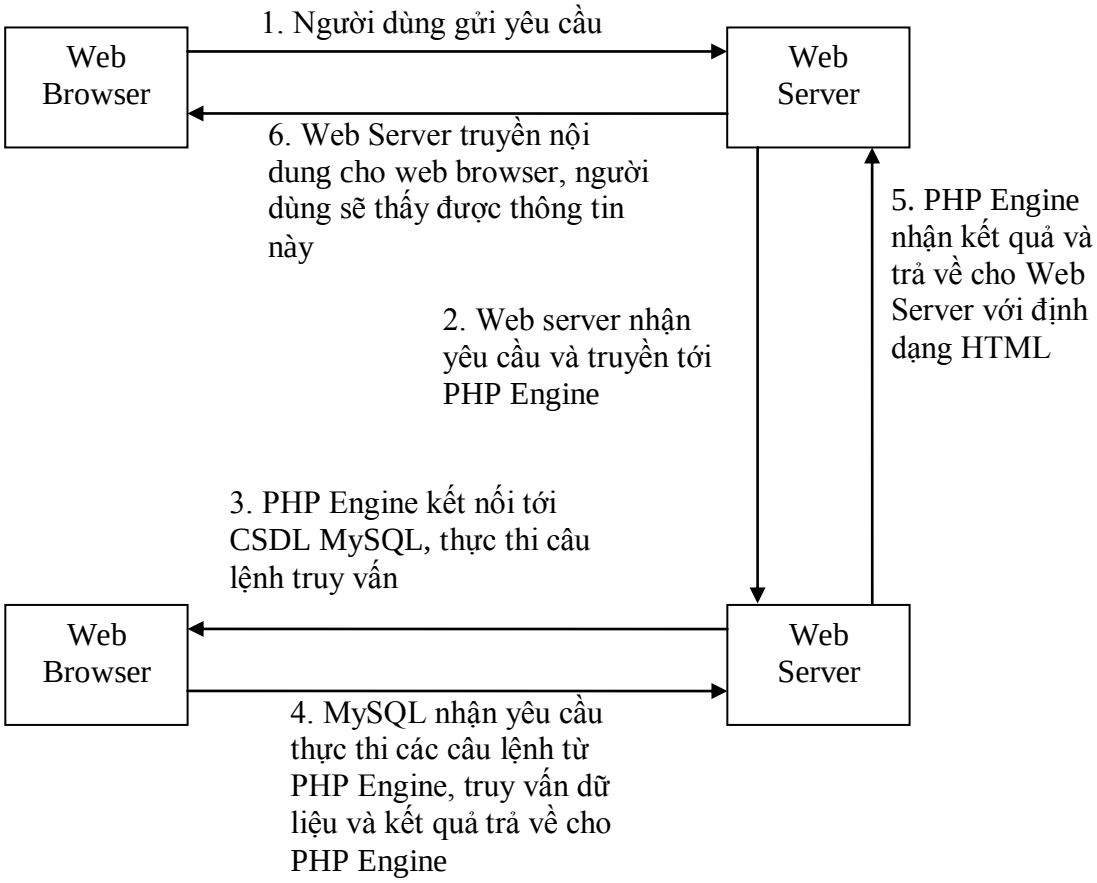
Cơ chế Web là cơ chế tương tác giữa người dùng – thông qua trình duyệt Web và WebServer.

+ Cơ chế tương tác từ người dùng với WebServer



Người dùng sẽ thông qua Web Browser để gửi yêu cầu tới WebServer và ngược lại Web Browser sẽ nhận phản hồi thông qua Web Browser đến người dùng.

+ Quy trình làm việc của PHP và MySQL thông qua Web Server



II. Quy trình thiết kế website

1. Xác định mục đích, yêu cầu của website

Để xây dựng một website có chất lượng và hiệu quả, trước tiên chúng ta cần xác định: Website dùng để làm gì? Độc giả là ai? Độc giả quan tâm đến cái gì? Cần xác định loại đọc giả là ai? Với mục tiêu "Đưa cho độc giả cái mà họ muốn xem, không phải là cái mà ta muốn họ xem".

Xây dựng website là một quá trình lâu dài, phải lên kế hoạch rõ ràng. Phải bám sát với mục đích và yêu cầu đã đăth ra.

2. Xác định độc giả

Sự thành công của một website phụ thuộc vào số lượng người truy cập(độc giả). Độc giả gồm nhiều đối tượng khác nhau như: Đọc giả chuyên nghiệp, độc giả không am hiểu nhiều về web hay mạng…

Tùy thuộc vào độc giả chính của website, chúng ta lựa chọn phong cách của website. Phong cách này thể hiện qua màu sắc, phông chữ, hình ảnh của trang web hay văn phong của website.

3. Thiết kế giao diện Website

a. Xác định kiểu chữ, màu sắc

Phụ thuộc vào các đặc điểm: người dùng, trình duyệt, độ phân giải, ngôn ngữ sử dụng

Gam màu phải thống nhất trong toàn bộ website

Phải làm cho đọc giả cảm nhận được kích thước của trang thông tin, biết họ đang ở đâu, có thể làm gì ?

*Lưu ý, hầu hết các trang web đều không vừa khớp với màn hình 14, 15 inch.

b. Xác định các kỹ thuật, công cụ thiết kế

Phụ thuộc nhiều yếu tố:

- Môi trường hosting
- Đội ngũ thiết kế
- Chi phí thiết kế
- Băng thông đường truyền

c. Cắt đoạn, tóm lược thông tin

Thông tin cần chia nhỏ, tóm lược lại vì:

- Người dùng có ít thời gian để đọc các tài liệu trên màn hình. Tuy nhiên cần lưu ý họ luôn có nhu cầu muốn tìm phần thông tin chủ định, không nên chia cắt quá nhỏ, tóm lược quá ngắn sẽ gây thất vọng
- Hình thức và cách tổ chức đồng nhất cho phép người dùng áp dụng kinh nghiệm tìm kiếm, khám phá thông tin và cho phép họ đoán được phần website mới, lạ sẽ được tổ chức như thế nào
- Thông tin ngắn gọn, súc tích thích hợp với màn hình máy tính (bị giới hạn tầm nhìn).
- Việc áp dụng phải linh động, nhất quán, với hệ thống logic và sự thuận tiện cho người dùng. Cách tốt nhất để phân chia và tổ chức thông tin là thực hiện theo bản chất của nội dung.

d. Xác định cấu trúc WebSite

Hệ thống phân cấp

Dùng để tổ chức các khối thông tin phức hợp, là hệ thống được dùng thông dụng nhất, gần với mô hình tổ chức thế giới thực nên dễ hình dung tổ chức website

Hệ thống các trang nối tiếp

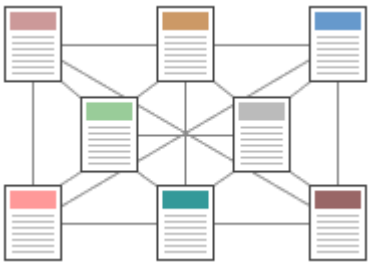
Được dùng để biểu diễn thông tin tuần tự, các bảng tường thuật nối tiếp theo thời gian, ví dụ như các thông tiin tra cứu tham khảo: tự điển, bácк khoa, tự điển thuật ngữ. Thích hợp cho hệ thống website nhỏ.

Ô Lưới

Từng đơn vị trong cấu trúc phải có cùng cấu trúc cho các chủ đề lớn và nhỏ, cấu trúc này khó hiểu khi xác định mối liên quan giữa các loại thông tin nhưng rất tốt đối với những đọc giả có kinh nghiệm, có sẵn kiến thức về hệ thống, chủ đề trong hệ thống

Mạng nhện

Mô hình này khai thác triệt để ưu điểm của hyperlink, tuy nhiên cấu trúc này phi thực tế nhất, khó hiểu, khó dự đoán cho người dùng, thích hợp với những site nhỏ, đọc giả chuyên nghiệp hoặc trình độ cao, tìm kiếm các kiến thức chuyên sâu



4. Các thành phần cơ bản của Website

a. Trang chủ (HomePage)

Tất cả các website đều được thiết lập xung quanh trang chủ (home page) giữ nhiệm vụ như một điểm xuất phát đến các trang web phúc tạp khác trong website. Trang chủ là địa chỉ web để hướng người dùng đến website của ta, là cái đầu tiên mà người dùng nhìn thấy khi truy cập đến website. Do đó trang chủ được thiết kế thích hợp là điều kiện cơ bản để website thành công.

b. Hệ thống Menu, Logo, định danh

Hệ thống menu phải rõ ràng, đầy đủ sẽ giúp đọc giả hình dung được cấu trúc, tổ chức website. Ngoài ra ta cần quan tâm đến vị trí, các thể hiện (có hay không có hiệu ứng), vị trí logo, định danh phải cố định nhất quán

c. Các trang thành viên

Ta xây dựng theo cấu trúc cơ bản của website, nhất quán, phù hợp với các thuộc tính đã được định dạng trước

III. Một số nguyên tắc khi phát triển website.

1 . Tổ chức website chặt chẽ và dễ sử dụng.

Website của bạn cần có cấu trúc càng rõ ràng dễ hiểu càng tốt. Điều quan trọng ở đây là phải làm sao để khách hàng thấy được ngay các thông tin mà họ hi vọng có thể thu được từ website của bạn. Nếu website của bạn có quá nhiều thông tin, bạn có thể làm cho trang chủ đơn giản bằng cách thiết kế bảng nội dung, bảng này cũng nên hết sức đơn giản và dễ sử dụng. Đồng thời sử dụng những từ và đoạn ngắn gọn dễ hiểu để thu hút người đọc.

2 . Sử dụng từ ngữ dễ hiểu.

Một ai đó sẽ không thể theo dõi được quảng cáo bán hàng của bạn cũng như mua những mặt hàng mà bạn đang cung cấp nếu như họ không thể hiểu được những gì bạn đang nói. Sử dụng những lời lẽ hoa mỹ để tán dương những sản phẩm bạn cung cấp thì rất dễ nhưng bạn sẽ không thể biết được có bao nhiêu người tới website và dự định của họ như thế nào?

Có thể bạn cung cấp những sản phẩm dịch vụ chất lượng tốt nhưng sẽ không ai mua nếu như họ không biết bạn đang chào bán những gì, hay không thể hiểu được lợi ích mà sản phẩm dịch vụ của bạn mang đến cho khách hàng. Hãy nhớ rằng khi một người đến thăm website của bạn, có thể anh ta chưa biết bạn là ai?. bạn đang chào bán sản phẩm gì?. Bạn phải giúp khách hàng hiểu rõ những vấn đề này trong thời gian ngắn nhất. Hãy dùng các câu ngắn gọn, cô đọng và giữ kiểu thiết kế cố định đối với tất cả các trang con.

3. Dễ dàng khám phá các đường link.

Bạn hãy tạo các đường link bằng chữ hay biểu tượng ở tất cả các trang con để mọi người có thể xem lại hoặc xem tiếp mà không phải sử dụng đến nút "Back" hay "Forward" của trình duyệt. Bạn cũng cần nhớ là phải có những chữ thay thế tất cả các đồ hoạ và đường liên kết trong trang của bạn. Đây là những từ sẽ xuất hiện thay thế đồ họa khi tuỳ chọn đồ hoạ trong trình duyệt bị tắt hoặc khi người ta nhấn nút "Stop" trước khi trang được tải về đầy đủ.

4. Thời gian tải về nhanh.

Bạn đừng nghĩ rằng tất cả mọi người đều sử dụng một đường truyền Internet có tốc độ cao. Liệu bạn có muốn mình phải đợi 10 phút để tải một trang về trước khi xem trang đó không?. Chắc chắn là không, vì thế bạn đừng hy vọng khách hàng sẽ đợi. Bạn cũng nên nhớ rằng 30 giây trước màn hình giống như 10 phút vậy.

Sử dụng đồ hoạ để trang trí là rất tốt nhưng đừng lạm dụng. Nếu bạn cần nhiều hình ảnh và đồ hoạ lớn thì nên có một biểu tượng nhỏ sẽ liên kết với hình ảnh đồng thời nhắc nhở người xem cần phải đợi. Sử dụng video và audio trong trang như một công cụ để bán hàng là ý tưởng khá hay, tuy nhiên bạn không nên sử dụng bởi hiện tại trừ các tỉnh thành lớn có đường truyền tốc độ cao ADSL hay cáp quang, vẫn còn đa số người vẫn đang sử dụng đường truyền Dial-Up với modem 28.8, 33.6 và 56.6.

Nhân tố thời gian là vô cùng quan trọng vì mọi người thường không kiên nhẫn khi vào mạng. Nếu trang của bạn phải mất thời gian quá lâu để tải về thì khách hàng có thể nhấn chuột và bỏ đi. Đừng để mất khách hàng chỉ vì trang web của bạn tải về quá chậm.

Hãy tăng tốc độ truyền của các trang web lên bằng cách:

Giảm kích cỡ đồ hoạ trong trang web của bạn. Nhiều file đồ hoạ không nhất thiết phải có kích cỡ như trên các trang web thông thường. Bạn chỉ cần 72 dpi cho độ phân giải của màn hình và đồ hoạ cũng chỉ cần 256 màu. Một đồ hoạ kích cỡ nhỏ 4" - 2" không nên lớn hơn 10K. Thu nhỏ kích cỡ đồ hoạ, độ sâu của màu. Hãy để chế độ phân giải đồ hoạ và hình ảnh nền ở mức 256 màu. Nếu bạn rất cần một đồ hoạ lớn thì bạn có thể cung cấp cho người xem một hình ảnh nhỏ với nút "phóng to" để xem tiếp một hình ảnh lớn hơn.

Quy định cụ thể kích cỡ file đồ họa trong mã HTML. Nếu bạn quy định cụ thể kích cỡ các file đồ hoạ trong mã HTML, trình duyệt web sẽ rút ngắn kích cỡ của trang nếu cần thiết và nơi hình ảnh sẽ xuất hiện, hiển thị văn bản và để một khoảng trống cho file đồ hoạ tải về.

Giảm số file trong trang web của bạn (cả file đồ hoạ và HTML kết hợp với nhau). Mọi người luôn xem nhẹ thủ thuật quản lý trang: giảm số file chứa trong website của bạn. Mọi người thường có tối đa bốn kết nối (socket) trong trình duyệt web của họ. Mỗi một socket sẽ cho phép chuyển một file về máy tính của bạn, vì thế nếu bạn có 4 socket thì bạn có thể tải cùng lúc 4 file về. Nếu bạn có 6 ảnh trong trang chủ và một file HTML thì tất cả là có 7 file

cần phải tải về. Trình duyệt sẽ tải 4 file về trước , sau khi tải xong một file socket sẽ tải tiếp file còn lại. Nói cách khác file thứ 5 sẽ chỉ được tải về khi file thứ nhất được tải xong. Và file thứ 6 sẽ chưa được tải về cho đến khi quá trình tải file thứ hai hoàn thành... quá trình tải về có thể kéo dài nếu có quá nhiều file đặc biệt khi những file này rất lớn. Theo như nguyên tắc, (giả sử đồ hoạ của bạn có kích cỡ khiêm tốn 5-12K) bạn hãy cố gắng có dưới 5 file mỗi trang.

5. Nội dung không có hình ảnh.

Nhiều người sử dụng ảnh "GIFS" và JavaScripts để tạo các logo và ký tự chạy ngang màn hình hay những gì tương tự. Điều này không chỉ làm tăng thời gian tải về mà còn làm người xem xao lãng nội dung bán hàng của bạn. Những người trên Internet là những con người của thông tin vì vậy bạn hãy chắc chắn rằng mình đang dành thời gian cho những thông tin có chất lượng chứ không phải là những hình ảnh vô bổ.

Nếu bạn có một nội dung vô giá trong trang web, hãy làm cho nó dễ đọc. Hãy chia thành những đoạn quan trọng, gạch chân hoặc bôi đậm những câu quan trọng trong từng đoạn và bạn đừng ngại trang trí với một số màu.

Tô màu văn bản thay thế file đồ hoạ nếu có thể. Nói cách khác thay vì sử dụng một file đồ họa để gây sự chú ý, bạn có thể sử dụng văn bản có màu sắc khác nhau.

Có thể bạn muốn cung cấpthông tin miễn phí dưới dạng bài báo hay bài phóng sự, và sau đó cố gắng bán hàng. Nếu bạn muốn cung cấp cho người sử dụng những thông tin bổ ích (với mục đích thu hút khách hàng), hãy thêm những nội dung có chất lượng chứ không phải là những hình ảnh bên ngoài. Trong trường hợp đó một chữ đáng giá hàng nghìn hình ảnh.

Thậm chí bạn muốn trang của mình sinh động hơn một chút (có những biểu tượng biến hình, các dòng chữ bôi đậm...) nhằm thu hút mọi người tiếp tục quan tâm tới sản phẩm và dịch vụ của bạn. Công việc của bạn chính là kiểm tra những kết quả mà khách hàng xem đem lại.

Tất cả sẽ phụ thuộc vào những sản phẩm và dịch vụ bạn đang bán cũng như đối tượng khách hàng bạn cần tiếp thị hay thị trường mục tiêu của bạn. Nếu bạn đang tiếp thị cho lớp trẻ thì sự sinh động của website sẽ làm tăng doanh số bán hàng. Nhưng nếu đối tượng khách hàng của bạn chủ yếu là những nhà kinh doanh có trình độ thì yếu tố sinh động đó có thể làm bạn giống như một hoạ sĩ nửa mùa. Đối với đối tượng khách hàng này bạn cần thu hút họ bằng những sự kiện, con số, sự trung thực và những lợi ích rõ ràng.

6. Dễ theo dõi "quá trình bán hàng".

Bạn phải tạo điều kiện để khách hàng hiểu rõ những lợi ích của sản phẩm và dịch vụ của bạn đem lại cũng như cung cấp cho khách hàng phương thức đặt hàng thuận tiện nhất. Liệu bạn đã xây dựng được uy tín đối với khách hàng trước khi bạn yêu cầu họ đặt hàng

chưa? Bạn đã tạo cho khách hàng sự yêu thích và hứng thú trước khi bạn mời họ đặt hàng chưa?. Bạn đã cung cấp cho khách hàng một số cách đặt hàng thuận tiện cả trên mạng và ngoài mạng chưa?. Và liệu bạn đã hướng dẫn khách xem tất cả từng bước một chưa?.

7. Tương thích với đa số trình duyệt web.

Nếu bạn sử dụng bảng biểu hãy xem xét cẩn thận việc nó sẽ hiển thị như thế nào ở các trình duyệt khác nhau (ví dụ Internet Explorer, Netscape) và ở tất cả các cấp độ phân giải (ví dụ 800 x 600, 1024 x 768, 1280 x 1024, 1400 x 1050).

8. Một số vấn đề quan trọng khác khi thiết kế website.

Đọc và kiểm tra cẩn thận tất cả các nội dung. Nếu bạn không quan tâm tới việc kiểm tra lỗi chính tả, người sử dụng sẽ nghi vấn làm sao họ có thể giao tiền của mình cho một công ty không thể tự sửa lỗi chính tả cho trang web của mình?. Hãy nhờ một người bạn hoặc đồng nghiệp đọc và sửa giúp bạn bởi họ có thể tìm thấy những lỗi mà bạn không bao giờ phát hiện ra được.

Trước khi đưa mọi việc vào hoạt động bạn cần có một đợt kiểm tra toàn bộ website (các đường liên kết, thời gian tải, form bán hàng...) và cố gắng kiểm tra bằng nhiều phương pháp.

Một điều hết sức quan trọng là bạn không nên nói ngay cho người xem biết bạn đang cố gắng bán hàng cho họ. Bất kể bạn đang có sản phẩm gì, cho dù sản phẩm của bạn có tốt như thế nào đi nữa thì hầu hết mọi người sẽ không ở lại trang của bạn nếu họ biết họ đang bị dụ dỗ mua hàng. Bạn cần để họ đọc, nhận ra được những lợi ích bạn sẽ đem lại cho họ và sau đó chỉ nên để họ biết rằng sản phẩm đó đang có bán. Nếu bạn thực hiện được điều này thì có nghĩa là khách hàng sẽ tự tìm thấy và mua sản phẩm bạn cung cấp.

Một yếu tố thành công khác trong marketing trực tuyến là bạn phải có khả năng chấp nhận được các giao dịch buôn bán trực tuyến. Khi bạn cung cấp sản phẩm thông tin thì bạn nên chào bán duới dạng điện tử thông qua email.

Nếu trang web của bạn lớn hơn 50K thì bạn hãy đặt một ghi chú nhỏ trên cùng của trang để thuyết phục khách hàng nên kiên nhẫn trong khi đồ họa được tải về (câu này sẽ hiện lên khi trang web của bạn tải về). Thậm chí ngay cả khi tại các trang web thử nghiệm mà bạn thấy không mất nhiều thời gian để tải về nhưng bạn cũng cần nhớ rằng khách hàng cũng có thể có đường kết nối chậm do đó làm tăng thời gian tải. Bạn không nên để khách hàng ra đi chỉ vì thời gian tải quá lâu mà bạn không thông báo về việc họ phải đợi.

Chương 2: Giới thiệu về ngôn ngữ HTML

I. Khái niệm cơ bản về html

1. HTML là gì?

HTML (viết tắt cho *HyperText Markup Language*, tức là "Ngôn ngữ Đánh dấu Siêu văn bản") do Tim Berner Lee phát minh và được W3C (World Wide Web Consortium) đưa thành chuẩn năm 1994. HTML là một ngôn ngữ đánh dấu được thiết kế ra để tạo nên các trang web được trình bày trên World Wide Web.

2. Thẻ HTML

HTML sử dụng các thẻ (tags) để định dạng dữ liệu. Các thẻ HTML không phân biệt chữ hoa, chữ thường. Các trình duyệt thường không báo lỗi Cú pháp HTML. Nếu viết sai thì kết quả hiển thị không đúng với dự định ban đầu.

HTML có 2 loại thẻ: thẻ đóng và thẻ mở

- Thẻ mở: **<tên_thẻ>**

Ví dụ 2.1: <html>, <body>, ,…

- Thẻ đóng tương ứng: **</tên_thẻ>**

Ví dụ 2.2: </html>, </body>, ,…

Có nhiều thẻ HTML, mỗi thẻ có 1 tên và mang ý nghĩa khác nhau.

* Chú ý: luôn có thẻ mở nhưng có thể không có thẻ đóng tương ứng. **Ví dụ:** <img> không có thẻ đóng

3. Cần gì để tạo một trang web

Có thể tạo trang HTML bằng bất cứ trình soạn thảo nào như Notepad, EditPlus,…Có nhiều trình soạn thảo HTML cho phép người sử dụng thực hiện một cách trực quan, kết quả sinh ra HTML tương ứng như:

- Microsoft FrontPage, notepad, notepad++
- Macromedia Dreamweaver

Trang HTML có phần mở rộng là .htm hoặc .html

II. Các thẻ định cấu trúc tài liệu

1. Thẻ html

Cặp thẻ này được sử dụng để xác nhận một tài liệu là tài liệu HTML, tức là nó có sử dụng các thẻ HTML để trình bày. Toàn bộ nội dung của tài liệu được đặt giữa cặp thẻ này.

Cú pháp:

```
<html>
    <!-- ….nội dung thẻ html -->
</html>
```

Trình duyệt sẽ xem các tài liệu không sử dụng thẻ **<html>** như những tệp tin văn bản bình thường.

2. Thẻ head

Thẻ **head** được dùng để xác định phần mở đầu cho tài liệu.

Cú pháp:

```
<head>
    <!-- Phần khai báo -->
</head>
```

3. Thẻ title

Cặp thẻ này chỉ có thể sử dụng trong phần mở đầu của tài liệu, tức là nó phải nằm trong thẻ phạm vi giới hạn bởi cặp thẻ <**head**>.

Cú pháp:

```
<title> Tiêu đề trang web </title>
```

4. Thẻ body

Thẻ này được sử dụng để xác định phần nội dung chính của tài liệu - phần thân (body) của tài liệu.

Trong phần thân có thể chứa các thông tin định dạng nhất định để đặt ảnh nền cho tài liệu, màu nền, màu văn bản siêu liên kết, đặt lề cho trang tài liệu... Những thông tin này được đặt ở phần tham số của thẻ.

Cú pháp:

```
 <body>
<!--… nội dung tài liệu html -->
</body>
```

Trên đây là Cú pháp cơ bản của thẻ **body**, tuy nhiên bắt đầu từ HTML 3.2 thì có nhiều thuộc tính được sử dụng trong thẻ **body**. Sau đây là các thuộc tính chính:

Background = Đặt một ảnh nào đó làm ảnh nền (background) cho văn bản. Giá trị của tham số này (phần sau dấu bằng) là URL của file ảnh. Nếu kích thước

	ảnh nhỏ hơn cửa sổ trình duyệt thì toàn bộ màn hình cửa sổ trình duyệt sẽ được lát kín bằng nhiều ảnh.
Bgcolor=	Đặt mầu nền cho trang khi hiển thị. Nếu cả hai tham số **background** và **bgcolor** cùng có giá trị thì trình duyệt sẽ hiển thị mầu nền trước, sau đó mới tải ảnh lên phía trên.
Text	Xác định màu chữ của văn bản, kể cả các đề mục.
Alink = Vlink = Link =	Xác định màu sắc cho các siêu liên kết trong văn bản. Tương ứng, alink (*active link*) là liên kết đang được kích hoạt - tức là khi đã được kích chuột lên; vlink (*visited link*) chỉ liên kết đã từng được kích hoạt;

Như vậy một tài liệu HTML cơ bản có cấu trúc như sau:

```
<html>
<head>
<title>tiêu đề của tài liệu</title>
</head>
<body các tham số nếu có>
... nội dung của tài liệu
</body>
</html>
```

III. Các thẻ định dạng khối

1. Thẻ định dạng khối văn bản <p>

Thẻ <**P**> được sử dụng để định dạng một đoạn văn bản.

Cú pháp:

```
<p> Nội dung đoạn văn bản </p>
```

2. Các thẻ định dạng đề mục h1/h2/h3/h4/h5/h6

HTML hỗ trợ 6 mức đề mục. Chú ý rằng đề mục chỉ là các chỉ dẫn định dạng về mặt logic, tức là mỗi trình duyệt sẽ thể hiện đề mục dưới một khuôn dạng thích hợp. Có thể ở trình duyệt này là font chữ 14 point nhưng sang trình duyệt khác là font chữ 20 point. Đề mục cấp 1 là cao nhất và giảm dần đến cấp 6. Thông thường văn bản ở đề mục cấp 5 hay cấp 6 thường có kích thước nhỏ hơn văn bản thông thường.

Dưới đây là các thẻ dùng để định dạng văn bản ở dạng đề mục:

…	Định dạng đề mục cấp 1
…	Định dạng đề mục cấp 2
…	Định dạng đề mục cấp 3
…	Định dạng đề mục cấp 4

<h5>…</h5> Định dạng đề mục cấp 5
<h6>…</h6> Định dạng đề mục cấp 6

3. Thẻ xuống dòng

Thẻ này không có thẻ kết thúc, nó có tác dụng chuyển sang dòng mới. Lưu ý, nội dung văn bản trong tài liệu HTML sẽ được trình duyệt Web thể hiện liên tục, các khoảng trắng liền nhau, các ký tự tab, ký tự xuống dòng đều được coi như một khoảng trắng. Để xuống dòng trong tài liệu, bạn phải sử dụng thẻ <**br**> hoặc </**br**>

Ví dụ: Sử dụng thẻ
 để định dạng xuống dòng văn bản

```
<body>
Tên đăng nhập: </br> Mật khẩu:
</body>
```

4. Thẻ pre và thẻ

Để giới hạn đoạn văn bản đã được định dạng sẵn bạn có thể sử dụng thẻ <**pre**>hoặc ** **. Văn bản ở giữa hai thẻ này sẽ được thể hiện giống hệt như khi chúng được đánh vào, ví dụ dấu xuống dòng trong đoạn văn bản giới hạn bởi thẻ <**pre**> hoặc ** ** sẽ có ý nghĩa chuyển sang dòng mới (trình duyệt sẽ không coi chúng như dấu cách)

Cú pháp:

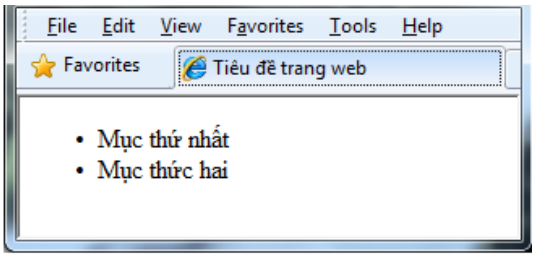
```
<pre> Văn bản được định dạng </pre>
<div> Văn bản được định dạng </div>
```

IV. Các thẻ định dạng danh sách

Kiểu 1: Danh sách không sắp xếp

Ví dụ:

```
<ul>
<li> Mục thứ nhất
<li> Mục thức hai
</ul>
```



Kiểu 2: Danh sách có sắp xếp <**ol**>, mỗi mục trong danh sách được sắp xếp thứ tự.

Cú pháp:

```
<ol type =1/a/A/i/I>
<li> Mục thứ nhất
```

```
<li> Mục thứ hai
<li> Mục thứ ba
</ol>
```

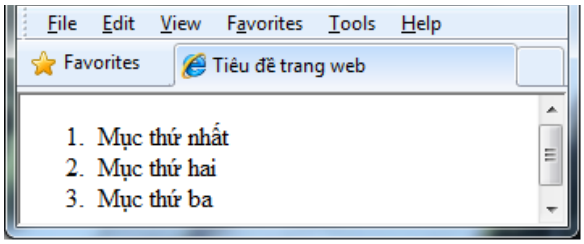
type =*1* Các mục được sắp xếp theo thứ tự 1, 2, 3...
=*a* Các mục được sắp xếp theo thứ tự a, b, c...
=*A* Các mục được sắp xếp theo thứ tự A, B, C...
=*i* Các mục được sắp xếp theo thứ tự i, ii, iii...
=*I* Các mục được sắp xếp theo thứ tự I, II, III...

Ngoài ra còn thuộc tính **START**= xác định giá trị khởi đầu cho danh sách.

Thẻ < **LI** > có thuộc tính **TYPE=** xác định ký hiệu đầu dòng (bullet) đứng trước mỗi mục trong danh sách. Thuộc tính này có thể nhận các giá trị : ***disc*** (chấm tròn đậm); ***circle*** (vòng tròn); ***square*** (hình vuông).

Ví dụ:

```
<ol type =1>
<li> Mục thứ nhất
<li> Mục thứ hai
<li> Mục thứ ba
</ol>
```



Kiểu 3: Danh sách thực đơn <**menu**>

Kiểu 4: Danh sách phân cấp <**dir**>

V. Các thẻ định dạng ký tự

1. Các thẻ định dạng in ký tự

Sau đây là các thẻ được sử dụng để quy định các thuộc tính như in nghiêng, in đậm, gạch chân... cho các ký tự, văn bản khi được thể hiện trên trình duyệt.

… **In chữ đậm**
** …**
** …** *In chữ nghiêng*
** … **
<u>… </u> <u>In chữ gạch chân</u>
<S> ... </S> ~~In chữ bị gạch ngang.~~
<strike> ... </strike>
<big> ... </big> In chữ lớn hơn bình thường bằng cách tăng kích thước font hiện thời lên một.
<**small**> ... </**small**> In chữ nhỏ hơn bình thường bằng cách giảm kích thước font hiện thời đi một.

<**sup**> ... </**sup**>	Định dạng chỉ số trên (SuperScript)
<**sub**> ... </**sub**>	Định dạng chỉ số dưới (SubScript)
<basefont>	Định nghĩa kích thước font chữ được sử dụng cho đến hết văn bản. Thẻ này chỉ có một tham số size= xác định cỡ chữ. Thẻ <basefont> không có thẻ kết thúc.
<**font**> ... </**font**>	Chọn kiểu chữ hiển thị. Trong thẻ này có thể đặt hai tham số size= hoặc color= xác định cỡ chữ và màu sắc đoạn văn bản nằm giữa hai thẻ. Kích thước có thể là tuyệt đối (nhận giá trị từ 1 đến 7) hoặc tương đối

2. Căn lề văn bản trong trang Web

Trong trình bày trang Web của mình các bạn luôn phải chú ý đến việc căn lề các văn bản để trang Web có được một bố cục đẹp. Một số các thẻ định dạng như **p**, **hn**, **img**... đều có tham số **ALIGN** cho phép bạn căn lề các văn bản nằm trong phạm vi giới hạn bởi của các thẻ đó.

Các giá trị cho tham số **align** = "left | center | right";

Ngoài ra, chúng ta có thể sử dụng thẻ **center** để căn giữa trang một khối văn bản.

Cú pháp:

```
<center> Văn bản sẽ được căn giữa trang</center>
```

3. Các ký tự đặc biệt

Ký tự & được sử dụng để chỉ chuỗi ký tự đi sau được xem là một thực thể duy nhất. Ký tự ; được sử dụng để tách các ký tự trong một từ.

Ký tự	Mã ASCII	Tên chuỗi
<	&#060	&lt
>	&#062	&gt
&	&#038	&amp

4. Sử dụng màu sắc trong thiết kế các trang Web

Một màu được tổng hợp từ ba thành phần màu chính, đó là: Đỏ (Red), Xanh lá cây (Green), Xanh nước biển (Blue). Trong HTML một giá trị màu là một số nguyên dạng hexa (hệ đếm cơ số 16) có định dạng như sau:

#RRGGBB

trong đó:

RR - là giá trị màu Đỏ.

GG - là giá trị màu Xanh lá cây.

BB - là giá trị màu Xanh nước biển.

Màu sắc có thể được xác định qua thuộc tính bgcolor= hay color=. Sau dấu bằng có thể là giá trị RGB hay tên tiếng Anh của màu. Với tên tiếng Anh, ta chỉ có thể chỉ ra 16 màu trong khi với giá trị RGB ta có thể chỉ tới 256 màu.

Sau đây là một số giá trị màu cơ bản:

Màu sắc	Giá trị	Tên tiếng Anh
Đỏ	#FF0000	RED
Đỏ sẫm	#8B0000	DARKRED
Xanh lá cây	#00FF00	GREEN
Xanh nhạt	#90EE90	LIGHTGREEN
Xanh nước biển	#0000FF	BLUE
Vàng	#FFFF00	YELLOW
Vàng nhạt	#FFFFE0	LIGHTYELLOW
Trắng	#FFFFFF	WHITE
Đen	#000000	BLACK
Xám	#808080	GRAY
Nâu	#A52A2A	BROWN
Tím	#FF00FF	MAGENTA
Tím nhạt	#EE82EE	VIOLET
Hồng	#FFC0CB	PINK
Da cam	#FFA500	ORANGE
Màu đồng phục hải quân	#000080	NAVY
	#4169E1	ROYALBLUE
	#7FFFD4	AQUAMARINE

Cú pháp:

```
<body >
.... phần nội dung của tài liệu được đặt ở đây
</body>
```

Một số thuộc tính cơ bản của thẻ body:

Các tham số ý nghĩa

link	Chỉ định màu của văn bản siêu liên kết
alink	Chỉ định màu của văn bản siêu liên kết đang đang chọn
vlink	Chỉ định màu của văn bản siêu liên kết đã từng mở
background	Chỉ định địa chỉ của ảnh dùng làm nền
bgcolor	Chỉ định màu nền
text	Chỉ định màu của văn bản trong tài liệu
scroll	YES/NO - Xác định có hay không thanh cuộn
topmargin	Lề trên
rightmargin	Lề phải
leftmargin	Lề trái

5. Chọn kiểu chữ cho văn bản

Cú pháp:

```
<font
face      = font-name
color= color
size      = n >
...
</font>
```

6. Khái niệm văn bản siêu liên kết

Văn bản siêu liên kết hay còn gọi là siêu văn bản là một từ, một cụm từ hay một câu trên trang Web được dùng để liên kết tới một trang Web khác. Siêu văn bản là môi trường trong đó chứa các liên kết (link) của các thông tin. Do WWW cấu thành từ nhiều hệ thống khác nhau, cần phải có một quy tắc đặt tên thống nhất cho tất cả các văn bản trên Web. Quy tắc đặt tên đó là URL (Universal Resource Locator).

Các thành phần của URL được minh hoạ ở hình trên.

Dịch vụ: Là thành phần bắt buộc của URL. Nó xác định cách thức trình duyệt của máy khách liên lạc với máy phục vụ như thế nào để nhận dữ liệu. Có nhiều dịch vụ như **http, wais, ftp, gopher, telnet**.

Tên hệ thống : Là thành phần bắt buộc của URL. Có thể là tên miền đầy đủ của máy phục vụ hoặc chỉ là một phần tên đầy đủ – trường hợp này xảy ra khi văn bản được yêu cầu vẫn nằm trên miền của bạn. Tuy nhiên nên sử dụng đường dẫn đầy đủ.

Cổng : Không là thành phần bắt buộc của URL. Cổng là địa chỉ socket của mạng dành cho một giao thức cụ thể. Giao thức http ngầm định nối với cổng 8080.

Đường dẫn thư mục : Là thành phần bắt buộc của URL. Phải chỉ ra đường dẫn tới file yêu cầu khi kết nối với bất kỳ hệ thống nào. Có thể đường dẫn trong URL khác với đường dẫn

thực sự trong hệ thống máy phục vụ. Tuy nhiên có thể rút gọn đường dẫn bằng cách đặt biệt danh (alias). Các thư mục trong đường dẫn cách nhau bởi dấu gạch chéo (/).

Tên file : Không là thành phần bắt buộc của URL. Thông thường máy phục vụ được cấu hình sao cho nếu không chỉ ra tên file thì sẽ trả về file ngầm định trên thư mục được yêu cầu. File này thường có tên là index.html, index.htm, default.html hay default.htm. Nếu cũng không có các file này thì thường kết quả trả về là danh sách liệt kê các file hay thư mục con trong thư mục được yêu cầu

Các tham số : Không là thành phần bắt buộc của URL. Nếu URL là yêu cầu tìm kiếm trên một CSDL thì truy vấn sẽ gắn vào URL, đó chính là đoạn mã đằng sau dấu chấm hỏi (?).URL cũng có thể trả lại thông tin được thu thập từ form. Trong trường hợp dấu thăng (#) xuất hiện đoạn mã đăng sau là tên của một vị trí (location) trong file được chỉ ra.

Để tạo ra một siêu văn bản chúng ta sử dụng thẻ <**A**>.

Cú pháp:

```
<a …> Liên kết </a>
```

Một số thuộc tính của thẻ a

href	Địa chỉ của trang Web được liên kết, là một URL nào đó.
name	Đặt tên cho vị trí đặt thẻ.
tableindex	Thứ tự di chuyển khi ấn phím Tab
title	Văn bản hiển thị khi di chuột trên siêu liên kết.
target	Mở trang Web được liên trong một cửa sổ mới (*_blank*) hoặc trong cửa sổ hiện tại (*_self*), trong một frame (tên frame).

7. Địa chỉ tương đối

URL được trình bày ở trên là URL tuyệt đối. Ngoài ra còn có URL tương đối hay còn gọi là URL không đầy đủ. Địa chỉ tương đối sử dụng sự khác biệt tương đối giữa văn bản hiện thời và văn bản cần tham chiếu tới. Các thành phần trong URL được ngăn cách bằng ký tự ngăn cách (ký tự gạch chéo /). Để tạo ra URL tương đối, đầu tiên phải sử dụng ký tự ngăn cách. URL đầy đủ hiện tại sẽ được sử dụng để tạo nên URL đầy đủ mới. Nguyên tắc là các thành phần bên trái dấu ngăn cách của URL hiện tại được giữ nguyên, các thành phần bên phải được thay thế bằng thành phần URL tương đối. Chú ý rằng trình duyệt không gửi URL tương đối, nó bổ sung vào URL cơ sở đã xác định trước thành phần URL tương đối xác định sau thuộc tính href=. Ký tự đầu tiên sau dấu bằng sẽ xác định các thành phần nào của URL hiện tại sẽ tham gia để tạo nên URL mới.

Ví dụ, nếu URL đầy đủ là: http://it-department.vnuh.edu.vn/HTML/index.htm thì:

Dấu hai chấm (:) chỉ dịch vụ giữ nguyên nhưng thay đổi phần còn lại. Ví dụ **://www.fpt.com/** sẽ tải trang chủ của máy phục vụ **www.fpt.com** với cùng dịch vụ http.

Dấu gạch chéo (/) chỉ dịch vụ và máy phục vụ giữ nguyên nhưng toàn bộ đường dẫn thay đổi. Ví dụ **/Javascript/index.htm** sẽ tải file **index.htm** của thư mục **Javascript** trên máy phục vụ **www.it-department.vnuh.edu.vn**.

Không có dấu phân cách chỉ có tên file là thay đổi. Ví dụ **index1.htm** sẽ tải file **index1.htm** ở trong thư mục HTML của máy phục vụ **www.it-department.vnuh.edu.vn**.

Dấu thăng (#): chỉ dịch vụ, máy phuc vụ, đường dẫn và cả tên file giữ nguyên, chỉ thay đổi vị trí trong file.

Do đường dẫn được xem là đơn vị độc lập nên có thể sử dụng phương pháp đường dẫn tương đối như trong UNIX hay MS-DOS (tức là **.** chỉ thư mục hiện tại còn **..** chỉ thư mục cha của thư mục hiện tại).

URL cơ sở có thể được xác định bằng thẻ <BASE>.

8. Kết nối mailto

Nếu đặt thuộc tính href= của thẻ <a> giá trị mailto:address@domain thì khi kích hoạt kết nối sẽ kích hoạt chức năng thư điện tử của trình duyệt.

<address>

trang web này được
<**a** href=”mailto:webmaster@vnuh.edu.vn” >
webmaster
<**a**> bảo trì
<\address>

9. Vẽ một đường thẳng nằm ngang

Cú pháp:

```
<hr>
```

Một số thuộc tính của thẻ hr

align	Căn lề (căn trái, căn phải, căn giữa)
color	Đặt màu cho đường thẳng
noshade	Không có bóng
size	Độ dày của đường thẳng
width	Chiều dài (tính theo pixel hoặc % của bề rộng cửa sổ trình duyệt).

Thẻ này giống như thẻ **BR**, nó cũng không có thẻ kết thúc tương ứng.

VI. Các thẻ chèn âm thanh, hình ảnh

1. Giới thiệu

Liên kết với file đa phương tiện cũng tương tự như liên kết bình thường. Tuy vậy phải đặt tên đúng cho file đa phương tiện. Phần mở rộng của file phải cho biết kiểu của file.

Kiểu	Phần mở rộng	Mô tả
Image/GIF	.gif	Viết tắt của Graphics Interchange Format. Khuôn dạng này xuất hiện khi mọi người có nhu cầu trao đổi ảnh trên nhiều hệ thống khác nhau. Nó được sử dụng trên tất cả các hệ thống hỗ trợ giao diện đồ hoạ. Định dạng GIF là định dạng chuẩn cho mọi trình duyệt WEB. Nhược điểm của nó là chỉ thể hiện được 256 màu. Mở rộng của chuẩn này là GIF89, được thêm nhiều chức năng cho các ứng dụng đặc biệt như làm ảnh nền trong suốt - tức là ảnh có thể nổi bằng cách làm màu nền giống với màu nền của trình duyệt.
Image/JPEG	.jpeg	Viết tắt của Joint Photographic Expert Group. Là khuôn dạng ảnh khác nhưng có thêm khả năng nén.Ưu điểm nổi bật của khuôn dạng này là lưu trữ được hàng triệu màu và độ nén cao nên kích thước file ảnh nhỏ hơn và thời gian download nhanh hơn. Nó là cơ sở cho khuôn dạng MPEG. Tất cả các trình duyệt đều có khả năng xem ảnh JPEG.
Image/TIFF	.tiff	Viết tắt của Tagged Image File Format. Được Microsoft thiết kế để quét ảnh từ máy quét cũng như tạo các ấn phẩm.
Text/HTML	.HTML, .htm	
PostScript	.eps, .ps	Được tạo ra để hiển thị và in các văn bản có chất lượng cao.
Adobe Acrobat	.pdf	Viết tắt của Portable Document Format. Acrobat cũng sử dụng các siêu liên kết ngay trong văn bản cũng giống như **HTML**. Từ phiên bản 2.0, các sản phẩm của Acrobat cho phép liên kết giữa nhiều văn bản. Ưu điểm lớn nhất của nó là khả năng WYSISYG.
Video/MPEG	.mpeg	Viết tắt của Motion Picture Expert Group, là định

		dạng dành cho các loại phim (video). Đây là khuôn dạng thông dụng nhất dành cho phim trên WEB.
Video/AVI	.avi	Là khuôn dnạg phim do Microsoft đưa ra.
Video/QuickTime	.mov	Do Apple Computer đưa ra, chuẩn video này được cho là có nhiều ưu điểm hơn MPEG và AVI. Mặc dù đã được tích hợp vào nhiều trình duyệt nhưng vẫn chưa phổ biến bằng hai loại định dạng trên.
Sound/AU	.au	Là khuôn dạng dành cho âm nhạc điện tử hết sức thông dụng được nhiều trình duyệt trên các hệ thống khác nhau hỗ trợ. File Midi được tổng hợp số hoá trực tiếp từ máy tính.
Sound/MIDI	.mid	Định dạng audio theo dòng. Một bất tiện khi sử dụng các định dạng khác là file âm thanh thường có kích thước lớn - do vậy thời gian tải xuống lâu, Trái lại audio dòng bắt đầu chơi ngay khi tải được một phần file trong khi vẫn tải về các phần khác.Mặc dù file theo định dạng này không nhỏ hơn so với các định dạng khác song chính khả
Sound/RealAudio	.ram	năng dòng đã khiến định dạng này phù hợp với khả năng chơi ngay lập tức.
VRML	.vrml	Viết tắt của Virtual Reality Modeling Language. Các file theo định dạng này cũng giống như **HTML.** Tuy nhiên do trình duyệt có thể hiển thị được cửa sổ 3 chiều nên người xem có thể cảm nhận được cảm giác ba chiều.

2. Đưa âm thanh vào một tài liệu HTML

Cú pháp:

```
<bgsound  src = url    loop = n  >
```

Thẻ này không có thẻ kết thúc. Để chơi lặp lại vô hạn cần chỉ định **loop** = -1 hoặc **loop** = *infinite*. Thẻ **bgsound** phải được đặt trong phần mở đầu (tức là nằm trong cặp thẻ **head**).

3. Chèn một hình ảnh, một đoạn video vào tài liệu HTML

Để chèn một file ảnh (.jpg, .gif, .bmp) hoặc video (.mpg, .avi) vào tài liệu HTML, bạn có thể sử dụng thẻ **img.**

Cú pháp:

```
<img …/>
```

Một số thuộc tính của thẻ img

align = top/ middle/ bottom/ left/ right	Căn hàng văn bản bao quanh ảnh
alt = text	Chỉ định văn bản sẽ được hiển thị nếu chức năng show picture của browser bị tắt đi hay hiển thị thay thế cho ảnh trên những trình duyệt không có khả năng hiển thị đồ hoạ. Văn bản này còn được gọi là nhãn của ảnh. Đối với trình duyệt có khả năng hỗ trợ đồ hoạ, dòng văn bản này sẽ hiện lên khi di chuột qua ảnh hay được hiển thị trong vùng của ảnh nếu ảnh chưa được tải về hết. Chú ý phải đặt văn bản trong hai dấu nháy kép nếu triong văn bản chứa dấu cách hay các ký tự đặc biệt - trong trường hợp ngược lại có thể bỏ dấu nháy kép.
border = n	Đặt kích thước đường viền được vẽ quanh ảnh (tính theo pixel).
src = url	Địa chỉ của file ảnh cần chèn vào tài liệu.
width/height	Chỉ định kích thước của ảnh được hiển thị.
hspace/vspace	Chỉ định khoảng trống xung quanh hình ảnh (tính theo pixel) theo bốn phía trên, dưới, trái, phải.
title = title	Văn bản sẽ hiển thị khi con chuột trỏ trên ảnh
dynsrc = url	Địa chỉ của file video.
start = fileopen/mouseover	Chỉ định file video sẽ được chơi khi tài liệu được mở hay khi trỏ con chuột vào nó. Có thể kết hợp cả hai giá trị này nhưng phải phân cách chúng bởi dấu phẩy.
loop = n/infinite	Chỉ định số lần chơi. Nếu LOOP = INFINITE thì file video sẽ được chơi vô hạn lần.

VII. Các thẻ định dạng bảng biểu

Sau đây là các thẻ tạo bảng chính:

<**table**> ... </**table**>	Định nghĩa một bảng
<**tr**> **...** </**tr**>	Định nghĩa một hàng trong bảng
<**td**> **...** </**td**>	Định nghĩa một ô trong hàng
<**th**> **...** </**th**>	Định nghĩa ô chứa tiêu đề của cột
<caption> ... </caption>	Tiêu đề của bảng

Cú pháp:

```
<table >
... định nghĩa các dòng
<tr >
```

```
    ... định nghĩa các ô trong dòng
<td >
... nội dung của ô
</td>
...
</tr>
...
</table>
```

ý nghĩa các tham số:

align / valign	Căn lề cho bảng và nội dung trong mỗi ô.
border	Kích thước đường kẻ chia ô trong bảng, được đo theo pixel. Giá trị 0 có nghĩa là không xác định lề, giữa các ô trong bảng chỉ có một khoảng trắng nhỏ để phân biệt. Nếu chỉ để border thì ngầm định border=1. Với những bảng có cấu trúc phức tạp, nên đặt lề để người xem có thể phân biệt rõ các dòng và cột.
bordercolor	Màu đường kẻ
bordercolordark	Màu phía tối và phía sáng cho đường kẻ nổi.
bordercolorlight	
background	Địa chỉ tới tệp ảnh dùng làm nền cho bảng
bgcolor	Màu nền
cellspacing	Khoảng cách giữa các ô trong bảng
cellpadding	Khoảng cách giữa nội dung và đường kẻ trong mỗi ô của bảng.
colspan	Chỉ định ô sẽ kéo dài trong bao nhiêu cột
rowspan	Chỉ định ô sẽ kéo dài trong bao nhiêu hàng

VIII. FORM

Form HTML là một phần của tài liệu, nó chứa các phần tử đặc biệt gọi là các điều khiển. Các điều khiển được sử dụng để nhập thông tin từ người dùng và cung cấp một số tương tác.

Các form cho phép người sử dụng nhập dữ liệu trên trang web thông qua các điều khiển (control). Dữ liệu này có thể được xác nhận hợp lệ từ phía máy khách và được chuyển đến máy chủ để xử lý thêm.

Tất cả các điều khiển đều có tên được quy định qua thuộc tính **name**. Một số điều khiển không cần lấy dữ liệu thì thuộc tính name không quan trọng

Sau đây ta sẽ tìm hiểu về các loại điều khiển

X

Hồ Diên Lợi

1. Form

From dùng để chứa mọi đối tượng khác. Để tạo form ta dùng thẻ:

<form>…</form>

Một số thuộc tính của form:

- **name**="tên_form"
- **action**="địa chỉ nhận dữ liệu"
- **method**="phương thức gửi dữ liệu":
    - **GET** : thông tin xử lý theo phương thức get được hiển thị lên url
    - **POST:** thông tin xử lý theo phương thức post không hiển thị lên url

2. Hộp nhập văn bản 1 dòng (Oneline Textbox)

Oneline Textbox dùng để nhập các văn bản ngắn (trên 1 dòng) hoặc mật khẩu, sử dụng thẻ **<input>** để đưa vào form

Các thuộc tính:

- **name**="tên_đt"
- **type="text"**: nhập văn bản thường
- **type="password"**: nhập mật khẩu
- **value**="giá trị mặc định"

Ví dụ 14

```
<html>
<head>
    <title>Registered form</title>
</head>
<body>
<form name="application_form" action="test.php" method="post">
    <h1> Registered form</h1>
     <p>User name: <input type="text" name="username" value="" size="30px" />
    <p>Password: <input type="password" name="pass" value="123456" size="30px" />
    </form>
</body>
</html>
```

3. Radio Button

Radio button cho phép chọn một lựa chọn trong một nhóm lựa chọn được đưa ra. Các điều khiển radio trong một nhóm phải có cùng tên. Vào một thời điểm, người dùng chỉ có thể chọn một lựa chọn. Các nút radio nên đặt thuộc thuộc tính giá trị. Sử dụng thẻ **<input>** để đưa đối tượng radio vào form, mỗi ô cần 1 thẻ

Thuộc tính:

- **name**="tên_đt": Các đối tượng cùng tên thì thuộc cùng nhóm.
- **type="radio"**
- **value**="giá trị": đây là giá trị chương trình sẽ nhận được nếu ta chọn ô này.
- **checked**: nếu có thì nút này mặc định được chọn

4. Checkbox

Checkbox cho phép người dùng có thể chọn một hoặc nhiều lựa chọn trong một nhóm lựa chọn được đưa ra bằng cách đánh dấu tích. Sử dụng thẻ **<input>** để đưa đối tượng checkbox vào form, mỗi ô cần 1 thẻ

Thuộc tính:

- **name**="tên_đối_tượng"
- **type="checkbox"**
- **value**="giá trị": đây là giá trị chương trình sẽ nhận được nếu nó được chọn
- **checked**: nếu có thì nút này mặc định được chọn

5. Nút lệnh (Button)

Cho phép người sử dụng ra lệnh thực hiện một số công việc nào đó. Có 3 loại nút thường dùng

- ***submit***: khi người dùng nhấp vào nút submit, dữ liệu tự động được chuyển đến vị trí được xác định trong thuộc tính ACTION
- ***reset***: đưa mọi dữ liệu về trạng thái mặc định
- ***normal***: người lập trình tự xử lý

Đưa đối tượng button vào form ta dùng thẻ **<input>**

Thuộc tính:

- **name**="tên_đối_tượng"
- **type="submit"**: nút submit

- **type="reset"**: nút reset
- **type="button"**: nút thông thường (normal)
- **value**="tiêu đề nút"

6. Combo Box (Drop-down menu)

Combo box gồm một danh sách có nhiều phần tử, ta có thể chọn 1 phần tử trong danh sách xổ xuống bằng cách kích vào mũi tên bên phải hộp danh sách. Tại một thời điểm chỉ có 1 phần tử được chọn

Thẻ tạo hộp danh sách:

<select>Danh sách phần tử**</select>**

Thuộc tính:

- **name**="tên_đối_tượng"

Thẻ tạo 1 phần tử trong danh sách:

<option>Tiêu đề phần tử**</option>**

Thuộc tính:

- **value**="giá trị": giá trị chương trình nhận được nếu phần tử được chọn
- **selected**: nếu có thì phần tử này mặc định được chọn

7. Listbox

Tương tự như combobox, listbox là một danh sách gồm nhiều phần tử, tuy nhiên ta có thể nhìn thấy và lựa chọn các phần tử cùng một lúc

Thẻ tạo listbox:

<select>…</select>

Thuộc tính: tương tự như combobox nhưng có 2 thuộc tính khác:

- **size**="số dòng"
- **multiple**: cho phép lựa chọn nhiều phần tử cùng lúc

Thẻ tạo 1 phần tử trong danh sách:

<option>Tiêu đề phần tử**</option>**

Thuộc tính:

- **value**="giá trị": giá trị chương trình nhận được nếu phần tử được chọn
- **selected**: nếu có thì phần tử này mặc định được chọn

8. Hộp nhập văn bản nhiều dòng (TextArea)

Textarea cho phép người dùng nhập văn bản dài trên nhiều dòng.

Thẻ tạo textarea:

<textarea>Nội dung mặc định</textarea>

Các thuộc tính:

- **name**="tên_đối_tượng"
- **rows**="số dòng"
- **cols**="số cột"

Trong đó:

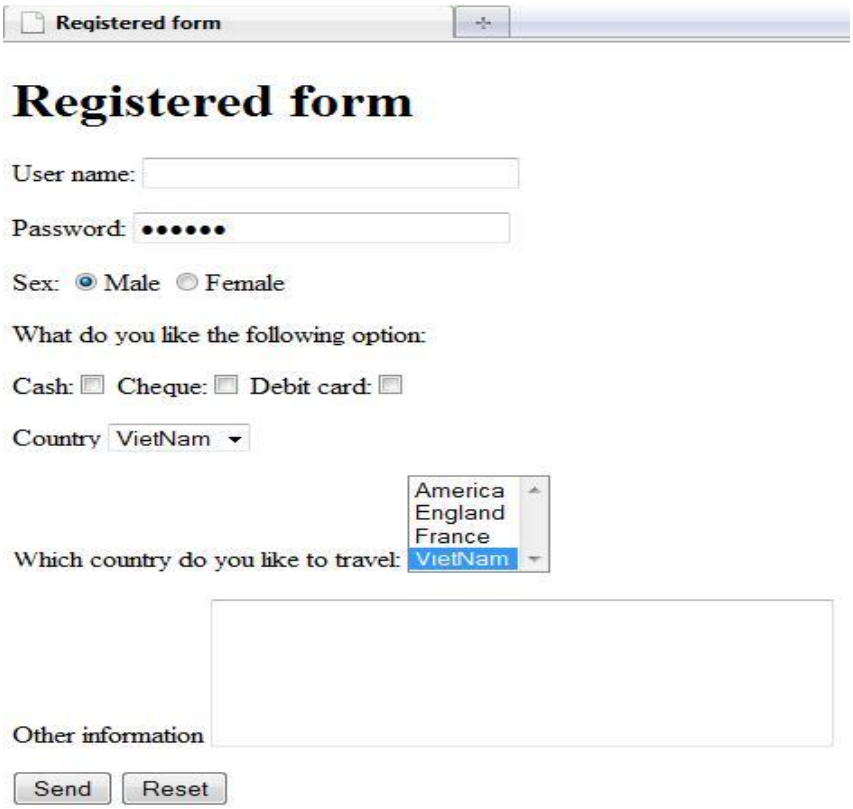
- rows: số dòng văn bản
- cols: số ký tự chuẩn trên dòng.

Ví dụ 15:

```
<html>
<head>
	<title>Registered form</title>
</head>
<body>
<form name="application_form" action="test.php" method="post">
 <h1> Registered form</h1>
<p>User name: <input type="text" name="username" value="" size="30px" />
<p>Password: <input type="password" name="pass" value="123456" size="30px" />
<p>Sex: <input type="radio" name="sex" checked="checked" value="M" />Male <input type="radio" name="sex" value="F" />Female</p>
<p>What do you like the following option:</p>
<p>Cash:<input type="checkbox" name="checkpay" value="cash" /> Cheque:<input type="checkbox" name="checkpay" value="cheque" /> Debit card:<input type="checkbox" name="checkpay" value="card" /></p>
<p>Country
	<select name="country">
	<option value="usa">America</option>
	<option value="eng">England</option>
	<option value="fra">France</option>
	<option value="vie" selected="selected">VietNam</option>
	</select>
</p>
<p>Which country do you like to travel:
```

```
        <select name="country" size="4" multiple="multiple">
        <option value="1">America</option>
              <option value="2">England</option>
          <option value="3">France</option>
          <option value="4" selected="selected">VietNam</option>
        </select>
</p>
        <p>Other information
        <textarea name="other_info" rows="5" cols="40"></textarea>
</p>
<p>
        <input type="submit" name="send" value="Send" />
          <input type="reset" name="reset" value="Reset" />
</p>
</form>
</body>
</html>
```

Kết quả hiển thị trên trình duyệt



IX. Một số thẻ đặc biệt

1. Thẻ <meta>

Thẻ <meta> được khái báo trong cặp thẻ <head>, thẻ <meta> thường được sử dụng để khái báo loại font sử dụng, tìm kiếm, xóa cache, chuyển trang…

a. Thẻ <meta> với font

Để sử dụng font Unicode đặc biệt Unicode tiếng việt trên trang web, chúng ta phải khai báo thẻ <meta> trong thẻ <head>.

```
<meta http-equiv = “Content-Type” content = “text/html; charset = UTF-8”>
```

b. Thẻ <meta> cho phép người dùng tìm kiếm.

Khi bạn đưa trang web của bạn lên internet, để người dùng có thể tìm thấy web site của bạn qua các công cụ trên như: Google, Yahoo …, khi đó chúng ta khai báo thẻ <meta> như sau

```
<meta http-equiv = “Content-Type” content = “text/html; charset = UTF-8”>
<META NAME = “author” CONTENT = “http://www.hutc.edu.vn/”>
```

Bạn có thể khái báo các thông tin khác của trang web để khi người dùng có thể tìm kiếm thông qua các thông tin này.

```
<META NAME = “keywords” CONTENT = “Công thương, kỹ thuật công nghiệp 2. ….. ”>
```

Các từ khóa này sẽ được đem so sánh với các từ khóa người dùng gõ và tìm kiếm trên Internet, nếu từ khóa người dùng tìm kiếm thuộc một trong số từ khóa của bạn đã khai báo trong thẻ này, web site của bạn sẽ được xuất hiện trong danh sách tìm kiếm được.

c. Thẻ <meta> dạng tự động chuyển đến URL

Để tự động chuyển đến địa chỉ URL hay UNC kế tiếp sau khi trang web nạp lên với thời gian nhất định, bạn có thể khái báo trong thẻ JavaScript.

Ví dụ:

```
 <html>
<head>
<title>Welcome to PHP and MySQL</title>
<META http-equiv=refresh content="8; URL=http://www.saigoninfotech.com">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>
<body bgcolor="#FFFFFF" text="#000000">
Trang này tự động chuyển đến trang <b>www.saigoninfotech.com</b> sau 8 giây
```

```
</body>
</html>
```

d. Thẻ <meta> dùng xóa cache

Thông thường sau khi nạp trang web nào đó lên trình duyệt web, nội dung của trang web đó có thể lưu vào trong bộ nhớ truy cập nhanh (cache).

Điều này có nghĩa là sau khi duyệt một vòng các trang web khác, bạn quay về gọi trang web đã truy cập trước đó, trình duyệt web nạp rất nhanh, do chúng đã lưu trang trong bộ nhớ cache.

Tuy nhiên, khi bạn là người phát triển ứng dụng web, có những trang web bạn phải xóa cache mỗi khi người dùng gọi nó. Nghĩa là, trang web này thường có thay đổi cấu trúc cho mỗi lần gọi, bạn cần khai báo thẻ <meta> như ví dụ sau:

Ví dụ:

```
<html>
<head>
<title>Welcome to PHP and MySQL</title>
<meta http-equiv="pragma" content="no-cache">
<meta http-equiv="expires" content="0">
<meta http-equiv="cache-control" content="no-cache">
</head>
<body bgcolor="#ffffff"> Xoa Cache
</body>
</html>
```

2. Thẻ <marquee>

Thẻ <marquee> cho phép bạn khái báo một chuỗi chuyển động theo chiều hướng khác nhau trên trang web. Thẻ này thường được sử dụng để quảng cáo một đề tài nào đó.

Có 4 chiều di chuyển của chuỗi, bạn có thể khái báo tùy thuộc vào các tham số UP, DOWN, BACK, RIGHT của thuộc tính DIRECTION.

Ví dụ:

```
<html>
<head>
<title>Welcome to php and mysql</title>
```

```
</head>
<body>
<marquee direction=right scrolldelay=2 scrollamount=1 width=100% >
<font face="arial" size="1" color=#ff33ff>
<br>ashley- ana - chanthaly - kathleen - lena </marquee><br>
<marquee direction=up scrolldelay=2 scrollamount=1 width=100%
style="filter:wave(add=5, phase=1, freq=5, strength=20);"><font face="arial" size="1"
color=#ff33ff>
<br>ashley
<br>ana
<br>chanthaly
<br>kathleen
<br>lena </marquee>
</body>
</html>
```

3. Thẻ <style>

Thẻ <style> cho phép bạn định dạng tất cả nội dung trình bày trên trang web theo một kiểu nhất định. Điều này có nghĩa là mọi thẻ trên trang web có khai báo sử dụng một phần tử nào đó được khái báo thẻ <style>, chúng sẽ có kiểu định dạng như bạn định nghĩa.

Bạn khái báo thẻ <style> trong thẻ <head>.

Ví dụ: Định dạng style, chèn ảnh nền và đặt hình nền không lặp.

```
<style>
        background-image: url("hinhnen.gif");
        background-repeat: no-repeat;
</style>
```

4. Thẻ <link>

Khi khai báo các phần tử trong trang style.css, bạn có thể khái báo chúng trong một trang web bằng thẻ <link>. Để sử dụng bạn cũng khái báo như trường hợp sử dụng phần tử trong ví dụ sau:

Ví dụ: Chèn tyle.css vào trong tài liệu html

```
<link href="tyle.css" type="text/css" rel="stylesheet" />
```

5. Thẻ <script>

Trong trang web, bạn muốn kiểm soát tất cả các hành động của người dùng, bạn cần khái báo vá sử dụng một số phương thức và thuộc tính của Client Script hay các phương thức do bạn định nghĩa.

Để có thể khái báo kịch bản trên trang web, bạn sử dụng thẻ <script> với tên ngôn ngữ chỉ định JavaScript hay VBScript.

Cú pháp:

```
<script language = “javascript”>
// mã javascript
<script>
<script language = “vbscript”>
// mã vbscript
<script>
```

Ngoài ra trong trường hợp có nhiều phương thức do bạn định nghĩa được sử dụng chung trong nhiều trang web, bạn cũng có thể khai báo chúng trong một tập tin có tên mở rộng .js hay .vb. Sau đó bạn có thể chèn tập tin này và sử dụng như cách chèn trực tiếp.

Chương 3: Thiết kế CSS

I. Giới thiệu về CSS

CSS (Cascading Style Sheets) được hiểu một cách đơn giản đó là cách mà chúng ta thêm các kiểu hiển thị (font chữ, kích thước, màu sắc...) cho một tài liệu Web

Một số đặc điểm của Cascading Style Sheets:

- CSS quy định cách hiển thị của các thẻ HTML bằng cách quy định các thuộc tính của các thẻ đó (font chữ, màu sắc). Để cho thuận tiện bạn có thể đặt toàn bộ các thuộc tính của thẻ vào trong một file riêng có phần mở rộng là ".css"

- CSS nó phá vỡ giới hạn trong thiết kế Web, bởi chỉ cần một file CSS có thể cho phép bạn quản lí định dạng và layout trên nhiều trang khác nhau. Các nhà phát triển Web có thể định nghĩa sẵn thuộc tính của một số thẻ HTML nào đó và sau đó nó có thể dùng lại trên nhiều trang khác.

- Có thể khai báo CSS bằng nhiều cách khác nhau. Bạn có thể đặt đoạn CSS của bạn phía trong thẻ <head>...</head>, hoặc ghi nó ra file riêng với phần mở rộng ".css", ngoài ra bạn còn có thể đặt chúng trong từng thẻ HTML riêng biệt

- Tuy nhiên tùy từng cách đặt khác nhau mà độ ưu tiên của nó cũng khác nhau. Mức độ ưu tiên của CSS sẽ theo thứ tự sau.

Style đặt trong từng thẻ HTML riêng biệt

Style đặt trong phần <head>

Style đặt trong file mở rộng .css

Style mặc định của trình duyệt

Mức độ ưu tiên sẽ giảm dần từ trên xuống dưới.

CSS có tính kế thừa: giả sử rằng bạn có một thẻ đã được khai báo ở đầu file css với các thuộc tính như sau:

```
#vidu {
 width: 200px;
 height: 300px;
```

```
}
```

Ở một chỗ nào đó trong file css bạn lại khai báo một lần nữa thẻ với các thuộc tính.

```
#vidu {
width: 400px;
background-color: #CC0000;
}
```

Sau đoạn khai báo này thì thẻ sẽ có thuộc tính:

```
#vidu {
width: 400px; /* Đè lên khai báo cũ */
height: 300px;
background-color: #CC0000;
}
```

II. Cú pháp

1. Định dạng thuộc tính thẻ html

Các thuộc tính của thẻ html không được phong phú và đa dạng do vậy chúng ta có thể thay đổi lại thuộc tính mặc định của thẻ. Chúng ta có thể định dạng các thẻ ngay trên tài liệu html bằng thẻ <style> hoặc được định dạng trong một tập tin khác sau đó chèn vào tài liệu html.

Cú pháp:

```
name_tag
{
property_1: values;
property_2: values;
…
}
```

Ví dụ: Định dạng hình nền và các thuộc tính khác

```
body
{
background-image: url("images\background_image.gif");
background-repeat: no-repeat;
}
```

2. Định dạng một kiểu mới

a. Kiểu được chèn vào thẻ html bằng thuộc tính class của thẻ

Chúng ta có thể định dạng một kiểu riêng trong tài liệu html hoặc trong một tập tin riêng. Sau đó đưa định dạng đó vào trong thể html bằng thuộc tính class của thẻ.

Cú pháp:

```
.name_style
{
property_1: values;
property_2: values;
…
}
```

Ví dụ: Định dạng thuộc tính hình nền trang web

```
. mystyle
{
background-image: url(“images\background_image.gif”);
background-repeat: no-repeat;
}
```

Sau đó, ta áp dụng định dạng trên vào thẻ <body class= “mystyle”>

…

<body class= “mystyle”>

…

b. Kiểu được chèn vào thẻ html bằng thuộc tính id của thẻ

Chúng ta có thể định dạng một kiểu riêng trong tài liệu html hoặc trong một tập tin riêng. Sau đó đưa định dạng đó vào trong thể html bằng thuộc tính Id của thẻ.

Cú pháp:

```
# name_style
{
property_1: values;
property_2: values;
…
}
```

Ví dụ: Định dạng thuộc tính hình nền trang web

```
# mystyle
{
```

```
background-image: url(“images\background_image.gif”);
background-repeat: no-repeat;
}
```

Sau đó, ta áp dụng định dạng trên vào thẻ <body id= “mystyle”>

…

<body id= “mystyle”>

…

3. Định dạng ngay trong thẻ html

Ngoài các cách trên, chúng ta có thể định dạng kiểu trên thẻ html.

Cú pháp:

```
<name_tag  style= “property_1:values [; property_2: values; …] ”>
<!-- Nội dung thẻ -->
</name_tag>
```

Ví dụ:

```
<body style="background-image:url(anh_nen.JPG); background-repeat:no-repeat">
Nội dung tài liệu html
</body>
```

III. Sử dụng css trong tài liệu HTML

1. CSS được khai báo trong một tập tin riêng

Khi xây dựng website các trang web thường có những định dạng giống nhau tạo nên sự thống nhất của một website ví dụ như: Màu nền hay hình nền, các kiểu định dạng chữ, ký tự…

Nếu trang nào ta cũng sử dụng kiểu định dạng này, khi đó ta thấy mã lệnh trong một website được lặp đi lặp lại nhiều lần trên nhiều trang khác nhau.

Khi người dùng muốn thay đổi kiểu dáng hay định dạng của website người dùng chỉ cần thay đổi kiểu trong tập tin riêng này, khi đó toàn bộ trang website sẽ thay đổi theo.

Để chèn tập tin riêng chúng ta sử dụng thẻ <link> trong vùng thẻ <head>

```
<link href="name_style.css" rel="stylesheet" type="text/css" />
```

2. Định dạng ngay trên tài liệu html

Thường chúng ta thường định dạng css trong vùng thẻ head sử dụng thẻ <style>

Cú pháp:

```
<style>
<!-- Nội dung định dạng css -- >
</style>
```

IV. Một số thuộc tính thường dùng

1. Định kiểu nền

a. Màu nền

Để xác lập màu nền cho một thành phần của trang web ta sử dụng thuộc tính **background-color.** Các giá trị màu của background-color tương tự như color.

Ví dụ 4:

Mở file style.css ở trên thêm vào các thuộc tính màu nền như sau:

```
body { background-color:#000000;}
p{ background-color:#6495ed; color:#ffffff; }
h1 { background-color:#009fff; }
div {        background-color:#ffbf00; }
```

Kết quả hiển thị

b. Ảnh nền

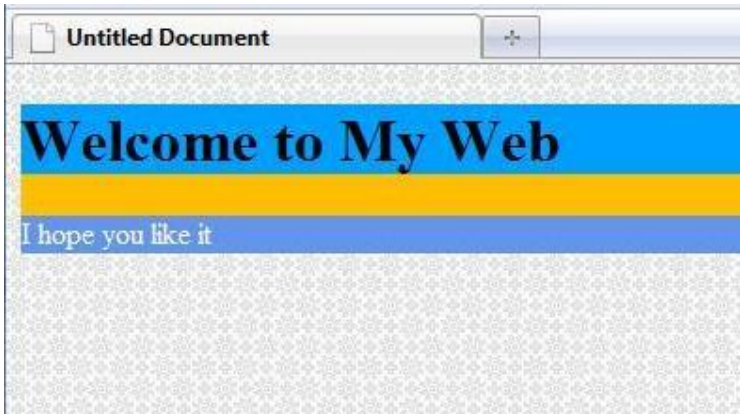
Để chèn ảnh nền vào một thành phần trên trang web chúng ta sử dụng thuộc tính background-image. Theo mặc định, ảnh nền sẽ được lặp lại để phủ kín toàn bộ trang web

Ví dụ 5

Mở file style.css ở trên và thay đổi nền cho trang web như sau

```
body {  background-image:url(page_bg.jpg); }
```

Lưu lại và xem kết quả



Lưu ý:

Để tiện cho sự quản lý các file trong website, ta nên tạo một thư mục images riêng để chứa hình ảnh. Khi đó, đường dẫn trong url như sau:

```
    background-image:url(images/page_bg.jpg);
```

Lặp lại ảnh nền:

Thuộc tính background-repeat cung cấp cho chúng ta các điều khiển giúp kiểm soát trình trạng lặp lại của ảnh nền. Thuộc tính này có 4 giá trị:

- repeat-x: Chỉ lặp lại ảnh theo phương ngang.
- repeat-y: Chỉ lặp lại ảnh theo phương dọc.
- repeat: Lặp lại ảnh theo cả 2 phương, đây là giá trị mặc định.
- no-repeat: Không lặp lại ảnh.

Ví dụ 6

Mở file style.css ở trên và sửa lại như sau

```
body {
background-image:url(page_bg.jpg);
background-repeat:no-repeat;
}
```

Lưu lại và xem kết quả.

Định vị ảnh nền

Theo mặc định, ảnh nền khi được chèn sẽ nằm ở góc trên, bên trái màn hình. Tuy nhiên với thuộc tính **background-position** ta có thể đặt ảnh nền ở bất cứ vị trí nào trong không gian của thành phần mà nó làm nền.

Ví dụ 7:

Mở file style.css và sửa lại như sau

```
body {
	background-image:url(page_bg.jpg);
	background-repeat:no-repeat;
	background-position:top right;
}
```

Mở trình duyệt và xem kết quả

2. Định kiểu chữ

a. Màu chữ

Sử dụng thuộc tính **color** để định dạng màu cho chữ trong CSS. Có nhiều cách để xác định giá trị của thuộc tính color

- Tên màu: red, blue, yellow,...
- Giá trị RGB: rgb(255,0,0)
- Giá trị HEX: #ff0000

Ví dụ 8:

```
body {color:blue}
h1 {color:#00ff00}
h2 {color:rgb(255,0,0)}
```

b. Canh lề:

Sử dụng thuộc tính **text-align** để canh chỉnh văn bản cho các thành phần trong trang web. Text-align có 4 giá trị :

- left (canh trái – mặc định)
- right (canh phải)
- center (canh giữa)
- justify (canh đều).

Ví dụ 9

```
h1 {  text-align:right  }
 p  {  text-align:justify  }
```

c. Trang trí chữ

Thuộc tính **text-decoration** dùng để thiết lập hay xóa các trang trí cho chữ

Text-decoration thường được dùng để xóa hiệu ứng gạch chân của link cho mục đích trang trí

Text-decoration thêm các hiệu ứng gạch chân (underline), gạch xiên (line-through), gạch đầu (overline), và một hiệu ứng đặc biệt là văn bản nhấp nháy (blink).

Ví dụ 10:

```
a {text-decoration:none}
h1 { text-decoration:underline }
h2 { text-decoration:overline}
```

d. Chuyển đổi chữ hoa/thường

Để chuyển đổi kiểu chữ hoa/thường ta dùng thuộc tính **text-transform**. Thuộc tính này có tất cả 4 giá trị:

- uppercase (in hoa)
- lowercase (in thường)
- capitalize (in hoa ký tự đầu tiên trong mỗi từ)
- none (không áp dụng hiệu ứng – mặc định).

Ví dụ 11

```
p { text-transform:uppercase }
h1 { text-transform:capitalize }
```

e. Thuộc tính letter-spacing:

Thuộc tính **letter-spacing** được dùng để định khoảng cách giữa các ký tự trong một đoạn văn bản.

Ví dụ 12

```
p { letter-spacing:3px }
h1{ letter-spacing:5px }
```

3. Định kiểu font

a. Tên font (font-family)

Thuộc tính font-family xác định các font sẽ được dùng để hiển thị trên trang web. Có hai loại tên font được dùng để chỉ định trong font-family:

- generic family
- font family

Generic family:

Generic family là tên của một họ gồm nhiều font. **Ví dụ:**

- **serif**
  - Times New Roman, Bodini, Garamond
- **sans-serif**
  - Trebuchet, Arial, Verdana, Futura, Gill Sans, Helvetica

- **cursive**
  - Poetica, Zapf-Chancery, Roundhand, Script
- **fantasy**
  - Critter, Cottonwood
- **monospace**
  - Courier, Courier New, Prestige, Everson Mono

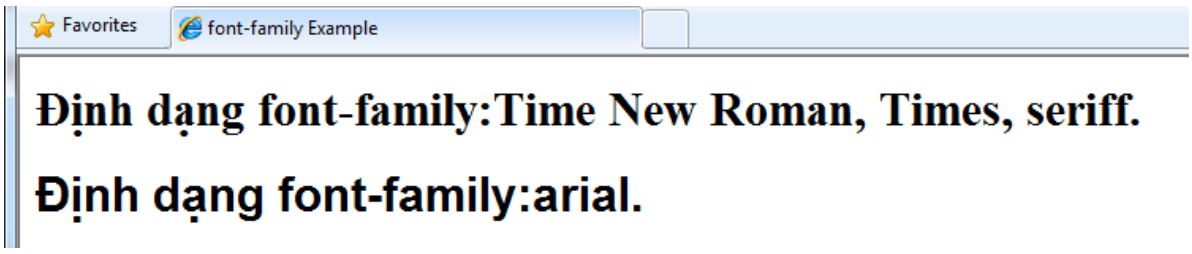
Font family:

Font family là tên cụ thể của một font. **Ví dụ:** Arial, Verdana, Time New Roman,…

Ví dụ 13

```
<html>
<head>
<title>font-family Example</title>
<style>
	body {font-size: 30px}
</style>
</head>
<body>
<p>
<strong style="font-family:'Times New Roman', Times, serif">Định dạng font-family:Time New Roman, Times, seriff..</strong>
</p>
<p>
<strong style="font-family: arial">Định dạng font-family:Time New Roman, Times, seriff..</strong>
</p>
</body>
</html>
```

Kết quả hiển thị



b. Kiểu font (font style)

Thuộc tính font style gồm 3 giá trị:

- Normal: in thườngg
- Italic: in nghiêng
- Oblique: tương tự như italic

Ví dụ 14

```
h1 { font-style: italic }
p { font-style: normal }
```

c. Cỡ font (font size)

Kích thước của một font được định bởi thuộc tính **font-size**. Font-size cung cấp 7 giá trị cho việc thiết lập size của font từ nhỏ nhất cho đến lớn nhất: xx-small, x-small, small, medium, large, x-large and xx-large. Chúng tương đương với giá trị của thẻ <font> với size="1" tới size="7".

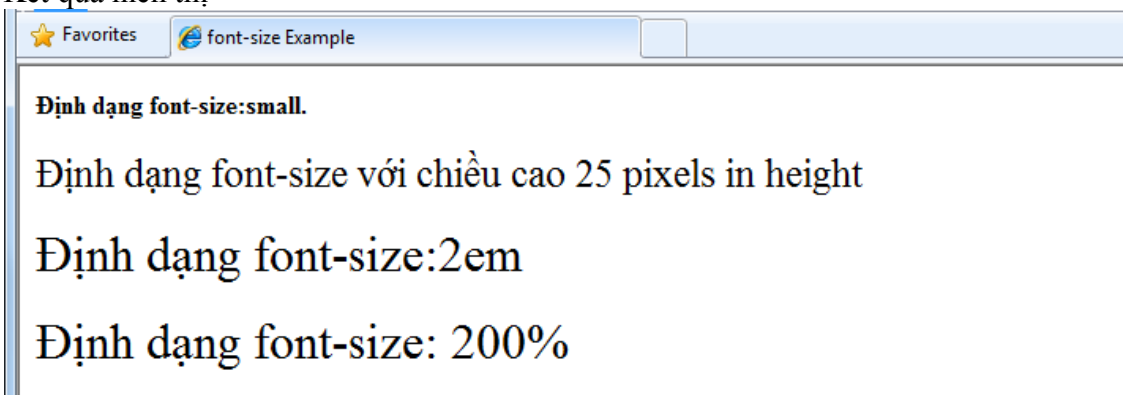
Ngoài ra các đơn vị dùng cho font thường là: pixel, em, %

Tùy theo mục đích sử dụng của website mà ta chọn những đơn vị phù hợp.

Ví dụ 15

```
<html>
<head>
<title>font-size Example</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
</head>
<body>
<p><b style="font-size: small">Định dạng font-size:small.</b></p>
<p style="font-size: 25px">Định dạng font-size với chiều cao 25 pixels in height</p>
<p style="font-size:2em">Định dạng font-size:2em </p>
<p style="font-size: 200%">Định dạng font-size: 200%</p>
</body>
</html>
```

Kết quả hiển thị

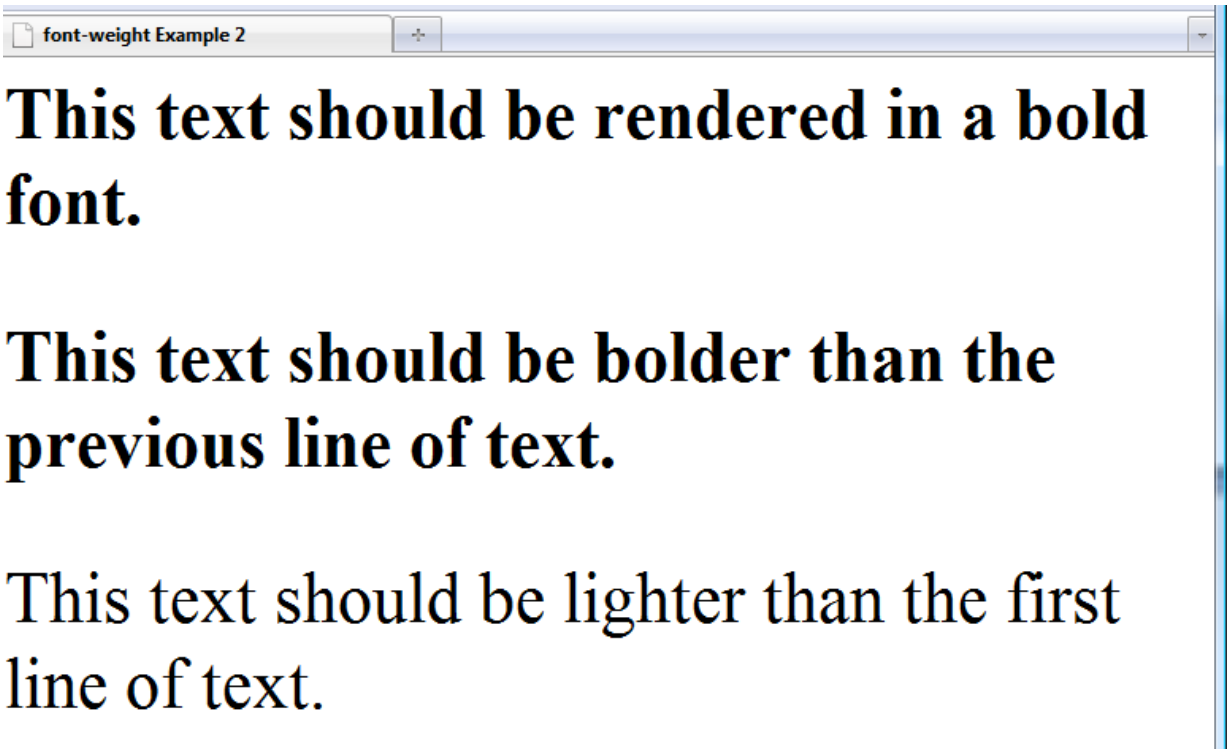


d. Thuộc tính font-weight:

Thuộc tính **font-weight** mô tả cách thức thể hiện của font chữ là ở dạng bình thường (normal) hay in đậm (bold). Ngoài ra, một số trình duyệt cũng hỗ trợ mô tả độ in đậm bằng các con số từ 100 – 900.

Ví dụ 16

```
<html>
<head>
<title>font-weight Example 2</title>
	<style>
		body {font-weight: 600; font-size: 45px}
	</style>
</head>
<body>
<p>This text should be rendered in a bold font.</p>
<p style="font-weight: bolder">This text should be bolder than the previous line of text.</p>
<p style="font-weight: lighter">This text should be lighter than
the first line of text.</p>
</body>
</html>
```



4. CSS Link

Trong CSS, ta có thể áp dụng các thuộc tính định dạng như font chữ, gạch chân, màu chữ,… cho một liên kết. Ngoài ra, CSS còn cung cấp một số hiệu ứng định dạng cho một đối tượng liên kết ở một trạng thái xác định như :

- Liên kết chưa được thăm (a:link)
- Rê chuột lên liên kết (a:hover)
- Liên kết được thăm (a:visited)
- Liên kết đang được kích hoạt – đang giữ nhấn chuột (a:active)

Ví dụ 17

```
<html>
<head>
<style type="text/css">
      a:link {color:#00FF00; font-size:14px; text-decoration:none;}
      a:hover { color:#FF00FF; font-size:1.2em; text-decoration:blink;}
      a:visited { color:#FF0000; text-decoration:none; background-color:#ff704d; }
      a:active { color:# 662D91; font-variant:small-caps; }
</style>
</head>
<body>
        <p><b><a href="default.asp" target="_blank">This is a link</a></b></p>
</body>
</html>
```

5. Định kiểu danh sách

CSS cũng cung cấp một số thuộc tính để định dạng danh sách, làm việc trình bày trang web trở nên phong phú hơn. Các thuộc tính về danh sách của CSS cho phép ta:

- Định dạng kí hiệu cho danh sách có thứ tự
- Định dạng kí hiệu cho danh sách không thứ tự
- Dùng hình ảnh đánh dấu danh sách

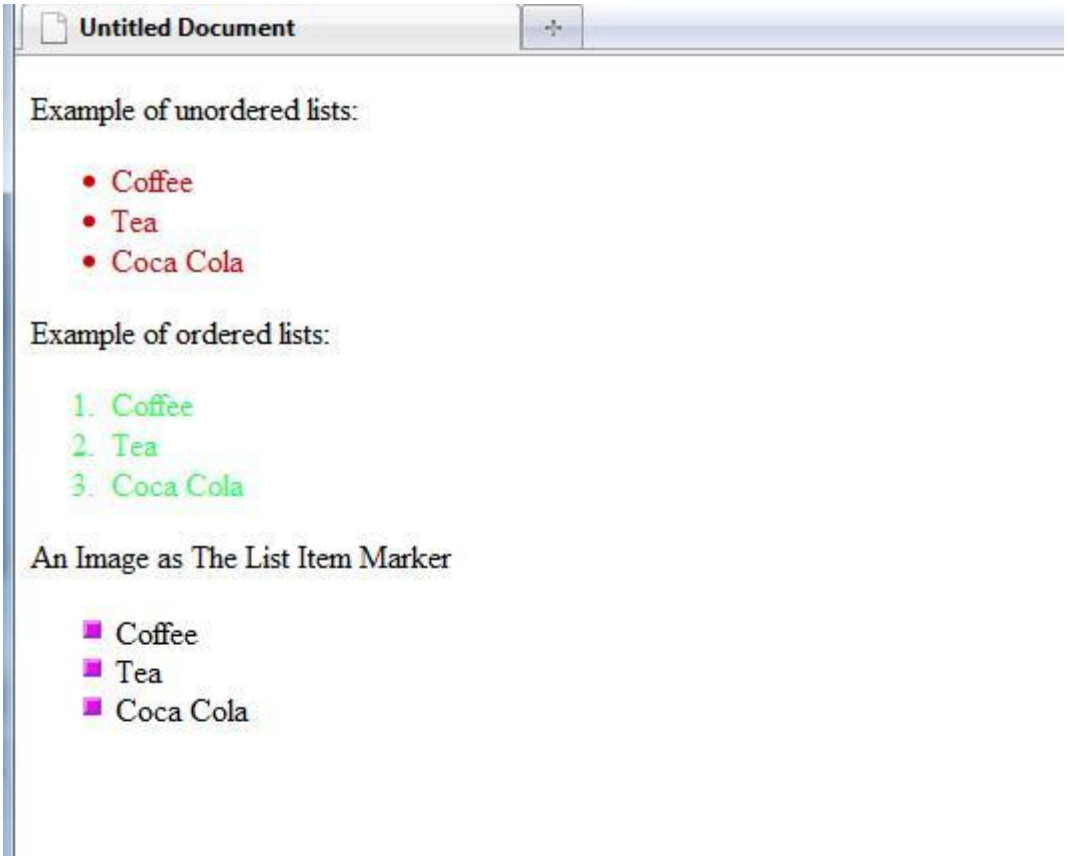
Thuộc tính **list-style-type** cho phép ta định dạng các kí hiệu đầu danh sách

Ví dụ 18

```
<html>
<head>
      <style type="text/css">
            ul{color:#D40000;}
            ol{color:#2AFF55;}
      </style>
```

```
</head>
<body>
<p>Example of unordered lists:</p>
<ul list-style-type:circle; >
     <li>Coffee</li>
     <li>Tea</li>
     <li>Coca Cola</li>
 </ul>
<p>Example of ordered lists:</p>
<ol list-style-type: upper-roman>
     <li>Coffee</li>
     <li>Tea</li>
     <li>Coca Cola</li>
   </ol>
<p>Example of ordered lists:</p>
<ul list-style-image:url('bullet.gif') >
     <li>Coffee</li>
     <li>Tea</li>
     <li>Coca Cola</li>
   </ul>
</body>
</html>
```

Kết quả



6. Định kiểu bảng

Với CSS, ta có thể thay đổi định dạng các thuộc tính của bảng như đường biên, độ rộng, độ cao của cột, màu,…

Các thuộc tính CSS dùng làm việc với bảng :

a. Border:

Dùng để định độ rộng đường viền của bảng

Ví dụ 19

table, th, td { border: 1px solid black; }

Border-collapse: có 3 giá trị

- Collapse: cho phép ta trộn các đường biên trong bảng
- Separate: cho phép ta tách các đường biên trong bảng
- Inherit:tương tự separate

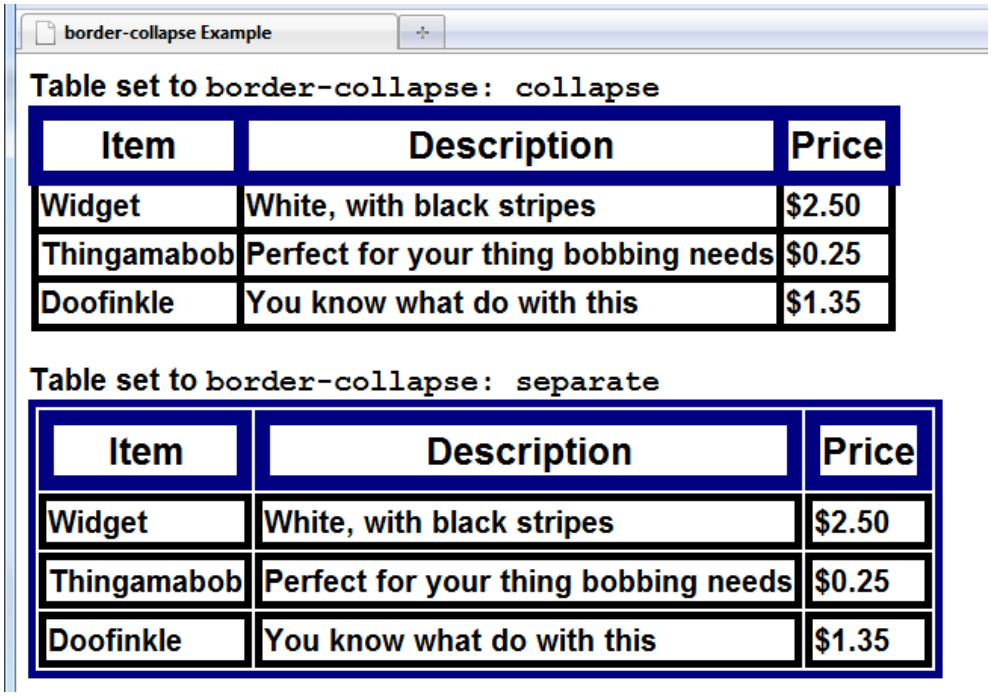
Ví dụ 20

```
<html>
<head>
<title>border-collapse Example</title>
```

```
<style>
	th {border: 10px solid navy; font: bold 25px Arial, Helvetica,
	sans-serif}
	td {border: 5px solid black; font: bold 20px Arial, Helvetica,
	sans-serif}
	body {font: bold 20px Arial, Helvetica, sans-serif}
</style>
</head>
<body>
	Table set to <code>border-collapse: collapse</code>
	<table style="border-collapse: collapse; border: 5px solid navy">
		<tr>
<th>Item</th>
<th>Description</th>
<th>Price</td>
		</tr>
		<tr>
<td>Widget</td>
<td>White, with black stripes</td>
<td>$2.50</td>
		</tr>
		<tr>
<td >Thingamabob</td>
<td >Perfect for your thing bobbing needs</td>
<td >$0.25</td>
		</tr>
		<tr>
<td>Doofinkle</td>
<td>You know what do with this</td>
<td>$1.35</td>
		</tr>
	</table>
	<p>
		Table set to <code>border-collapse: separate</code>
	<table style="border-collapse: separate; border: 5px solid navy">
		<tr>
<th>Item</th>
<th>Description</th>
<th>Price</td>
		</tr>
		<tr>
<td>Widget</td>
<td>White, with black stripes</td>
<td>$2.50</td>
		</tr>
```

```
            <tr>
<td >Thingamabob</td>
<td >Perfect for your thing bobbing needs</td>
<td >$0.25</td>
            </tr>
            <tr>
<td>Doofinkle</td>
<td>You know what do with this</td>
<td>$1.35</td>
            </tr>
        </table>
        </p>
</body>
</html>
```

Kết quả hiển thị



b. Width:

Định độ rộng của bảng, cột, ô

c. Height:

Định độ cao của bảng, cột, ô

Ví dụ 21

```
table { width:100%;}
th { height:50px; }
```

d. Text-align:

Căn lề cho text theo phương ngang trong ô của bảng gồm các giá trị như left, right, center

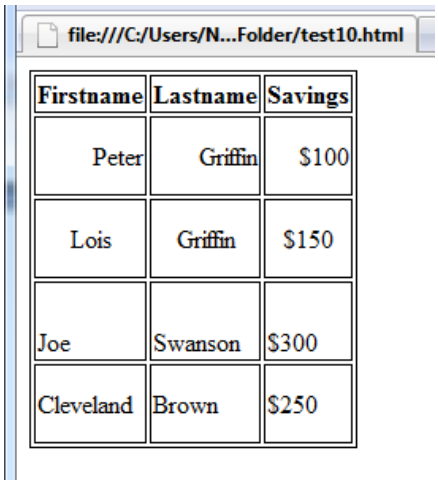
e. Vertical-align:

Căn lề cho text theo phương đứng trong ô của bảng gồm các giá trị như top, bottom, middle

Ví dụ 22:

```
<html >
<head>
<style type="text/css">
table, td, th { border:1px solid black;}
td{ height:50px;}
</style>
</head>
<body>
<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Savings</th>
</tr>
<tr style="text-align:right;">
<td>Peter</td>
<td>Griffin</td>
<td>$100</td>
</tr>
<tr style="text-align:center">
<td>Lois</td>
<td>Griffin</td>
<td>$150</td>
</tr>
<tr style="vertical-align:bottom;">
<td>Joe</td>
<td>Swanson</td>
<td>$300</td>
</tr>
<tr style="vertical-align:middle;">
<td>Cleveland</td>
<td>Brown</td>
<td>$250</td>
</tr>
</table>
</body>
</html>
```

Kết quả hiển thị



f. Padding:

Xác định khoảng cách giữa text và đường biên của ô trong bảng

Ví dụ 23:

```
td{ padding:15px; }
```

g. Background-color:

Màu nền ô

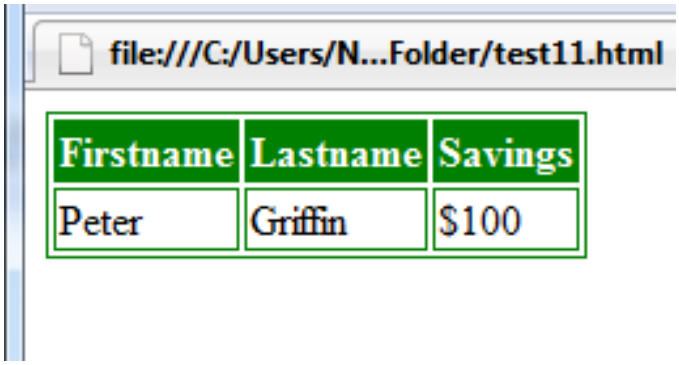
h. Color:

Màu chữ

Ví dụ 24:

```
<html>
<head>
<style type="text/css">
        table, td, th { border:1px solid green; }
        th { background-color:green; color:white; }
</style>
</head>
<body>
<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Savings</th>
</tr>
<tr>
<td>Peter</td>
<td>Griffin</td>
<td>$100</td>
```

```
</tr>
</table>
</body>
</html>
```



7. Thuộc tính Id và class của thẻ

Khi áp dụng một thuộc tính CSS cho một thành phần như p, a, img,… thì toàn bộ các thành phần này trong trang web đều nhận thuộc tính này. Nếu ta muốn một thành phần nào đó như liên kết trên menu có các thuộc tính khác với liên kết trong phần nội dung thì ta sẽ nhóm các thuộc tính đó vào trong id hoặc class

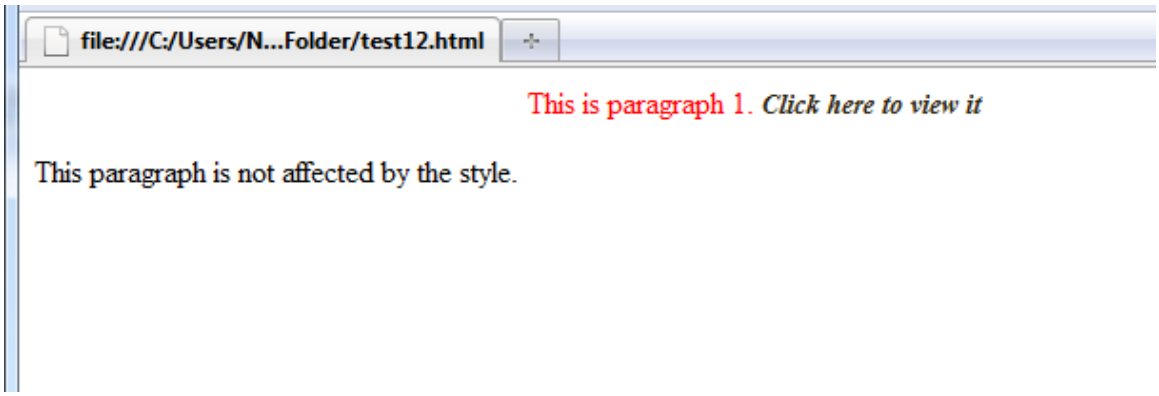
a. Thuộc tính Id

Id được dùng cho một đối tượng riêng biệt, và được xác định bởi ký hiệu "#".

Ví dụ 25

```
<html>
<head>
<style>
#para1
{
	text-align:center;
	color:red;
}
#para1 a
{
	text-decoration:none;
	color:#2e260f;
	font-weight:bold;
	font-style:italic;
	font-size:14px;
}
</style>
</head>
<body>
	<p id="para1">This is paragraph 1. <a href="id.html">Click here to view it</a></p>
```

```
    <p>This paragraph is not affected by the style.</p>
</body>
</html>
```



b. Thuộc tính Class

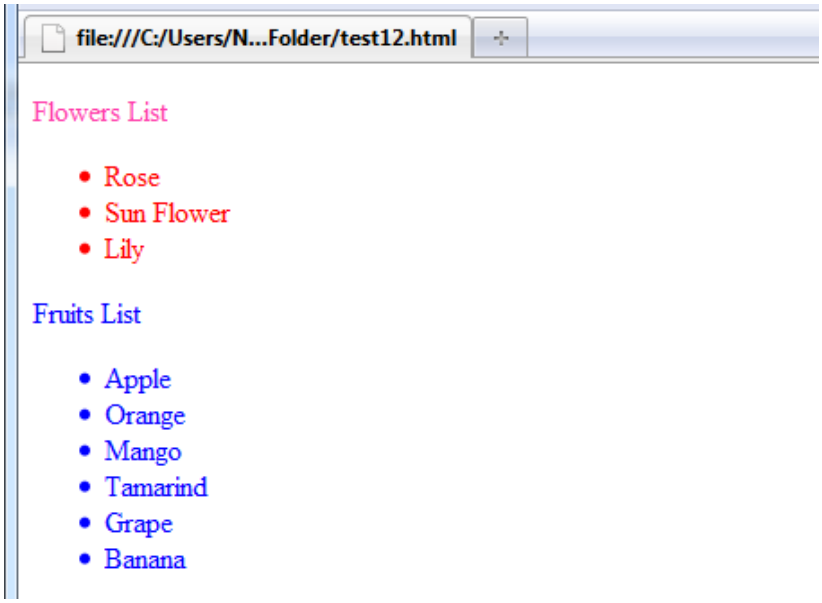
Class được dùng để nhóm một số thành phần có những thuộc tính đặc biệt. Nhưng khác với id, class được sử dụng cho nhiều đối tượng khác nhau và được xác định bởi ký hiệu "."

Ví dụ sau sẽ cho thấy rõ hơn sự khác biệt giữa id và class

Ví dụ 26

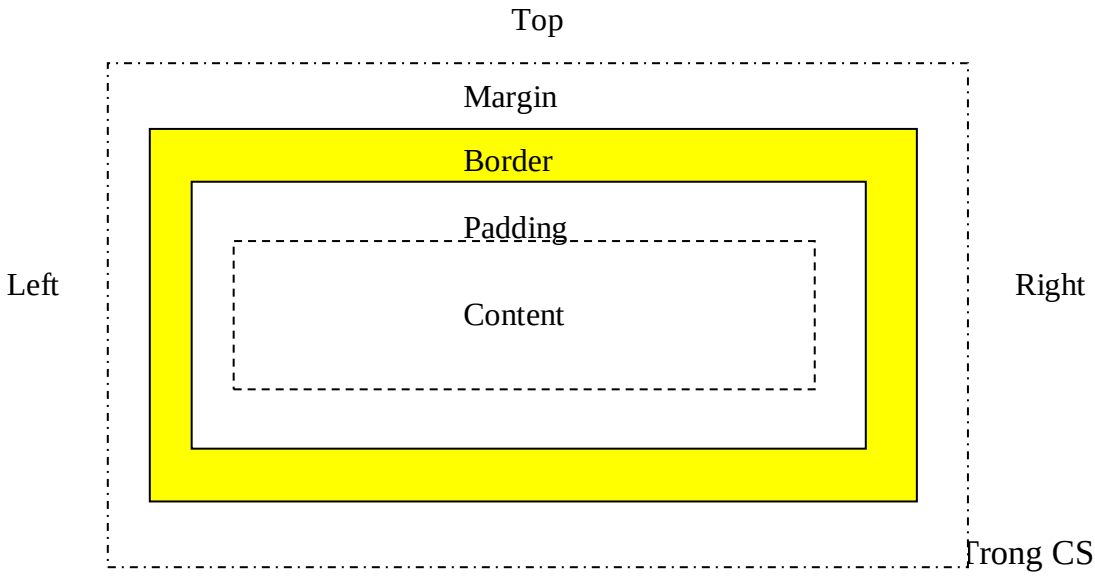
```
<html>
<head>
<style>
    #flower {color:#ff0000;}
    #flower p{ color:#ff3faa;}
    .fruit { color:#0000FF }
</style>
</head>
<body>
<div id="flower">
    <p>Flowers List</p>
    <ul>
    <li>Rose</li>
    <li>Sun Flower</li>
    <li>Lily</li>
  </ul>
</div>

<p class="fruit">Fruits List</p>
<div class="fruit">
<ul>
    <li >Apple</li>
    <li >Orange</li>
    <li >Mango</li>
```

```
        <li >Tamarind</li>
        <li >Grape</li>
        <li >Banana</li>
</ul>
</div>
</body>
</html>
```



* Lưu ý: không nên dùng kí tự đầu là chữ số đặt tên cho class và id

8. Mô hình hộp



Trong CSS, box model (mô hình hộp) mô tả cách mà CSS định dạng khối không gian bao quanh một thành phần. Nó bao gồm padding (vùng đệm), border (viền) và margin (canh lề) và các tùy chọn. Hình bên dưới mô tả cấu trúc minh họa mô hình hộp cho một thành phần web.

a. Thuộc tính margin:

Thuộc tính margin được dùng để canh lề cho một thành phần web hay cả trang web so với các đối tượng bên ngoài. Các thuộc tính về margin gồm:

- margin-top
- margin-right
- margin-bottom
- margin-left

Các đơn vị được dùng với thuộc tính margin:

- Auto: trình duyệt tự động thiết lập margin. Kết quả được thể hiện tùy thuộc vào trình duyệt
- Pixel, pt, em,…: thiết lập khoảng cách lề theo độ dài
- %: thiết lập khoảng cách lề theo phần trăm

Ví dụ 27

```
Body
{
	margin-top:80px;
	margin-bottom:40px;
	margin-left:50px;
	margin-right:30px;
	border:1px dotted #FF0000
}
Hoặc:
	body { margin:80px 30px 40px 50px; border:1px dotted #FF0000 }
```

b. Thuộc tính padding

Thuộc tính padding dùng để tạo khoảng cách giữa các nội dung bên trong với đường biên của khối. Các thuộc tính về padding gồm:

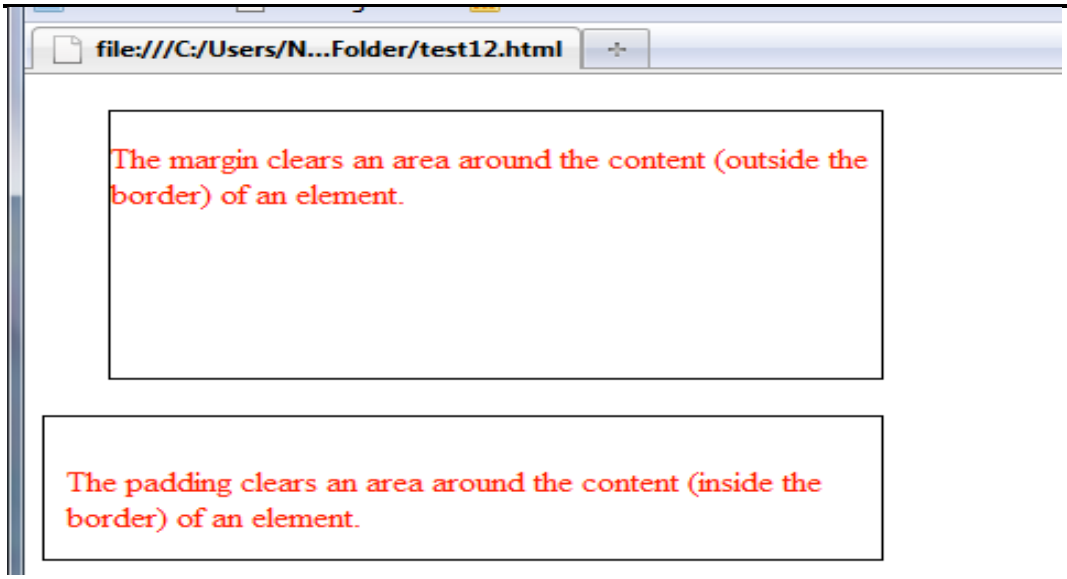
- padding-top
- padding-right
- padding-bottom
- padding-left.

Các đơn vị được dùng với thuộc tính padding: %, px, pt, em,…

Ví dụ 28

```
<html>
```

```
<head>
<style type="text/css">
	.padding
	{
		border:1px solid #000000;
		padding: 10px 20px 20px 10px;
		color:#FF1F00;
		height:50px;
		width:350px;
	}
	#margin
	{
		border:1px solid #000000;
		margin-top: 20px;
		margin-right:30px;
		margin-left:30px;
		margin-bottom:20px;
		color:#FF1F00;
		height:150px;
		width:350px;
	}
</style>
</head>
<body>
	<div id="margin">
		<p>The margin clears an area around the content (outside the border) of an element.</p>
	</div>
	<div class="padding">
		<p>The padding clears an area around the content (inside the border) of an element.</p>
	</div>
</body>
</html>
```



c. Border

Trong ví dụ trên ta thấy thuộc tính border được dùng để đóng khung, trang trí cho một đối tượng, phân cách các đối tượng giúp trang web trông dễ nhìn hơn. Border có các thuộc tính sau:

- **border-width:** độ rộng cho viền, có các giá trị thin (mảnh), medium (vừa), thick (dày), hay là một giá trị đo cụ thể như pixels
- **border-color:** màu viền
- **border-style:** kiểu viền, có 8 giá trị dotted, dashed, solid, double, groove, ridge, inset và outset. Ngoài ra, hai giá trị none hay hidden dùng để ẩn đường viền

Chúng ta cũng có thể dùng riêng các thuộc tính border-top, border-right, border-bottom hay border-left để chỉ định viền riêng cho các đối tượng.

Ví dụ 29

```
#border
{
border-top-width:thin;
border-top-color:#FF0000;
border-top-style:solid;
border-right-width:thick;
border-right-color:#AFAFAF;
border-right-style:dotted;
}
```

d. Thuộc tính Width và Height

Thuộc tính Width

Width quy định độ rộng cho một thành phần web, ngoài ra ta còn có một số thuộc tính đi kèm

- max-width: quy định chiều rộng tối đa cho một thành phần web.
- min-width: quy định chiều rộng tối thiểu cho một thành phần web.

Thuộc tính height:

Height: quy định chiều cao cho một thành phần web, ngoài ra ta còn có một số thuộc tính đi kèm

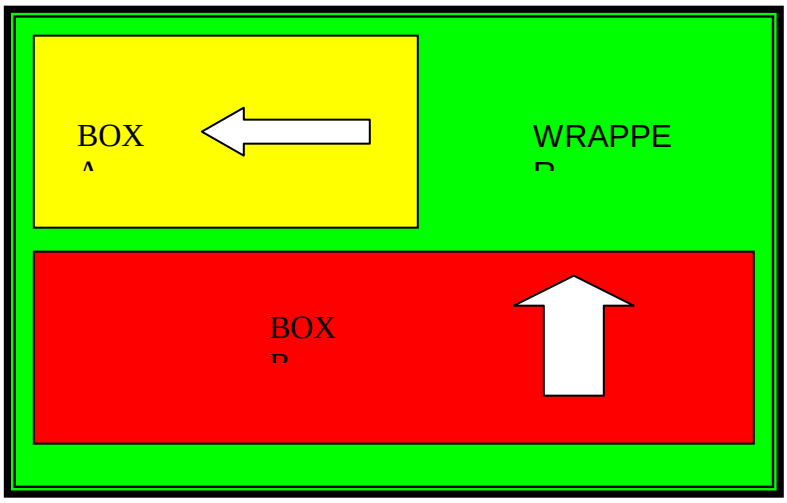
- max-height: quy định chiều cao tối đa cho một thành phần web.
- min-height: quy định chiều cao tối thiểu cho một thành phần web.

e. Thuộc tính float và clear

Thuộc tính Float:

Thuộc tính float dùng để cố định một thành phần web về bên trái hay bên phải không gian bao quanh nó. Đây là một thuộc tính rất cần thiết khi dàn trang, hiển thị văn bản thành cột, định vị trí ảnh và text

Trong hình minh họa dưới đây ta thấy 2 khối BOX A và BOX B được đặt trong khối WRAPPER. Khi ta sử dụng thuộc tính float cho BOX A cố định về phía trái thì BOX B sẽ tràn lên để lấp khoảng trống phía trên



Thuộc tính float có 3 giá trị:

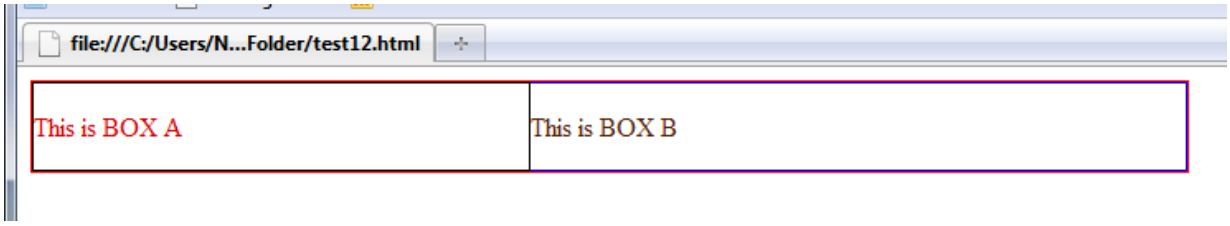
- Left: Cố định phần tử về bên trái.
- Right: Cố định phần tử về bên phải.
- None: Bình thường.

Ví dụ 30

```
<html>
<head>
<style type="text/css">
```

```
	#wrapper
	{
		border:1px solid #ff0000;
		width:700px;
	}
	#box_a
	{
		border: 1px solid #000000;
		width: 300px;
		float:left;
		color:#d40000;
	}
	#box_b
	{
		border: 1px solid #0000ff;
		color:#551f00;
	}
</style>
</head>

<body>
	<div id="wrapper">
		<div id="box_a">
			<p>This is BOX A</p>
		</div>
		<div id="box_b">
				<p>This is BOX B</p>
		</div>
	</div>
</body>
</html>
```

Kết quả hiển thị



Thuộc tính clear

Thuộc tính clear thường đi kèm với float, được dùng xử lý các phần tử liên quan tới phần tử đã được float để quyết định hướng xử sự của phần tử này. Ở ví dụ trên, khi BOX A được float qua trái thì mặc nhiên BOX B sẽ được tràn lên để lắp vào chỗ trống. Nhưng khi chúng

ta sử dụng thuộc tính clear cho BOX B thì chúng ta có quyền quyết định xem phần văn bản đó có được tràn lên hay không.

Thuộc tính clear có 4 giá trị: left, right, both và none.

Thuộc tính position

Bên cạnh thuộc tính float, clear, CSS cung cấp cho ta thuộc tính position để xác định tọa độ của một đối tượng nào đó trên cửa sổ trình duyệt. Ta có 2 cách để xác định tọa độ:

- Định vị tuyệt đối: position sẽ nhận giá trị absolute
- Định vị tương đối: position nhận giá trị relative

Ví dụ 31

```
#img1{position:absolute; top:50px; left:70px }
#img2 { position:relative; bottom:70px; right:50px }
```

Chương 4: Giới thiệu ngôn ngữ kịch bản Javascript

I. Giới thiệu về Javascript

- Javascript là ngôn ngữ kịch bản được sử dụng nhiều trên các website, và được hỗ trợ trên một số trình duyệt như: Internet Explorer, FireFox, Chrome, Opera, và Safari.

- Để học ngôn ngữ này bạn cần hiểu cơ bản về HTML hoặc XHTML.

Vậy javascript là gì?

- javascript là ngôn ngữ kịch bản được nhúng vào trong tài liệu html.

II. Ngôn ngữ javascript

1. Chèn mã lệnh javascript vào trong tài liệu HTML

a. Chèn mã lệnh trên vùng <body>

Cách chèn mã lệnh nay chỉ áp dụng khi mã lệnh javacript được chèn thực hiện một mục đích nào đó trên tài liệu html tại vị trí cần chèn vào.

Ví dụ 4.1: Chèn mã lệnh javascript vào trong tài liệu html

```
<body>
<script language="javascript" type="text/javascript">
// code here.
</script>
</body>
```

b. Chèn mã lệnh trên vùng <head>

Cách chèn mã lệnh nay thường khi mã lệnh javacript được chèn thực hiện một mục đích nào tại nhiều vị trí khác nhau trên tài liệu html.

Ví dụ 4.2: Chèn mã lệnh javascript vào trong vùng thẻ head

```
<head>
<script language="javascript" type="text/javascript">
// code here.
</script>
</head>
```

c. Chèn mã lệnh trực tiếp vào trong các thẻ HTML

Cách chèn mã lệnh nay chỉ áp dụng khi mã lệnh javacript được chèn thực hiện một mục đích nào đó trên thẻ html được chèn vào.

Ví dụ 4.3: Chèn mã lệnh javascript vào trong thẻ html

```
<p onclick="alert('Xin chao cac ban');"> Click here!!!</p>
```

d. Chèn mã lệnh bằng một tập tin riêng trên vùng <head>

Cách chèn mã lệnh nay thường khi mã lệnh javacript được chèn thực hiện một mục đích nào tại nhiều vị trí khác nhau trên tài liệu html, bên cạnh đó mã lệnh trong tập tin này không chỉ áp dụng cho một trang bất kỳ mà có thể áp dụng cho toàn bộ website.

Ví dụ 4.4: Chèn mã lệnh bằng một tập tin riêng có tên my_javascript.js

```
<head>
<script language="javascript" src="my_javascript.js" type="text/javascript">
// code here.
</script>
</head>
```

2. Lời chú thích

Chúng ta có thể thêm những khối ghi chú để biết phần mã lệnh tương ứng thực hiện điều gì. Các ghi chú được trình duyệt bỏ qua và chỉ thấy trong mã nguồn.

Cú pháp câu ghi chú:

- Dòng ghi chú nằm trên một dòng văn bản.

```
// dòng ghi chú trên 1 dòng.
```

- Dòng ghi chú nằm trên nhiều dòng văn bản.

```
/* dòng ghi chú thứ nhất
 dòng ghi chú khác….  */
```

3. Biến và cách xuất thông tin lên trình duyệt

a. Biến và cách khai báo biến

Biến trong javascript được sử dụng từ khóa var để khai báo, khi khai báo nhiều biến chúng ta phân cách chúng bởi dấu (,).

Tên biến không có khoảng cách trống, không được trùng tên với từ khóa, bắt đầu bằng ký tự, đặc điểm tên biến phân biệt chữ hoa và chữ thường.

Ví dụ 4.5:

```
var so_a, so_b;
var chuoi;
```

Sau khi khai báo biến chúng ta có thể khởi gán giá trị cho tên biến bằng cách dùng toán tử gán(=), khi đó kiểu dữ liệu của biến là kiểu của giá trị được khởi gán.

Ví dụ 4.6:

so_a = 9; //kiểu của so_a là kiểu số nguyên.

chuoi = “Hello” // kiểu của chuoi là kiểu chuỗi ký tự.

Để kiểm tra xem kiểu dữ liệu của tên biến ta sử dụng hàm typeof(<tên_biến>).

Chuyển đổi chuỗi số thành kiểu số ta sử dụng hàm parseInt() hoặc parseFloat().

Ví dụ 4.7:

```
chuoi = “123.45”;
so_a = parseInt(chuoi); //khi đó so_a = 123;
so_b = parseFloat(chuoi); //khi đó so_b =123.45;
```

Chuyển đổi số thành chuỗi ta sử dụng hàm toString()

Ví dụ 4.8:

```
chuoi = toString(so_a); // khi đó chuoi = “123”;
```

b. Xuất thông tin lên trình duyệt web

Để xuất thông tin lên trình duyệt web ta sử dụng Cú pháp sau:

document.write(<nội dung>);

hoặc document.writeln(<nội dung>);

Ví dụ 4.9:

```
document.write(“Chào các bạn”); // xuất ra chuỗi.
document.write(chuoi); // xuất ra biến chuỗi.
```

4. Các phép toán

Trong ngôn ngữ lập trình javascript các phép toán tương tự như các phép toán trong C.

Các phép tính toán học:

+	Phép cộng
-	Phép trừ
*	Phép nhân
/	Phép chia
%	Phép lấy phần dư
++	Tăng giá trị lên 1 đơn vị
--	Giảm giá trị xuống 1 đơn vị

Các phép gán:

Phép toán	Ví dụ	Tương tự	Kết quả
=	x=y		x=5

+=	x+=y	x=x+y	x=15
-=	x-=y	x=x-y	x=5
*=	x*=y	x=x*y	x=50
/=	x/=y	x=x/y	x=2
%=	x%=y	x=x%y	x=0

Phép toán + : được sử dụng để cộng chuỗi(nối chuỗi)

Ví dụ 4.10: Cộng hai chuỗi

```
st1="Hôm nay là";
st2="đẹp  trời";
st=st1+ “  ”+st2; // Kết quả st = “Hôm nay là một ngày đẹp trời”;
```

Phép cộng giữa chuỗi và số: khi cộng một chuỗi với một số kết quả sẽ là chuỗi.

Ví dụ 4.11: Cộng chuỗi và số

```
//Cộng số với số
x=5+5;
document.write(x);
// Cộng chuỗi với chuỗi
x="5"+"5";
document.write(x);
// Cộng số với chuỗi
x=5+"5";
document.write(x);
// Cộng chuỗi với số
x="5"+5;
document.write(x);
```

Phép so sánh:

==	Bằng
===	Bằng chính xác (giá trị và kiểu)
!=	Không bằng
>	Lớn hơn
<	Nhỏ hơn
>=	Lớn hơn hoặc bằng
<=	Nhỏ hơn hoặc bằng

Phép toán logic:

&&	Và
||	Hoặc
!	Phủ định

Phép toán điều kiện:

ten_bien =(<điều kiện>)?gia_tri_true:gia_tri_sai

5. Câu lệnh rẽ nhánh If...Else

Câu lệnh rẽ nhánh có 3 dạng:

Dạng 1: Câu lệnh if dạng khuyết.

Câu lệnh if dạng này được sử dụng để kiểm tra điều kiện, nếu điều kiện thõa mãn thì thực hiện một nhiệm vụ nào đó.

Cú pháp:

```
if (<Điều kiện>)
 {
   // mã lệnh nếu biểu thức điều kiện đúng
  }
```

Ví dụ 4.12: Lấy ra giờ hệ thống, nếu giờ nhỏ hơn 10 giờ thì in ra trình duyệt "Chào buổi sáng"

```
<script type="text/javascript">
var d=new Date(); // Lớp d là kiểu dữ liệu giờ
var time=d.getHours(); // lấy ra giờ
if (time<10)
 {
  document.write("<b>Chào buổi sáng</b>");
 }
</script>
```

Dạng 2: Câu lệnh if dạng đầy đủ

Câu lệnh if dạng này được sử dụng để kiểm tra điều kiện, nếu điều kiện thõa màn thì thực hiện một công việc này còn nếu sai thì thực hiện nhiệm vụ khác.

Cú pháp:

```
if (<Điều kiện>)
  {
```

```
 // Mã lệnh nếu điều kiện đúng;
 }
else
 {
 //Mã lệnh nếu điều kiện sai
 }
```

Ví dụ 4.13: Lấy ra giờ hệ thống, nếu giờ hiện tại nhỏ hơn 10 thì xuất ra màn hình là "Good morning!", ngược lại "Good day!"

```
<script type="text/javascript">
var d = new Date();
var time = d.getHours();
if (time < 10)
 {
 document.write("Good morning!");
 } else {
  document.write("Good day!");
 }
</script>
```

Dạng 3: Câu lệnh if lồng nhau:

Câu lệnh if dạng lồng được sử dụng khi điều kiện đưa ra có thể xẩy ra hơn hai trường hợp.

Cú pháp:

```
if (<Điều kiện1>)
 {
  // Mã lệnh nếu biểu thức <Điều kiện 1> đúng;
 } else if (<Điều kiện2>)
 {
  // Mã lệnh nếu biểu thức <Điều kiện 2> đúng;
 } else
 {
  // Mã lệnh nếu biểu thức <điều kiện 1> và <điều kiện 2> sai.
  }
```

Ví dụ 4.14:

```
<script type="text/javascript">
var d = new Date()
var time = d.getHours()
if (time<10)
```

```
 {
 document.write("<b>Good morning</b>");
 }
else if (time>10 && time<16)
 {
 document.write("<b>Good day</b>");
 }
else
 {
 document.write("<b>Hello World!</b>");
 }
</script>
```

6. Câu lệnh lựa chọn Switch

Câu lệnh switch được sử dụng khi biểu thức có thể trả về nhiều giá trị khác nhau, mỗi giá trị như vậy thì thực hiện một công việc khác nhau.

Cú pháp:

```
switch(<biểu_thức>)
{
case gt1:
 //Thực hiện lệnh nếu biểu_thức = gt1
 break;
case gt2:
 //Thực hiện lệnh nếu biểu_thức = gt2
 break;
default:
 //Thực hiện lệnh nếu biểu_thức không bằng gt1 hoặc gt2
}
```

Ví dụ 4.15:

```
<script type="text/javascript">
var d=new Date();
theDay=d.getDay();
switch (theDay)
{
case 1:  document.write("Thứ hai");
 break;
case 2:  document.write("Thứ ba");
```

```
 break;
case 3:  document.write("Thứ tư");
 break;
case 4:  document.write("Thứ năm");
  break;
case 5:  document.write("Thứ sáu");
 break;
case 6:  document.write("Thứ bảy");
 break;
default:
  document.write("Chủ nhật");
}
</script>
```

7. Định nghĩa hàm

Ngoài những hàm javascript đã định nghĩa sẵn, chúng ta có thể định nghĩa để thực hiện một nhiệm vụ nào đó.

Cấu trúc của hàm:

```
function <tên hàm>(<tham số nếu có>)
{
	// mã lệnh.
}
```

8. Hộp thông báo

Khi người dùng nhập thông tin hoặc tác động đến trang web thì một thông báo hiện ra cảnh báo hay nhắc nhở người.

Kiểu thông báo: Hộp thông báo alert() chỉ có nút lệnh OK.

Cú pháp:

```
alert("Dòng thông báo");
```

Ví dụ 1.16:

```
<html>
<head>
<script type="text/javascript">
function show_alert()
{
alert("I am an alert box!");
```

```
}
</script>
</head>
<body>
<input type="button" onclick="show_alert()" value="Show alert box" />
</body>
</html>
```

Kiểu thông báo confirm() có hai nút lệnh nút lệnh OK và Cancel.

Nếu Click vào Ok thì kết quả trả về là true, nếu click vào Cancel thì kết quả trả về là false.

Cú pháp:

```
confirm("Dòng thông báo");
```

Ví dụ 4.17:

```
<html>
<head>
<script type="text/javascript">
function show_confirm()
{
var r=confirm("Press a button");
if (r==true)
 {
  document.write("You pressed OK!");
  } else   {
  document.write("You pressed Cancel!");
  }
}
</script>
</head>
<body>
<input type="button" onclick="show_confirm()" value="Click here" />
</body>
</html>
```

Kiểu thông báo prompt() cho phép người dùng nhập vào giá trị và chỉ có hai nút lệnh OK và Cancel.

Nếu click vào Ok thì kết quả trả về là giá trị được nhập từ hộp prompt(), nếu click vào Cancel thì giá trị trả về là null.

Cú pháp:

```
prompt("Dòng thông báo","Giá trị nhập vào mặc định");
```

Ví dụ 4.18:

```
<html>
<head>
<script type="text/javascript">
function show_prompt()
{
var name=prompt("Please enter your name","Harry Potter");
if (name!=null && name!="")
 {
 document.write("Hello " + name + "! How are you today?");
 }
}
</script>
</head>
<body>
<input type="button" onclick="show_prompt()" value="Show prompt box" />
</body>
</html>
```

9. Câu lệnh lặp For

Câu lệnh for dùng để lặp lại công việc với số lần lặp được xác định trước.

Cú pháp:

```
for (var=startvalue;var<=endvalue;var=var+increment)
{
 // Mã lệnh thực hiện;
}
```

Ví dụ 4.19:

```
<html>
<body>
<script type="text/javascript">
var i=0;
for (i=0;i<=5;i++)
{
document.write("The number is " + i);
```

```
document.write("<br />");
}
</script>
</body>
</html>
```

10. Câu lệnh lặp While

Câu lệnh lặp While:

Câu lệnh while dùng để lặp lại công việc với số lần lặp chưa xác định trước, số lần lặp phục thuộc vào điều kiện.

Cú pháp:

```
while (<biểu thức điều kiện>)
 {
   // Thực hiện mã lệnh;
  }
```

Kiểm tra điều kiện, nếu biểu thức điều kiện còn đúng thì thực hiện mã lệnh, điều kiện sai thì thoát khỏi vòng lặp.

Ví dụ 4.20:

```
<html>
<body>
<script type="text/javascript">
var i=0;
while (i<=5)
 {
 document.write("The number is " + i);
 document.write("<br />");
 i++;
 }
</script>
</body>
</html>
```

Câu lệnh do ... while

Thực hiện mã lệnh, sau đó kiểm tra điều kiện nếu điều kiện còn chưa đúng thì thực hiện thì thực hiện mã lệnh, điều kiện đúng thì thoát khỏi vòng lặp.

Cú pháp:

```
do
 {
 // Thực hiện mã lệnh;
 }
while (var<=endvalue);
```

Ví dụ 4.21:

```
<html>
<body>
<script type="text/javascript">
var i=0;
do
  {
  document.write("The number is " + i);
  document.write("<br />");
  i++;
  }
while (i<=5);
</script>
</body>
</html>
```

11. Câu lệnh lặp For...In

Câu lệnh for …in là một câu lệnh đặc biệt không có trong C chỉ có trong javascript, nó được dùng để duyệt các phần tử trong một đối tượng trong javascript. Ví dụ như trong đối tượng mảng.

Cú pháp:

```
for (variable in object)
  {
  // Mã lệnh thực hiện;
  }
```

Ví dụ 4.22:

```
<html>
<body>
<script type="text/javascript">
var x;
var mycars = new Array();
mycars[0] = "Saab";
```

```
mycars[1] = "Volvo";
mycars[2] = "BMW";
for (x in mycars)
 {
  document.write(mycars[x] + "<br />");
 }
</script>
</body>
</html>
```

12. Sự kiện trong Javascript

a. Sự kiện onLoad và onUnload

Sự kiện	Chức năng
onFocus	
onBlur	Sự kiện onBlur được thực hiện khi người dùng con trỏ di chuyển ra khỏi thẻ.
onChange	
onClick	
onActive	
onLoad	Sự kiện onLoad được thực hiện khi người dùng tải dữ liệu lên trình duyệt web.
onUnload	Sự kiện onUnload được thực hiện khi người dùng đó lại quá trình tải dữ liệu lên trình duyệt web.
onKeypress	
onSubmit	Sự kiện onSubmit được thực hiện khi người dùng thực hiện submit một form.
MouseOver	Sự kiện onMouseover được thực hiện khi người dùng đưa chuột lên thẻ.
onMouseOut	Sự kiện onMouseout được thực hiện khi người dùng đưa chuột ra khỏi thẻ.

13. Câu lệnh Try...Catch

Câu lệnh này kiểm tra một đoạn mã lệnh có bị lỗi Cú pháp hay không, nếu bị lỗi thì thông báo lỗi phát sinh ra.

Cú pháp:

```
try
  {
  //mã lệnh
  }
catch(err)
  {
  //thông báo lỗi
```

```
 }
```

Khi chạy thử "mã lệnh" nếu mã lệnh bị lỗi Cú pháp thì "thông báo lỗi" sẽ đưa ra.

Ví dụ 4.23:

```
<html>
<head>
<script type="text/javascript">
var txt="";
function message()
{
try
  {
  adddlert("Welcome guest!");
  }
catch(err)
  {
  txt="There was an error on this page.\n\n";
  txt+="Error description: " + err.description + "\n\n";
  txt+="Click OK to continue.\n\n";
  alert(txt);
 }
}
</script>
</head>
<body>
<input type="button" value="View message" onclick="message()" />
</body>
</html>
```

14. Câu lệnh Throw

Cú pháp:

```
throw(exception)
```

Ví dụ 4.24:

```
<html>
<body>
<script type="text/javascript">
var x=prompt("Enter a number between 0 and 10:","");
```

```
try
  {
  if(x>10)
    {
    throw "Err1";
    }
  else if(x<0)
    {
    throw "Err2";
    }
  else if(isNaN(x))
    {
    throw "Err3";
    }
  }
catch(er)
  {
  if(er=="Err1")
    {
    alert("Error! The value is too high");
    }
  if(er=="Err2")
    {
    alert("Error! The value is too low");
    }
  if(er=="Err3")
    {
    alert("Error! The value is not a number");
    }
  }
</script>
</body>
</html>
```

15. Ký tự đặc biệt Text

Mã	Hiện ra
\'	Dấu nháy đơn (’)

\"	Dấu nháy kép (”)
\&	Dấu và (&)
\\	Dấu xổ phải (\ )
\n	Xuống dòng mới
\r	carriage return
\t	Nhảy tab
\b	Khoảng trắng
\f	form feed

III. Đối tượng trong javascrip

Javascript là một ngôn ngữ lập trình hướng đối tượng, cho phép chúng ta định nghĩ ra đối tượng và sử dụng đối tượng đó, bên cạnh đó javascript cũng định nghĩa cho chúng ta một số đối tượng.

1. Đối tượng String

Đối tượng chuỗi được tạo ra để lưu trữ chuỗi các ký tự.

Ví dụ 4.25:

hoten = “Hồ Diên Lợi”;

Các khái báo chuỗi: Đối tượng chuỗi được khai báo như sau:

var ten_chuoi = “chuỗi ký tự”;

Ví dụ 4.26:

var hoten = “Hồ Diên Lợi”;

Một phương thức của đối tượng chuỗi:

- Độ dài của chuỗi: length
- Cộng hai chuỗi ký tự (+)
- Chuyển chuỗi ký tự thành chuỗi chữ in hoa: toUpperCase()
- Chuyển chuỗi ký tự thành chuỗi chữ thường: toLowerCase( )
- Cắt lấy chuỗi con từ vt1 đến vt2: substring(vt1,vt2)
- Cắt lấy một chuỗi: substr(vt, number)
- indexOf( )
- match(re);

- replace(re, *replacementString*);
- charCodeAt(3)

2. Đối tượng Date

Tạo một đối tượng date ta sử dụng các Cú pháp sau:

```
var myDate = new Date();
var myDate = new Date(yyyy, mm, dd, hh, mm, ss);
var myDate = new Date(yyyy, mm, dd);
var myDate = new Date("monthName dd, yyyy hh:mm:ss");
var myDate = new Date("monthName dd, yyyy");
var myDate = new Date(epochMilliseconds);
```

Một số phương thức đối với đối tượng Date:

getTime( )	setTime(val)	0-...
getSeconds( )	setSeconds(val)	0-59
getMinutes( )	setMinutes(val)	0-59
getHours( )	setHours(val)	0-23
getDay( )	setDay(val)	0-6
getDate( )	setDate(val)	1-31
getMonth( )	setMonth(val)	0-11
getFullYear( )	setFullYear(val)	1970-...

3. Đối tượng Array

- Concat(): nooid chuỗi

Ví dụ 4.27: Nối chuỗi

```
<html>
<body>
<script type="text/javascript">
var parents = ["Jani", "Tove"];
var children = ["Cecilie", "Lone"];
var family = parents.concat(children);
document.write(family);
</script>
</body>
</html>
```

- Sort

Ví dụ 4.28:

```
<html>
<body>
<script type="text/javascript">
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.write(fruits.sort());
</script>
</body>
</html>
```

Ví dụ 4.29:

- Split
- Length
- splice

4. Đối tượng Math

- Hàm round()

```
<html>
<body>
<script type="text/javascript">
document.write(Math.round(0.60) + "<br />");
document.write(Math.round(0.50) + "<br />");
document.write(Math.round(0.49) + "<br />");
document.write(Math.round(-4.40) + "<br />");
document.write(Math.round(-4.60));
</script>
</body>
</html>
```

- Hàm random()

```
<html>
<body>
<script type="text/javascript">
//return a random number between 0 and 1
document.write(Math.random() + "<br />");
//return a random integer between 0 and 10
document.write(Math.floor(Math.random()*11));
</script>
</body>
</html>
```

Chương 5: Ngôn ngữ PHP

I. Tổng quan về PHP

1. Cú pháp PHP

Cách 1 : Cú pháp chính:

<?php Mã lệnh PHP **?>**

Cách 2: Cú pháp ngắn gọn

<? Mã lệnh PHP **?>**

Cách 3: Cú pháp giống với ASP.

<% Mã lệnh PHP **%>**

Cách 4: Cú pháp bắt đầu bằng script

<script language=php>

.....

</script>

Mặc dù có 4 cách thể hiện. Nhưng đối với 1 lập trình viên có kinh nghiệm thì việc sử dụng cách 1 vẫn là lựa chon tối ưu.

2. Xuất giá trị ra trình duyệt

Để xuất dữ liệu ra trình duyệt chúng ta sử dụng các hàm hàm sau:

- Hàm **echo()**

Cú pháp:

```
echo (<thông tin>);
```

Trong đó:

+ Thông tin có thể là hằng, biến hay biểu thức.

+ Có thể viết **echo** <thông tin>;

Ví dụ 5.1: Xuất ra trình duyệt nội dung "Chào các bạn"

```
<?php
    echo "Chào các bạn";
    echo ("Chào các bạn");
?>
```

- Hàm **print()**

Cú pháp:

```
print(<thông tin>);
```

Trong đó:

+ Thông tin có thể là hằng, biến hay biểu thức.

+ Có thể viết print <thông tin>;

Ví dụ 5.2: Xuất ra trình duyệt nội dung "Chào các bạn"

```
<?php
    print "Chào các bạn";
    print ("Chào các bạn");
?>
```

3. Lời chú thích

Để giải thích các đoạn mã hay câu lệnh trong PHP thực hiện chức năng, nhiệu vụ ngông ngữ đã có pháp người dùng viết các ghi chú.

Lời ghi chú không thực thi khi chạy đoạn mã.

Trong HTML, các ghi chú được đặt trong <!-- phần ghi chú -->

Trong PHP có ba dạng ghi chú:

Dạng 1: # đây là ghi chú.

Dạng này chỉ áp dụng ghi đó chỉ nằm trên một dòng văn bản

Dạng 2: // đây là ghi chú.

Dạng này cũng chỉ áp dụng ghi đó chỉ nằm trên một dòng văn bản

Dạng 3: /* đây là một ghi chú dài

Áp dụng cho nhiều hàng */

Dạng ghi chú này thường áp dụng cho đoạn ghi chú dài gồm nhiều dòng văn bản

4. Biến trong PHP

Biến là một ô nhớ trong bộ nhớ chính giúp biểu diễn thông tin thực tế trong chương trình.

a. Khai báo biến

- Biến trong PHP được bắt đầu bằng ký tự $ và theo sau là tên biến.

- Nguyên tắc đặt tên biến: tên biết bắt đầu bằng một ký tự hoặc là dấu _theo sau có thể là ký tự, số hoặc dấu _

Chú ý:

PHP không yêu cầu khai báo biến trước khi sử dụng, tuy nhiên chúng ta nên khai báo và khởi gán giá trị ban đầu cho biến trước khi sử dụng.

Không khai báo tên biến trùng với tên hàm, tên biến phân biệt chữ HOA và chữ thường.

Ví dụ 5.3: Khai báo biến

```
<?php
        $chuoi = "Hello world";
        $_123 = 123.4567;
?>
```

b. Gán giá trị cho biến.

Khi ta gán giá trị cho biến, nếu biến chưa được khai báo lúc này biến vừa được khai báo vừa được khởi tạo, nếu biến đã được khai báo thì biến này sẽ có giá trị bằng với giá trị được gán.

- Biến có thể khởi gán giá trị trực tiếp:

```
$tên_biến = <giá trị của biến>;
```

Ví dụ 5.4: Khởi gán giá

```
<?php
        $number = 120;
        echo $number;
?>
```

- Biến có thể khởi gán giá trị của biến khác hoặc của một biểu thức.

```
$tên_biến =$tên_biến_khác hoặc biểu thức;
```

Ví dụ 5.5: Khởi gán giá trị của một biểu thức.

```
<?php
        $number = 120;
        $gia =1000;
        $thanh_tien =$number * $gia;
        echo $thanh_tien;
?>
```

c. Phạm vi hoạt động của biến

Một biến trong PHP có phạm vi hoạt động như sau:

- Biến cục bộ: Khi một biến được khai báo trong một hàm thì nó được xem là biến cục bộ và nó chỉ có ý nghĩa sử dụng trong hàm đó. Khi gán giá trị cho biến bên ngoài thì biến ngoài hàm này sẽ được xem như một biến hoàn toàn khác với biến trong hàm cho dù cùng tên. Khi ra khỏi hàm có biến cục bộ được khai báo thì biến và giá trị cho nó sẽ được hủy bỏ.

Ví dụ 5.6: Biến cục bộ

```
<?php
$a= 10; //biến toàn cục
function test()
{
	echo $a; // biến cục bộ
}
test(); // không cho kết quả
echo $a; // kết quả 10
?>
```

- Biến toàn cục: Là biến có thể truy cập ở bất kỳ nơi nào trong chương trình. Tuy nhiên, để có thể sử dụng và cập nhật được biến toàn cục thì phải được khai báo toàn cục (khai báo với global hay $_GLOBALS) trong hàm mà nó được sử dụng.

Ví dụ 5.7: Biến toàn cục

```
<?php
$a = 10;
$b = 20;
function sum()
{
	global $a, $b;
	$b = $a+$b;
}
sum();
echo $b; // kết quả 30
?>
```

- Biến static: khác với biến cục bộ, biến static không mất giá trị của nó khi ra khỏi hàm và nó giữa nguyên giá trị đó khi hàm đó được gọi thêm một lần nữa.

Ví dụ 5.8: Biến static

```
<?php
function hien()
{
	static $a =0;
```

```
    echo $a."</br>";
    $a++;
}
hien(); // kết quả 0
hien(); // kết quả 1
hien(); // kết quả 2
?>
```

5. Hằng

a. Khái báo hằng

Hằng là một giá trị không thể chỉnh sửa được thông qua việc thực hiện chương trình, quy tắc đặt tên hằng cũng giống như quy tắc đặt tên biến.

Chúng ta có thể định nghĩa hằng bằng cách sử dụng hàm **define()**. Một khi hằng được định nghĩa, nó không bị thay đổi.

Chỉ có các kiểu dữ liệu boolean, integer, float, string mới có thể chứa hằng.

Cú pháp:

```
define("TÊN_HẰNG", "giá trị");
```

Ví dụ 5.9: Tính diện tích đường tròn

```
<?php
define("PI", 3.14);
$r = 10;
echo "Diện tích đường tròn". 2*PI*$r*$r;
?>
```

Chú ý: PHP cung cấp một lượng lớn các hằng đã được định nghĩa trước để bất kỳ trang nào có thể thực thi được.

- _FILE_ : tên của script file đang được thực hiện.
- _LINE_: số dòng của mã script đang được thực hiện trong script file hiện tại.
- _PHP_VERSION_: version của PHP
- TRUE
- FALSE
- E_ERROR: báo hiệu có lỗi.
- E_PARSE: báo lỗi sai khi biên dịch.

- E_NOTICE: một vài sự kiện có thể là lỗi hoặc không.
- E_ALL: tất cả các lỗi.
- ….

b. Sử dụng hằng

Đối với hằng đã khai báo, chúng ta dùng tên hằng mỗi khi sử dụng.

Ví dụ 5.10: Tính diện tích và chu vi hình tròn

```
<?php
        define('PI',3.14); // định nghĩa PI = 3.14
        $r = 10;
        $chu_vi = 2*PI*$r; // 62.8
        echo $chu_vi."</br>";
        $dien_tich = 2*PI*$r*$r; //628
        echo $dien_tich;
?>
```

Những điểm khác nhau giữa hằng và biến:

- Phía trước tên hằng không có dấu $
- Hằng chỉ có thể được khai báo bằng hằng **define()**
- Không khai báo lại hằng khi đã được thiết lập.

6. Kiểu dữ liệu

a. Kiểu dữ liệu

Kiểu dữ liệu	Mô tả
boolean	Chỉ có một trong hai giá trị TRUE và FALSE
integer	Kiểu số nguyên, giá trị có thể là số trong hệ thập phân, thập lục phân và bát phân.
float/double	Kiểu số thực
string	Kiểu dữ liệu chuỗi, ký tự. Trong đó, mỗi ký tự chiếm 1 byte. Mỗi chuỗi có thể chứa một hay nhiều ký tự thuộc 256 ký tự khác nhau. Không có vấn đề gì xảy ra khi chuỗi quá lớn vì chuỗi không có giới hạn về kích thước. Mỗi chuỗi được ghi theo những cách sau: - Dùng dấu nháy đơn ' ' để bao chuỗi. - Dùng dấu nháy đôi " " để bao chuỗi.
array	Kiểu dữ liệu là mảng các phần tử.
object	Kiểu dữ liệu là đối tượng của lớp

Ví dụ 5.11: Kiểu dữ liệu số

```
<?php
        $kieu_b = TRUE;
        echo $kieu_b."</br>"; // kết quả là 1
        $inta = 1234;
        echo $inta."</br>"; // kết quả là 1234
        $intb = -123;
        echo $intb."</br>"; // kết quả là  -123
        $intc = 0123;
        echo $intc."</br>"; // kết quả là 83
        $intd = 0x1A;
        echo $intd."</br>"; // kết quả là 26
        $float_a = 1.234;
        echo $float_a."</br>"; // kết quả là  1.234
        $float_b = 1.2e3;
        echo $float_b."</br>"; // kết quả là 1200
        $float_c = 7E-10;
        echo $float_c."</br>"; // kết quả là 7.0E-10
?>
```

Ví dụ 5.12: Kiểu dữ liệu string

```
<?php
        $name = 2010;
        $chuoi1 ='Chúc mừng năm mới năm $name';
        $chuoi2 ="Chúc mừng năm mới năm $name";
        echo $chuoi1."</br>"; // Chúc mừng năm mới năm $name
        echo $chuoi2."</br>"; // Chúc mừng năm mới năm 2010
?>
```

Ví dụ 5.13: Kiểu dữ liệu mảng

```
<?php
$mang = array(1,2,3,4,5);
print_r($mang);
// hàm in ra giá tri của mảng theo dạng Array ( [0] => 1 [1] => 2 [2] => 3 [3] => 4 [4] => 5 )
?>
```

Ví dụ 5.14: Kiểu object

```
<?php
class ten_class
```

```
{
	function xuatchao()
	{
		echo "Hello my class!";
	}
}
$a = new ten_class();
$a ->xuatchao(); // kết quả Hello my class
?>
```

b. Chuyển đổi kiểu dữ liệu

Trong quá trình tính toán chúng ta có thể thực hiện việc chuyển đổi kiểu dữ liệu cho biến bằng cách ghi tên kiểu dữ liệu mà biến muốn chuyển đổi vào phía trước biến.

Thường chúng ta chuyển đổi kiểu dữ liệu cho biến vì trong quá trình tính toán kiểu dữ liệu cũ của biến có thể không còn phù hợp nữa.

Ví dụ 5.15: Chuyển đổi kiểu dữ liệu

```
<?php
	$don_gia = 5000;
	$so_luong = 10000;
	$thanh_tien = (double)($so_luong * $don_gia);
	echo $thanh_tien;
?>
```

7. Các toán tử

a. Toán tử toán học

Toán tử toán học gồm các phép toán sau:

+ (cộng), - (trừ), * (nhân), / (chia), % (chia lấy phần dư), ++ (tăng giá trị biến lên 1 đơn vị)
- - (giảm giá trị biến 1 đơn vị)

b. Toán tử nối chuỗi

c. Toán tử gán kết hợp

Phép toán	Ví dụ	Tương đương
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y

*=	x*=y	x=x*y
/=	x/=y	x=x/y
.=	x.=y	x=x.y
%=	x%=y	x=x%y

d. Toán tử so sánh

Toán tử so sánh gồm các phép toán sau:

==	Bằng
!=	Không bằng
<>	Khác
>	Lớn hơn
<	Nhỏ hơn
>=	Lớn hơn hoặc bằng
<=	Nhỏ hơn hoặc bằng

e. Toán tử logic

Toán tử logic gồm các phép sau:

&&	Và
||	So sánh
!	Phủ định

f. Toán tử @

Trong trương hợp biểu thức hay phép toán của chúng ta phát sinh lỗi, nếu chúng ta không muốn xuất hiện ra thông báo lỗi thì chúng ta dùng toán tử @

Ví dụ 5.16: Khi chưa sử dụng toán tử @

```
<?php
    $a = 10;
    $b = 0;
    $c =$a/$b;
    echo  "Kết quả :".$c;
?>
```

Màn hình xuất hiện:

Warning: Division by zero in **C:\wamp\www\vd.php** on line **4**

Kết quả :

Ví dụ 5.17: Khi sử dụng toán tử @

```
<?php
    $a = 10;
```

```
    $b = 0;
    $c =@($a/$b);
    echo  "Kết quả :".$c;
?>
```

Màn hình xuất hiện:

Kết quả :

g. Tham chiếu &

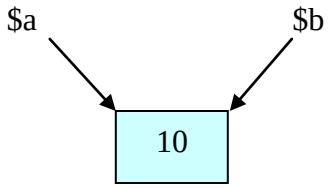
Trong PHP tham chiếu có nghĩa là lấy cùng giá một trị bằng nhiều tên biến khác nhau.

Ký hiệu tham chiếu là &.

Ví dụ 5.18: Biến $a và $b tham chiếu đến ô nhớ lưu giá trị biến $a

```
<?php
    $a = 10;
    $b = &$a;
    echo  "Kết quả :".$b;
?>
```

Ví dụ trên có nghĩa, $b không lấy giá trị của biến $a mà biến $b và biến $a cùng lấy một nội dung trong cùng một ô nhớ.



8. Các hàm kiểm tra giá trị

a. Kiểm tra tồn tại isset()

Hàm isset() dùng để kiểm tra xem biến có giá trị hay không.

Hàm này có thể dùng để kiểm tra sự tồn tại của một hay nhiều biến khác nhau. Nếu tất cả các biến đều có giá trị kết quả trả về bằng true, ngược lại trả về giá trị false.

Cú pháp:

```
isset(<tên biến 1>, <tên biến 2>,…)
```

Ví dụ 5.19: Kiểm tra xem người dùng có nhập vào tên đăng nhập hay chưa, nếu đã nhập thì in ra "Xin chào <tên đăng nhập>" ngược lại thì in ra "Vui lòng nhập tên đăng nhập".

```
<?php
```

```
if (isset($_POST[“tên_đăng_nhập”]))
{
      echo “Xin chào: ”.$_POST[“tên_đăng_nhập”];
} else {
      echo “Vui lòng nhập tên đăng nhập”;
}
?>
```

Chú ý: Nếu muốn in kết quả của hàm isset() thì ta có thể dùng hàm var_dump().

Ví dụ 5.20: Kiểm tra tên biến có phải là

```
<?php
$chuoi ="abc";
var_dump(isset($chuoi)); // kết quả bool(TRUE)
$a = NULL;
$b =123;
var_dump(isset($a));// kết quả bool(FALSE)
var_dump(isset($b));// kết quả bool(TRUE)
var_dump(isset($a,$b));// kết quả bool(FALSE)
?>
```

b. Kiểm tra giá trị rỗng empty()

Hàm empty() dùng để kiểm tra biến có giá trị rỗng hay không.

Nếu biến có giá trị NULL, chuỗi rỗng hoặc O thì kết quả trả về là TRUE, ngược lại trả về giá trị là FALSE.

Hàm này ngược lại với hàm isset(), và thường được dùng để kiểm tra xem người dùng có nhập giá trị vào một đối tượng nào đó trên form hay không.

Những kết quả dưới đây được xem là rỗng:
+ “” : chuỗi rỗng.
+ 0: 0 khi kiểu là integer.
+ NULL
+ FALSE
+ array(): mảng rỗng.
+ var $var: một biến được khai báo nhưng không có giá trị trong lớp.

Cú pháp:

```
empty(<tên_biến>)
```

Ví dụ 5.21: Kiểm tra dữ liệu có rỗng hay không

```
<?php
if(empty($_POST["ten_dang_nhap"]))
{
	echo "Vui long nhap vao tên dang nhap";
	exit;
} else {
echo "Xin cho: ".$_POST["ten_dang_nhap"];
}
?>
```

c. Kiểm tra giá trị số is_numeric()

Hàm is_numeric() kiểm tra biến có kiểu giá trị kiểu số hay không.

Nếu giá trị của biến không phải là kiểu số thì kết quả trả về là TRUE, ngược lại thì kết trả về là FALSE.

Cú pháp:

```
is_numeric(<tên_biến>)
```

Ví dụ 5.22: Kiểm tra dữ liệu nhập vào có phải là kiểu số không?

```
<?php
if(is_numeric($_POST["so_luong"]))
{
	$so_luong = $_POST["so_luong"];
	$don_gia =$_POST["don_gia"];
	$thanh_tien = $so_luong * $don_gia;
	exit;
} else {
echo "So luong phai la kieu so"];
}
?>
```

d. Kiểm tra kiểu giá trị của tên biến

- is_int() và is_long()

Hàm is_int() hoặc is_long()kiểm tra giá trị của biến có phải là số nguyên hay không.

Nếu giá trị của biến là số nguyên thì kết quả trả về là TRUE, ngược lại trả về giá trị là FALSE.

Cú pháp:

```
is_int(<tên_biến>); hoặc is_long(<tên_biến>);
```

Chú ý: Trong trường hợp biến vượt quá phạm vị của số nguyên thì chúng ta có hàm is_long() dùng để kiểm tra giá trị của biến có phải là kiểu long hay không. Hàm này có Cú pháp tương tự như hàm in_long().

Ví dụ 5.23: Kiểm tra kiểu dữ liệu của biến có phải là số nguyên không.

```
<?php
    $a ="15";
    $b =15;
    echo is_int($a); // kết quả trả về là 0
    echo is_int($b); // kết quả trả về là 1
?>
```

- is_string()

Hàm is_string() kiểm tra giá trị của biến có phải là kiểu chuỗi hay không.

Nếu giá trị là kiểu chuỗi thì kết quả trả về là true, ngược lại trả về giá trị là false.

Cú pháp:

```
is_string(<tên_biến>)
```

Ví dụ 5.24: Kiểm tra kiểu dữ liệu nhập vào là kiểu chuỗi không?

```
<?php
    $a ="Hello";
    $b =15.5;
    echo is_string($a); //kết quả là 1
    echo is_string($b); // kết quả là 0
?>
```

- is_double()

Hàm is_double() kiểm tra giá trị của biến có phải là kiểu số có dấu chấm động.

Nếu giá trị của biến là kiểu số dấu chấm động, số lẽ thì giá trị trả về là true, ngược lại thì trả về là false.

Cú pháp:

is_double(<tên biến>)

Ví dụ 5.25: Kiểm tra dữ liệu nhập vào có phải là số thực không?

```
<?php
    $x =4.1123;
```

```
    echo is_double($x); // kết quả trả về là 1
?>
```

e. Xác định kiểu dữ liệu biến

Hàm gettype() kiểm tra kiểu dữ liệu của biến, hoặc giá trị là kiểu nào: integer, string, double, array, object, class…

Kết quả trả về của hàm là kiểu của giá trí hay kiểu của biến.

Cú pháp:

gettype(<tên biến>)

Ví dụ 5.26: Kiểm tra kiểu dữ liệu nhập.

```
<?php
    $n = "day la chuoi";
    $a =123;
    $b =123.456;
    $mang =array(1,2,3);
    echo gettype($n)."</br>";
    echo gettype($a)."</br>";
    echo gettype($b)."</br>";
    echo gettype($mang);
?>
```

II. Câu lệnh điều khiển

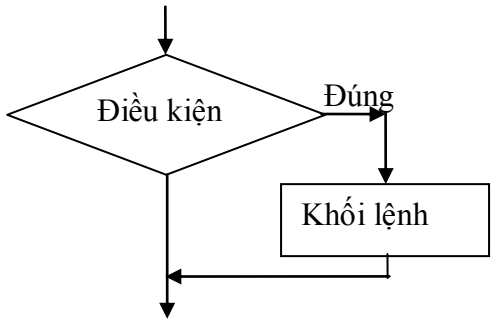
1. Câu lệnh rẽ nhánh If...Else

Dạng 1: Câu lệnh if dạng khuyết

Nếu điều kiện đúng thì thực hiện khối lệnh bên trong if sẽ được thực hiện. Ngược lại thì bỏ qua.

Cú pháp:

```
if ( <điều kiện>)
{
    // khối lệnh
}
```



Điều kiện có thể là biểu thức so sánh giá trị TRUE/FALSE hoặc là một giá trị số. Nếu giá trị số khác 0 thì giá trị trả về là TRUE, ngược lại trả về giá trị FALSE.

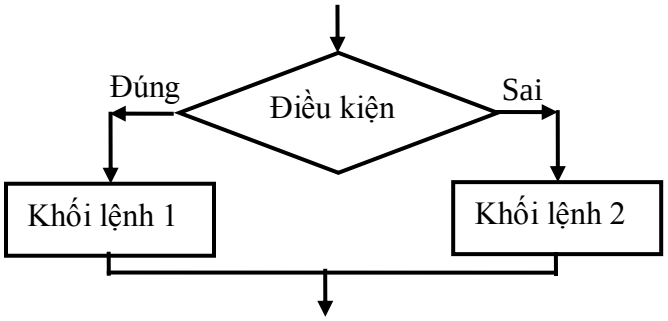
Khối lệnh: các lệnh sẽ được thực hiện khi điều kiện có giá trị là TRUE.

Dạng 2: Câu lệnh if dạng đầu đủ

Nếu điều kiện đúng thì thực hiện khối lệnh 1 bên trong if sẽ được thực hiện. Ngược lại thì thực hiện khối lệnh 2.

Cú pháp:

```
if ( <điều kiện>)
{
        // khối lệnh1
} else {
        // khối lệnh 2
}
```



Toán tử ?

Toán tử dấu ? dùng để thay thế câu lệnh if … else với một câu lệnh bên trong.

(<điều kiênh>)? <kết quả khi điều kiện đúng>: <kết quả khi điều kiện sai>;

Ví dụ 5.27: Tìm số lớn nhất trong hai số

```
<?php
if(strlen($_POST['a'])&& strlen($_POST['b']))
{
        $a = $_POST['a'];
        $b =$_POST['b'];
        $kq = ($a>$b)?$a:$b;
} else {
$kq = "Bạn chưa nhập thông tin vào";
}
?>
```

Dạng 3: Câu lệnh if lồng

Trong trường hợp có nhiều điều kiện thì chúng ta sử dụng câu lệnh if lồng nhau.

Cú pháp:

```
if (<điều kiện 1>)
{
        // khối lệnh 1
} elseif (<điều kiện 2>)
{
        // khối lệnh 2
```

```
}
…
else
{
      // khối lệnh khi không thỏa các điều kiện trên
}
```

2. Câu lệnh lựa chọn switch

Trong trường hợp có nhiều điều kiện xẩy ra. Trong trường hợp muốn so sánh giá trị của biến với biểu thức, và đối với mỗi giá trị này sẽ có những xử lý khác nhau thì ta dùng switch … case.

Cú pháp:

```
switch (<biểu thức>)
{
case <giá trị 1> :
        // khối lệnh khi biểu thức thõa mãn điều kiện 1
        break;
case <giá trị 2> :
        // khối lệnh khi biểu thức thõa mãn điều kiện 2
        break;
….
default:
      // khối lệnh khi không thõa tất cả các case trên.
}
```

3. Câu lệnh lặp

a. Cấu trúc for/foreach

- Cấu trúc for

For được dung khi chúng ta biết trước số lần lặp, biến đếm chạy trong khoảng giới hạn của vòng lặp, và giá trị lặp.

Vòng lặp sẽ kết thúc khi biến đếm vượt qua giới hạn của vòng lặp.

Cú pháp:

```
for ($biến chạy = <giá trị đầu>; <điều kiện của vòng lặp>; <giá trị lặp>)
{
      // Khối lệnh
}
```

- Câu trúc foreach

Cấu trúc foreach thường được dùng đẻ duyệt tập hợp(mảng). Cấu trúc này sẽ duyệt từ phần tử đầu tiên đến phần tử cuối cùng của tập hợp(mảng).

Cú pháp:

```
foreach ($tên_mảng as $giá_trị)
{
	// khối lệnh
}
```

Ví dụ 5.28: Vòng lặp foreach

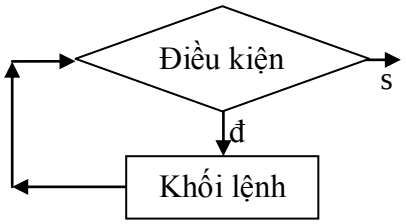
```
<?php
if(strlen($_POST['mang']))
{
	$mang =explode(",",$_POST['mang']);
	foreach($mang as $pt)  {
	  echo $pt."  ";
	 }
}
?>
```

b. Cấu trúc while

Khi chúng ta không xác định được số lần lặp(số lần lặp phụ thuộc vào điều kiện tại thời điểm thực thi) thì chúng ta sử dụng cấu trúc whlie.

Cú pháp:

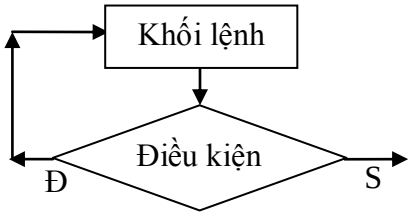
```
while (<điều kiện>)
{
	// Khối lệnh
}
```



c. Cấu trúc do … while

Cú pháp:

```
do
{
	// khối lệnh
}
while (<điều kiện>);
```



4. Sử dụng break và continue trong cấu trúc lặp

a. Lệnh break

break cho phép ta thoát khỏi cấu trúc điều khiển dựa trên kết quả của biểu thức logic

Ví dụ 5.28: Kiểm tra số nguyên tố

```
<?php
$so = 15;
$kq =true;
for ($i=2; $i<=$so; $i++)
{
	if($so%$i==0)
	{
		$kq= false;
		break;
	}
}
?>
```

b. Lệnh continue

Khi gặp continue, các lệnh bên dưới continue tạm thời không thực hiện tiếp, khi đó con trỏ sẽ nhảy về đầu vòng lặp để kiểm tra giá trị của biểu thức điều kiện còn đúng hay không. continue thường đi kèm với một biểu thức logic

Ví dụ 5.29: Tính tổng các phần tử lẻ từ 1 đến 10

```
<?php
$tong = 0;
for ($i=1; $i<=10; $i++)
{
	if($i%2==0)
	{
		continue;
	}
	$tong = $tong+$i;
}
echo $tong;
```

```
?>
```

5. Kiểu mảng

a. Khái niệm mảng

Mảng nói chung là một biến đặc biệt, nó bao gồm một dãy các ô nhớ có nhiều ô nhớ con cho phép biểu diễn thông tin dạng danh sách trong thực tế. Các phần tử mảng có thể có kiểu dữ liệu khác nhau.

b. Khai báo mảng và sử dụng mảng

Cách 1: Khai báo mảng chưa biết số phần tử mảng

Cú pháp: $ten_mang = array();

Ví dụ 5.30: Khai báo mảng chưa biết số phần tử mảng

```
<?php
$mang = array();
for($i=0; $i<10; $i++)
        $mang[$i] = $i;
?>
```

Cách 2: Khai báo biết trước số phần tử mảng

Cú pháp: $ten_mang = array(<số phần tử mảng>);

Ví dụ 5.31: Khai báo mảng biết trước số phần tử mảng

```
<?php
$mang = array(20);
for($i=0; $i<20; $i++)
        $mang[$i] = $i;
?>
```

Cách 3: Nếu khai báo mảng biết trước giá trị của mảng thì chúng ta vừa khai báo vừa gán giá trị.

Cú pháp: $ten_mang = array([khóa =>] giá_trị_1, …);

Trong đó:

+ Khóa: có thể là số nguyên dương hoặc chuỗi.

+ Khóa không được trung nhau.

+ giá_trị_1, …: có thể dùng tất cả các kiểu dữ liệu.

Ví dụ 5.32:

```
<?php
    $mang_1 = array(1,2,3,4,5); // không tạo giá trị cho khóa
    $mang_2 = array(1=>"Một", 2 =>"Hai", 3=>"Ba", 4 =>"Bốn", 5=>"Nam");
    $mang_3 = array("mot"=>1, "hai" =>2, "ba"=>3, "bon" =>4, "nam"=>5);
?>
```

Cách 4: Gán giá trị cho từng phần tử mảng

Cú pháp:

```
$ten_mang[ ] =<giá trị>;
Hoặc  $ten_mang[<giá trị khóa>] =<giá trị>;
```

c. Truy xuất phần tử mảng.

Cú pháp:

```
$ten_bien = $ten_mang[<giá trị khóa>];
```

d. Các thao tác trên mảng

- Đếm số phần tử mảng

Để đếm số phần tử của mảng một chiều ta sử dụng hàm count(< tên_biến_mảng>)

$so_pt_mang = count($ten_mang);

- Duyệt mảng

+ Duyệt mảng có khóa tự động

Ví dụ 5.33:

```
<?php
    $n = count($mang_1);
    for($i  = 0; $i< =$n; $i++)
        echo "\t".$mang_1[$i];
?>
```

+ Duyệt mảng có khóa do người dùng tạo:

Ví dụ 5.34:

```
<?php
foreach( $mang_2 as $gia_tri)
{
    echo "\t $gia_tri";
}
?>
```

+ Duyệt để lấy cả giá trị của khóa và giá trị của phần tử:

Ví dụ 5.35:

```
<?php
    foreach( $mang_2 as $khoa => $gia_tri)
    {
        echo "<br> [$khoa] => $gia_tri";
    }
?>
```

e. Một số hàm

- Tìm kiếm trên mảng: array_search()

Hàm này sẽ tìm kiếm một giá trị trên mảng, nếu tìm thấy sẽ trả về khóa của phần tử chứa giá trị đó, nếu không tìm thấy sẽ trả về giá trị NULL

Cú pháp:

$khoa = array_search($gia_tri_can_tim, $mang);

Ví dụ 5.36:

```
<?php
    $mang = array(0=>'xanh', 1=>'đỏ', 2=>'tím', 3=>'vàng');
    $khoa = array_search('đỏ', $mang);
    echo $khoa;
?>
```

- Ghép mảng: array_merge()

Ghép hai mảng hay nhiêu mảng với nhau, kết quả trả về là một mảng mới được tạo ra từ các mảng.

Cú pháp:

$mang_ghep = array_merge($mang_1, $mang_2,…);

Chú ý: Khi các mảng dùng để ghép có khóa trùng nhau thì mảng ghép sẽ chỉ lấy phần tử có khóa trùng của mảng cuối cùng.

Ví dụ 5.37:

```
<?php
    $mang1= array("màu"=>"đỏ", 2, 4);
    $mang2 = array("a", "b", "màu"=> "xanh", "hình"=>"tròn", 4);
    $mang_chung = array_merge($mang1, $mang2);
    print_r($mang_chung);
?>
```

- Đếm số lần xuất hiện: array_count_values()

Dùng để đếm số lần xuất hiện của các phần tử trong mảng. Kết quả trả về là một mảng trong đó khóa chính là giá trị trên mảng cần đếm và giá trị sẽ là số lần xuất hiện của nó trong mảng.

Cú pháp:

$mang_slxh = array_count_values($ten_mang);

Ví dụ 5.38:

```
<?php
    $mang = array(1, "hello",1, "world", "hello", 2 , "xin chào",1);
    $mang_slxh = array_count_values($mang);
    print_r($mang_slxh);
?>
```

- Tạo mảng duy nhất: array_unique()

Hàm sẽ bỏ đi những giá trị lặp đi lặp lại trong mảng. Kết quả trả về là một mảng mới mà trong đó mỗi phần tử trong mảng chỉ xuất hiện một lần.

Cú pháp:

$mang_day_nhat = array_unique($ten_mang);

Ví dụ 5.39:

```
<?php
    $mang = array(1, 3, 1, 2, 5, 1, 3, 4);
    $mang_dn = array_unique($mang);
    print_r($mang_dn);
?>
```

- Tìm các giá trị khác nhau của một mảng so với mảng khác: array_diff()

Hàm sẽ so sánh giữa hai mảng và lọc ra những phần tử chỉ có trong mảng thức nhất mà không có trong mảng thứ hai. Kết quả trả về là một mảng mới với những phần tử chỉ xuất hiện duy nhất trong một mảng.

Cú pháp:

array_diff($mang_1, $mang_2);

Ví dụ 5.40:

```
<?php
    $mang_1 = array("a"=>"xanh", "đỏ", "tím", "vàng");
    $mang_2 = array("b"=>"xanh", "vàng", "đỏ");
    $mang =array_diff($mang_1, $mang_2);
    print_r($mang);
?>
```

III. Xây dựng hàm trong PHP

1. Hàm do người dùng định nghĩa

a. Khai báo hàm

Để khai báo hàm chúng ta sử dụng từ khóa function, tiếp sau đó là tên hàm và danh sách các tham số (nếu có), các lệnh của hàm được đặt trong cặp ngoặc { }.

Cú pháp:

```
function ten_ham([danh sách tham số])
{
    // khối lệnh bên trong hàm
    [returm gia_trị;]
}
```

Trong đó:

Nếu hàm cần tham số để xử lý thì truyền vào tham số để xử lý, ngược lại có thể bỏ trống.

Nếu hàm trả lại giá trị thì giá trị trả về của hàm là: returm gia_trị, nếu không có giá trị trả về thì không có lệnh trả về.

Ví dụ 5.41: Xuất ra câu chào "Hello world";

```
<?php
function cau_chao()
{
	echo "Hello world";
}
?>
```

Ví dụ 5.42: Hàm trả về giá trị

```
<?php
function tinhtong($a,$b)
{
	$s =$a+$b;
	return $s;
}
$so_a =5;
$so_b = 10;
$tong = tinhtong($so_a,$so_b)+tinhtong(2,3);
echo "Tổng = $tong"; // kết quả trả về 20
?>
```

b. Sử dụng hàm

Hàm sau khi được tạo có thể được gọi lại thông qua tên hàm, nếu hàm có các thông tin bên trong thì cung cấp đầy đủ các thông tin, nếu hàm có giá trị trả về thì phải có biến để nhận giá trị của hàm.

Cú pháp:

```
ten_ham([danh sách các giá trị]);
```

Trong đó:

- ten_ham được gọi đúng với hàm được định nghĩa.
- Danh sách giá trị: cung cấp các thông tin cho các tham số của hàm.

Ví dụ 5.43: Gọi hàm xuất ra câu chào

```
<?php
    cau_chao();
?>
```

2. Hàm trong thư viện hàm

a. Kiểu dữ liệu string

Kiểu dữ liệu string dùng để lưu trữ chuỗi các ký tự.

Ví dụ 5.44:

```
$hoten = "Hồ Diên Lợi";
```

Một số hàm xử lý chuỗi.

- Hàm **ltrim(str [,char]):** Xoá khoảng trắng từ bên trái của chuỗi, nếu có tham số char thì sẽ bỏ luôn các các ký tự bên trái trong char.

Ví dụ 5.45: Cắt bỏ các ký tự dư thừa bên trái chuỗi

```
<?php
    $st="aaaa   Hoàng Nam";
    $st = ltrim($st,'a');
    echo $st; //"Hoàng Nam"
?>
```

- Hàm **rtrim(str [,char]):**Xoá khoảng trắng từ bên phải của chuỗi, nếu có tham số char thì sẽ bỏ luôn các các ký tự bên phải trong char.

Ví dụ 5.46: Cắt bỏ các ký tự dư thừa bên phải

```
<?php
    $st="Hoàng Nam    aaaa  ";
    $st = rtrim($st,'a');
    echo $st; //"Hoàng Nam"
?>
```

- Hàm **trim($st [,char]):** Loại bỏ kí tự thừa ở đầu và cuối của xâu, nếu có tham số char thì sẽ bỏ luôn các các ký tự bên phải trong char.

Ví dụ 5.47: Cắt bỏ các ký tự dư thừa bên trái và bên phải

```
<?php
    $st="  aaaa      Hoàng Nam    aaaa  ";
    $st = trim($st,'a');
    echo $st; //"Hoàng Nam"
?>
```

- Hàm **addslashes($st):** định dạng dữ liệu trong chuỗi để lưu vào CSDL.

Để lưu chuỗi có các dấu nháy ' hay cặp ", dấu \, dấu \\ thì chúng ta dùng thêm dấu \ vào phía trước chúng như sau: \', \", \\, \\\.

Ví dụ 5.48: Định dạng dữ liệu

```
<?php
        $st ="Who're you?";
        echo $st."</br>";// Who're you?
        echo addslashes($st); //Who\'re you?
?>
```

Vì các dấu nháy khi lưu vào trong CSDL sẽ xuất hiện dấu \ trước các ký tự, khi đọc lên trình duyệt chúng ta cần loại bỏ các dấu đó. Để loại bỏ ta sẽ sử dụng hàm **stripslashes()**

- Hàm **stripslashes():**

Ví dụ 5.49: Loại bỏ các ký tự.

```
<?php
$st ="Who\'re you?";
echo stripslashes($st); //Who're you?
?>
```

- Hàm **ucfirst($st)** : Viết hoa kí tự đầu tiên của một xâu.

Ví dụ 5.50: Viết hoa ký tự đầu tiên của một xâu.

```
<?php
        $st = 'xin chào!';
        echo ucfirst($st);          // Xin chào!
?>
```

- Hàm **ucwords($st):** Viết hoa kí tự đầu tiên của mỗi từ.

Ví dụ 5.51: Viết hoa kí tự đầu tiên của mỗi từ.

```
<?php
        $st = 'xin chào!';
        echo ucwords($st); // Xin Chào!
?>
```

- Hàm **strtolower($st)** : Biến kí tự bất kỳ thành chữ thường.

Ví dụ 5.52: Biến kí tự bất kỳ thành chữ thường

```
<?php
        $st = 'HỒ dIên LợI!';
```

```
        echo strtolower($st); // hồ diên lợi
?>
```

- Hàm **strtoupper($st):** biến kí tự bất kỳ thành chữ hoa.

Ví dụ 5.53: Biến kí tự bất kỳ thành chữ hoa.

```
<?php
        $st = 'HỒ dIên LợI!';
        echo strtoupper($st); // Hồ DIÊN LỢI
?>
```

- Hàm **strlen($st)**: Kết quả trả về độ dài của xâu

Ví dụ 5.54: Xuất ra trình duyệt độ dài lớn

```
<?php
        $hoten = 'Hồ Diên Lợi';
        echo "Độ dài xâu  " . strlen($hoten); //Kết quả trả về là 13
?>
```

- Hàm **strcmp($str1, $str2)**: hàm so sánh chuỗi không phân biệt chữ hoa và chữ thường, hàm này trả về kết quả là:

+ = 0: nếu hai chuỗi bằng nhau

+ < 0: nếu chuỗi $str1 nhỏ hơn chuỗi $str2

+ > 0: nếu chuỗi $str1 lớn hơn chuỗi $str2

Ví dụ 5.55: So sánh chuỗi

```
<?php
        echo strcmp('chao','Chao'); // kết quả 0
        echo strcmp('chao chu','chao anh'); // kết quả là 1
?>
```

- Hàm tìm một chuỗi trong 1 chuỗi **strstr($st1, $st2) và strchr($s1, $st2)**: Hàm trả về kết quả là một chuỗi con của chuỗi $st1 được lấy từ vị trí xuất hiện đầu tiên của chuỗi $st2 đến hết chuỗi $st1 nếu tìm thấy chuỗi $st2 trong chuỗi $st1, nếu không tìm thấy thì trả về giá trị FALSE.

Ví dụ 5.56: Tìm kiếm chuỗi trong chuỗi

```
<?php
        $st="ho die nloi@yahoo.com";
        echo strstr($st, "@")."</br>"; //@yahoo.com
        echo strchr($st, "#"); //
```

```
?>
```

- Hàm tìm vị trí chuỗi con **strpos($st1,$st2):** Hàm trả về vị trí chuỗi con đầu tiên của cuổi $st2 trong chuỗi $st1, nếu không tìm thấy thì kết quả trả về là FALSE

Ví dụ 5.57: Trả về vị trí đầu tiên của chuỗi trong chuỗi

```
<?php
        $st="hodienloi@yahoo.com";
        echo strpos($st,"yahoo"); // vị trí 10
        echo strpos($st,"gmail"); //
?>
```

- Hàm tìm kiếm và thay thế **str_replace($st1, $st2, $st)**: Hàm tìm kiếm chuỗi $st1 trong chuỗi $st, nếu tìm thấy thì thay thế chuỗi $st1 bằng chuỗi $st2 trong chuỗi $st.

Ví dụ 5.58: Tìm kiếm và thay thế yahoo bằng gmail, hotmail bằng gmail

```
<?php
$st="hodienloi@yahoo.com";
echo str_replace('yahoo', 'gmail', $st); //hodienloi@gmail.com
echo str_replace('hotmail', 'gmail', $st); //hodienloi@yahoo.com
?>
```

- Hàm **strrev($st):** Đảo ngược 1 xâu.

Ví dụ 5.59: Đảo xâu ký tự

```
<?php
echo strrev("Hello world!"); // kết quả trả về là "!dlrow olleH"
?>
```

- Hàm tách chuỗi **explode($ch,$st):** Hàm tách chuỗi $st thành nhiều phần tử con bằng cách chỉ định chuỗi tách $ch và gán từng chuỗi con vào các phần tử của mảng

Ví dụ 5.60: Tách chuỗi thành nhiều phần tử mảng

```
<?php
$st="Xin chào tất cả các bạn trong lớp";
$mang = explode(' ',$st);
for($i=0 ; $i<count($mang); $i++)
      echo "Từ thứ $i là: ".$mang[$i]."</br>";
?>
```

- Hàm kết hợp chuỗi **implode($ch, $mang)**: Hàm kết hợp các phần tử của mảng thành một chuỗi các phần tử khi ráp thành chuỗi sẽ cách nhau bằng một chỉ thị $ch.

Vi dụ 5.61: Nối nhiều phần mảng thành một chuỗi.

```
<?php
        $array = array('Xin', 'chào', 'tất', 'cả', 'các', 'bạn');
        $st = implode(" ", $array);
        echo $st; //Kết quả xuất ra là:  Xin chào tất cả các bạn
?>
```

- Hàm đổi số thành ký tự trong bảng mã ASCII chr():

Ví dụ 5.62: Chuyển mã thành ký tự trong bảng mã ASCII

```
<?php
for($i =15 ;$i<25; $i++)
echo chr($i).","; //kết quả là: □ , □ , □ , □ , □ , □ , □ , □ , □ , □ ,
?>
```

- Phép toán nối chuỗi (.)

Ví dụ 5.63: Nối hai chuỗi

```
<?php
        $ho_dem = 'Hồ Diên';
        $ten = 'Lợi';
        $ho_ten =$ho_dem .' '. $ten;
        echo $ho_ten; // Hồ Diên Lợi
?>
```

b. Kiểu dữ liệu số

- Hàm giá trị tuyệt đối: **abs**(x)

Ví dụ 5.64: Giá trị tuyệt đố của -5

```
<?php
echo abs(-5);
?>
```

- Hàm làm tròn: **round**(x[,i])

Ví dụ 5.65: Hàm làm tròn số

```
<?php
        $so = 1234.56789;
        echo round($so,2) // kết quả là 1234.57
?>
```

- Hàm làm lấy phân nguyên: **floor**(x)

Ví dụ 5.66: Hàm lấy phần nguyên

```
<?php
        $so = 123.678;
        echo floor($so); // kết quả của hàm là 123
?>
```

- Hàm e^x : **exp**(x)

Ví dụ 5.67: Hàm e^x

```
<?php
        echo exp(3);
?>
```

- Hàm lượng giác: sin(x), cos(x), tan(x) với x radian

Ví dụ 5.68: Tính sin, cos, tan của 30 độ

```
<?php
        define("PI",3.14);
        echo sin(PI*30/180); // kết quả  0.4997701026431
        echo cos(PI*30/180);// kết quả 0.86615809440546
        echo tan(PI*30/180); //Kết quả 0.57699640039287
?>
```

- Hàm căn bậc hai: **sqrt**(x)

Ví dụ 5.69:

```
<?php
        $so = 36;
        echo sqrt($so); // Kết quả là 6
?>
```

- Hàm ngẫu nhiên trong khoảng: **rand**(n1,n2)

Ví dụ 5.70:

```
<?php
```

```
        echo rand(10,20);
?>
```

- Hàm logaric: **log**(x)

Ví dụ 5.71:

```
<?php
        $so = 20;
        echo log($so); // kết quả 2.995732273554
?>
```

- Hàm mũ: pow(a, x)

Ví dụ 5.72:

```
<?php
        $so = 10;
        echo pow($so,2);
?>
```

- Hàm pi(): Hàm trả về giá trị là số pi, hàm không có tham số.

Ví dụ 5.73:

```
<?php
      $so = pi();
      echo $so*10; // 31.415926535898
?>
```

- Hàm **range(gt1,gt2):** Hàm lấy giá trị nguyên trong khoảng gt1… gt2

Ví dụ 5.74:

```
<?php
foreach (range(0, 12) as $number)
{
   echo $number;
}
echo "</br>";
foreach (range(0, 100, 10) as $number)
{
   echo $number;
}
?>
```

- Hàm **number_format():** Định dạng số

Ví dụ 5.75:

```
<?php
$number = 1234.56;
echo number_format($number)."</br>"; //1.235
echo number_format($number, 2, ',', ' ')."</br>"; //1 234,46
$number = 1234.5678;
echo number_format($number, 2, '.', '');//1234.57
?>
```

c. Kiểu dữ liệu ngày, giờ

- Hàm **checkdate()** kiểm tra ngày nhập vào có hợp lệ không?

Cú pháp: **checkdate** ($month , $day, $year)

Trong đó: Hàm trả về giá trị đúng hoặc sai.

$month, $day, $year: là kiểu dữ liệu số nguyên, là tháng ngày năm được nhập vào để kiểm tra.

Ví dụ 5.76:

```
<?php
	$day = 29;
	$month = 2;
	$year = 2010;
	$kq=checkdate($month, $day, $year);
	if($kq)
	{
		echo "Ngay thang hop le";
	}
	else
	{
		echo "Ngay thang khong hop le";
	}
?>
```

- Hàm **date()**

Cú pháp: **date ($format [,$timestamp])**

Trong đó:

Hàm trả về một chuỗi được qui định bởi chuỗi định dạng $format

Nếu $format =

D	Ngày có dạng hai chữ số 01- 31
D	Thứ trong tuần có 3 ký tự Mon - Sun
J	Ngày được có dạng 1-31
L/l	Thứ trong tuần được viết đầy đủ Monday - Sunday
N	Cho ra số thứ tự của ngày trong tuần (1-7)1: Monday – 7: Sunday
W	Cho ra số thứ tự ngày trong tuần 0: Monday – 6: Sunday
Z	Cho ra ngày thứ máy trong năm(0 - 365)
W	Cho ra số thứ tự tuần trong 1 năm
F	Cho ra tên tháng đầy đủ từ January tới December trong năm
M	Cho ra tháng từ 01 - 12
M	Cho ra tên tháng chỉ có 3 ký tự đầu từ Jan đến Dec
N	Cho ra tháng từ 1 - 12
Y	Cho ra năm có 4 chữ số 2009
Y	Cho ra năm có 2 chữ số 09
A	Dạng AM và PM
A	Dạng am và pm
G	Định dạng đồng hồ 12
H	Cho ra giờ 0 -23 h
H	Cho ra giờ 1- 12h
I	Cho ra phút (0-59)
S	Cho ra giây(0-59)

- Lấy giá trị của ngày hiện tại: getdate()

Hàm trả về mảng gồm 11 phần tử để lưu trữ các giá trị(seconds, minutes, hours, mday, wday, mon, year, yday, weekday, month, 0) của ngày tháng năm hiện tại.

Ví dụ:

```
<?php
    $today = getdate();
    print_r($today);
?>
```

- Lấy thời gian hiện tại: time()

- Chuyển chuỗi thành thời gian: strtotime()

- Kiểm tra và chuyển thời gian sang đơn vị giây: mktime()

- Định dạng thời gian thành só nguyên: idate()

IV. Biểu mẫu form

1. Đặc điểm form

Form là một thành phần của trang web. Chúng ta sẽ xây dựng form bằng cách thêm vào form nội dung và các đối tượng thể hiện (textField, Textarea, Button, RadioButton, CheckBox, List/Menu) sau đó định dạng chúng.

Các thuộc tính cơ bản của form:

- name: tên form
- action: hành động
- method: phương thức
- vị trí: _top, _parent, _self, _black

Chú ý: khi muốn lấy được giá trị trên form đưa về xử lý ở trang nào thì action sẽ chỉ ra trang đó.

Phương thức là cách thức lấy giá trị trên form. Chúng ta có hai phương thức sau: POST và GET

+ POST: chuyển giá trị trên form và để nhận được các giá trị này chúng ta dùng biến $_POST hoặc $_REQUEST.

+ GET: chuyển giá trị trên form và để nhận được các giá trị này chúng ta dùng biến $_GET hoặc $_REQUEST.

2. Biểu mẫu sử dụng phương thức $_POST

a. Đặc điểm

- Biến $_POST được dùng để lấy các giá trị trên form thông qua phương thức POST. Thông tin được gửi từ form với phương thức này không giới hạn dung lượng thông tin gửi đi.

- Thông tin được gửi bằng phương thức POST sẽ không hiện thị lên địa chỉ URL nên người dùng không thể thấy được.

b. Cách sử dụng

Cú pháp lấy giá trị của một đối tượng trên form sau khi form submit:

$_POST[‘tên điều kiển’];

Ví dụ: Tạo biểu mẫu nhập vào 2 số tính tính tổng và cho ra kết quả form

```
<?php
if(strlen($_POST["so_a"])&& strlen($_POST["so_b"]))
{
    $a = $_POST["so_a"];
```

```
        $b = $_POST["so_b"];
        $kq = $a + $b;
}
else
{
        $kq = "Bạn chưa nhập giá trị vào textfield";
}
?>
<form name="form1" method="post" action="vd.php">
  <table width="600" border="0" align="center" cellpadding="6" cellspacing="4">
    <tr>
      <td colspan="2" align="center" bgcolor="#FFCCFF"><span class="style1">TÍNH TỔNG CỦA HAI SỐ VÀ TRẢ GIÁ TRỊ LẠI FORM</span></td>
    </tr>
    <tr>
      <td width="142" align="right" valign="middle" bgcolor="#FFCCFF">Số A</td>
      <td width="422" bgcolor="#FFCCFF"><label>
        <input  type="text" name="so_a" value="<?php echo $_POST["so_a"]; ?>">
      </label></td>
    </tr>
    <tr>
      <td align="right" valign="middle" bgcolor="#FFCCFF">Số B </td>
      <td bgcolor="#FFCCFF"><label>
        <input  type="text" name="so_b" value="<?php echo $_POST["so_b"]; ?>">
      </label></td>
    </tr>
    <tr>
      <td align="right" valign="middle" bgcolor="#FFCCFF">Kết quả</td>
      <td bgcolor="#FFCCFF"><label>
        <input name="kq" type="text" id="kq" value="<?php echo $kq;?>" size="30">
      </label></td>
    </tr>
    <tr>
      <td align="right" valign="middle" bgcolor="#FFCCFF">&amp;nbsp;</td>
      <td bgcolor="#FFCCFF"><label>
        <input type="submit" name="button" id="button" value="Submit">
      </label></td>
    </tr>
```

```
  </table>
</form>
```

3. Biểu mẫu sử dụng phương thức $_GET

Lấy giá trị bằng phương thức

Chương 6: Hướng đối tượng trong PHP

I. Khái niệm

Một lớp được hiểu là một kiểu dữ liệu đặc biệt bao gồm các thuộc tính và các phương thức được định nghĩa từ trước. Đây là sự trừu tượng hóa của đối tượng. Một đối tượng sẽ được xác lập khi nó được thực tế hóa từ một lớp. Khác với kiểu dữ liệu thông thường, một lớp là một đơn vị (trừ tượng) bao gồm sự kết hợp giữa các phương thức và các thuộc tính.

II. Tạo lớp

Lớp là tập hợp các biến và các hàm – theo thuật ngữ của OOP, lớp là tập hợp các thuộc tính và các phương thức. Ngoài các lớp thư viện được xây dựng sẵn, chúng ta có thẻ tạo ra các lớp riêng còn gọi là lớp do người dùng tự định nghĩa.

Cấu trúc của một lớp bao gồm các từ khóa class, tên_lớp, dấu ngoặc {} để bao lại các câu lệnh bên trong lớp:

Cú pháp:

```
class <tên_lớp>
{
    // Khai báo các thuộc tính của lớp.
    // Gán và lấy giá trị của thuộc tính.
    // Tạo đối tượng của lớp.
}
```

Trong đó:

- Khai báo các thuộc tính của lớp

Thuộc tính là thành phần lưu trữ các tính chất, đặc điểm của đối tượng. Ứng với mỗi thuộc tính chúng ta khai báo một biến bắt đầu bằng từ kháo var để lưu trữ giá trị của chúng.

Cách khai báo:

```
var <tên_thuộc_tính_1>;
……..
```

Chú ý: Chúng ta cũng có thể khởi gán giá trị cho tên biến.

Ví dụ:

```
Class vidu_lop
{
  var $text;
  ……..
}
```

- Lấy các giá trị cho các thuộc tính

+ Gán giá trị cho thuộc tính

Một thuộc tính ban đầu cần phải có giá trị, do đó chúng ta cần phải gán giá trị cho thuộc tính.

Cú pháp:

```
function set_name ($giá_trị)
{
	$this -><$tên_thuộc_tính> =$giá_trị;
}
```

Ví dụ: Gán giá trị cho thuộc tính name.

```
function set_vidu($text)
{
	$this ->name =$text;
{
```

+ Lấy giá trị của thuộc tính.

Trước khi muốn dùng thuộc tính thì chúng ta cần phải lấy giá trị của thuộc tính.

Cú pháp:

```
function get_name()
{
	return $this -> <tên_thuộc_tính>;
}
```

Ví dụ: Lấy giá trị của thuộc tính

```
function get_vidu()
{
	return $this ->name;
{
```

Chú ý: Trong lớp chúng ta có thể truy cập thông qua con trỏ $this. Con trỏ lớp dùng để chỉ lớp hiện tại đang làm việc.

III. Sử dụng lớp

Sau khi xây dựng xong lớp chúng ta có thế sử dụng lớp: trước tiên cần khởi tạo đối tượng, sau đó gán giá trị cho các thuộc tính và gọi sử dụng các phương thức của lớp.

- Tạo đối tượng:

Để khai báo một đối tượng chúng ta dùng từ khóa new tiếp đó là tên lớp.

Cú pháp:

$name_opp = new <tên_class>();

Ví dụ: Tạo đối tượng

```
$opp = new vidu_lop();
```

- Gán giá trị cho thuộc tính lớp.

$name_opp ->set_name(<giá trị>);

Ví dụ: Gán giá trị cho

```
$opp->set_vidu("Chào bạn");
```

- Sử dụng các phương thức lớp

Để sử dụng phương thức chúng ta chỉ cần gọi tên phương thức với các giá trị truyền vào cho tham số nếu có.

Cú pháp:

$name_opp ->get_name(tham số nếu có);

Ví dụ: Sử dụng phương thức

```
$opp->get_vidu();
```

Ví dụ: Tính tổng hai số

```
<?php
    class phep_tinh
    {
        var $so1;
        var $so2;
        // so thu nhat
            // Lay gia tri cua so thu nhat
            function lay_gt_1()
            {
                return $this->a;
            }
            // Gan gia tri cho so thu nhat
            function gan_gt_1($so1)
            {
                $this->a = $so1;
            }
        // so thu hai
            // Lay gia tri cua so thu hai
```

```
                function lay_gt_2()
                {
                        return $this->b;
                }
                // Gan gia tri cho so thu hai
                function gan_gt_2($so2)
                {
                        $this->b = $so1;
                }
        // Phuong thuc
                function tong()
                {
                        return $this->a + $this->b;
                }
    }
    $tinh = new phep_tinh();
    $tinh->gan_gt_1(6);
    $tinh->gan_gt_1(8);
    echo "a+b =".$tinh->tong();

?>
```

IV. Kế thừa

1. Khái niệm kế thừa

2. Chồng hàm

Chương 7: Tạo web động

I. Sử dụng tập tin dùng chung

Để sử dụng các đoạn mã bên ngoài, chúng ta có thể sử dụng khai báo tiền xử lý include và require. Cho phép chúng ta xây dựng các hàm các hằng số, và bất kỳ đoạn mã nào sau đó có thể chèn vào các đoạn kịch bản.

Require khác include là, nó có thể làm thay đổi nội dung của trang hiện tại khi biên dịch, các trang này dùng để khai báo các biến, các hằng số hay các đoạn mã đơn giản không có vòng lặp. Khi đó include cho phép thực hiện các câu lệnh phức tạp – có câu lệnh tạo chu trình. Nó chỉ sử dụng các hàm như những hàm ngoài của chương trình.

Ví dụ:

<table>
<tr><td colspan="3">Header</td></tr>
<tr><td colspan="3">Menu_main</td></tr>
<tr><td>Memu_left</td><td>Content</td><td>Hot information</td></tr>
<tr><td colspan="3">Fooder</td></tr>
</table>

1. REQUIRE

a. Cách sử dụng

Đối với phương thức require(), tất cả nội dung bên trong file được chèn vào sẽ được biên dịch.

Khi chúng ta muốn sử dụng đoạn chương trình đã được viết sẵn ở vị trí nào trong trang thì chúng ta chỉ cần dùng require() để chèn file chứa đoạn chương trình này ở vị trí đó.

Cú pháp:

require("tên và đường dẫn của tập tin");

Ví dụ:

Đoạn chương trình dưới đây được viết ở trang chao.php

```
<?php
```

```
    echo "Hello my class";
?>
```

Còn đoạn chương trình dưới được viết ở trong home.php

```
<?php
    echo "Đây là đoạn chương trình hướng dẫn học PHP và MYSQL<br>";
    require("chao.php");
    echo "<br>Đã đến với chương trình này";
?>
```

b. Các tập tin đương dùng trong require()

PHP không quy định cách đặt tên tập tin được chèn vào bằng require(). Vì vậy, chúng ta có thể đặt tên tập tin tùy ý. Khi chúng ta dùng require() để chèn tập tin này vào, nội dung của tập tin sẽ trở thành một phần trong trang web.

Thông thường, các câu lệnh PHP đopch viết trong các tập tin .html sẽ không thực hiện được. Chúng sẽ thực hiện được khi chúng được viết dưới trong các tập tin .php. Vì vậy, khi viết các câu lệnh PHP trong các tập tin được dùng để chèn này, ta nên chọn kiểu tập tin thích hợp như . inc hay . php để các lệnh này có thể biên dịch được khi thực thi.

c. Thẻ PHP và require()

Các lệnh PHP cần phải được bao bởi cặp thẻ PHP **<?php** và **?>**. Nếu chúng ta không có thẻ PHP **<?php** và **?>** khi viết lệnh PHP thì các lệnh PHP này sẽ trở thành dạng văn bản hoặc HTML và không thể thực thi được.

d. Dùng require() cho các template

Nếu ứng dụng web của chúng ta có cùng một mẫu thiết kế nhưng chỉ khác nhau về nội dung bên trong thì chúng ta sẽ tạo ra một mẫu template và chỉ cần khai báo các biến trình bày dữ liệu bên trong template.

Khi template này được thực thi, tất cả các biến này sẽ có giá trị và trình bày như một định dạng template được sử dụng nhiều lần.

Ví dụ: Thông thường trong ứng dụng web phần header và fooder thường được hiện thi ở hầu hết các trang, vì vậy ta sẽ tạo ra trang header.inc và fooder.inc để chứa định dạng và nội dung phần cuối trang. Sau đó, ở trang nào của ứng dụng có sử dụng header và fooder thì chúng ta sẽ chèn hai trang này vào.

Điểm đặc biệt và quan trọng nhất của kết hợp này là sau khi chúng ta đã thực hiện nhiều trang có chèn các trang header.inc và fooder.inc, chúng ta có thể dễ dàng thay đổi trang

header.inc và fooder.inc này. Và chỉ cần lưu lại các thay đổi thì tất cả các trang đã chèn các trang này sẽ tự động thay đổi.

2. INCLUDE

a. Cách sử dụng

include() cũng có cách sử dụng tương tự require(). Tuy nhiên, chúng có một điểm khác nhau cần phải lưu ý đó là khi nội dung bị lỗi thì dùng require() sẽ xuất hiện thông báo lỗi, trong khi đó dùng include() sẽ chỉ xuất hiện cảnh báo.

Trong những tập tin có dùng require() thì ta không nên sử dụng các câu trúc điều khiển vì sẽ không hiệu quả.

Cú pháp:

include("tên tập tin và đường dẫn đến tập tin");

Ví dụ: Trang tính toán có nội dung như sau:

```
<?php
if($a ==1)
{
	require("tinh_tong.php");
}
else
{
	require("tinh_tong.php");
}
?>
```

Trong ví dụ này khi thực thi trang tinh_toan.php, nội dung bên trong cả hai tập tin là tinh_tong.php và tinh_hieu.php đều được biên dịch, trong khi đó chỉ có một trong hai trường hợp đúng và chỉ cần biên dịch một tập tin khi thỏa mãn điều kiện đúng là đủ. Như vậy, dùng require() trong trường hợp này sẽ không còn thích hợp, thay vào đó chúng ta dùng include().

```
<?php
if($a ==1)
{
	include("tinh_tong.php");
}
else
{
	include("tinh_tong.php");
}
```

```
?>
```

b. require_once() và include_once()

Hàm require_once() và include_once() là hai dạng biến đổi của hàm require() và include().

Mục đích của hai hàm này trở nên hữu ích khi chúng ta bắt đầu sử dụng chúng để chèn các thư viện và các hàm vào, sử dụng các cấu trúc này giúp chúng ta tránh được việc chèn cùng một hàm hay thư viện lần thứ hai bởi khi hàm khai báo lại một hàm đã được xây dựng sẽ được phát sinh lỗi.

Việc chèn vào hai lần cùng một tập tin thường xảy ra khi xây dựng các ứng dụng lớn, khi nhiều tập tin thư viện khác nhau được chèn vào trong cùng một require_once() hay include_once() thì trong lần đầu tiên cách thức hoạt động của nó cũng giống như require() hay include(). Tuy nhiên, require_once() và include_once() nếu được gọi để chèn tập tin đã được chèn thì file này sẽ không chèn được chèn vào nữa. Hàm này là một công cụ thông minh cho việc tạo ra các thư viện dùng lại.

c. Đường dẫn của file được chèn

Sử dụng các hàm đã được giới thiệu ở trên để truy cập các thư viện có thể làm tăng tính mềm dẻo của ứng dụng. Tuy nhiên, vẫn còn có một vấn đề xảy ra.

d. Ví dụ

II. Mở tập tin và thư mục

1. Tập tin

a. Chế độ mở tập tin

Để mở một tập tin, chúng ta cần xác định chế độ mở. Có 3 tùy chọn cho chế độ mở file:

- Mở file ở chế độ read only, write only hay cả read và write.
- Mở file đã tồn tại: chúng ta có thể ghi đè hay ghi thêm vào nội dung đang có của file. Trong trường hợp ghi thêm vào nội dung đã có của file, có hai cách ghi là ghi vào đầu tập tin và ghi vào cuối của tập tin.
- Khi muôn ghi file hệ thống thì chúng ta cần chỉ định chế độ ghi file là nhị phân hoặc text.

Phân loại các chế độ mở file:

Chế độ	Mô tả
r	Chỉ đọc file, bắt đầu đọc đầu file
r+	Đọc và ghi file: Bắt đầu từ đầu file
w	Chỉ ghi file. Mở và xóa toàn bộ nội dung của file đã có hoặc tạo ra một file mới nếu file đó không tồn tại, sau đó ghi nội dung vào file.

w+	Đọc và ghi. Mở và xóa toàn bộ nội dung của file đã có hoặc tạo ra một file mới nếu file đó không tồn tại, sau đó ghi nội dung vào file
a	Chỉ ghi file. Mở và ghi nội dung vào cuối file hoặc tạo ra một file mới nếu file không tồn tại.
a+	Ghi và đọc dữ liệu. Mở và ghi nội dung vào cuối file hoặc tạo ra một file mới nếu file không tồn tại.
x	Tạo và mở file để ghi. Tạo ra một file mới và ghi nội dung vào file. Nếu file đã tồn tại, trả về giá trị FALSE và thông báo lỗi.
x+	Tạo và mở file để đọc và ghi. Tạo ra một file mới và ghi nội dung vào file. Nếu file đã tồn tại, trả về giá trị FALSE và thông báo lỗi.

b. Mở tập tin

Để mở tập tin chúng ta sử dụng hàm fopen().

Cú pháp:

fopen(<tên_tập_tin>,<chế độ mở>)

Ví dụ: Mở file vidu.txt trong thư mục vidu_web, với chế độ mở chỉ đọc.

```
<?php
$f = fopen("vidu.txt", "r");
?>
```

c. Đọc tập tin

- Kiểm tra kết thúc tập tin

Để kiểm tra trạng thái kết thúc tập tin hay chưa chúng ta sử dụng hàm feof()

Cú pháp:

feof($f)

Trong đó $f là biến khai báo nhận giá trị trả về của hàm fopen(). Hàm trả về kết quả là TRUE nếu con trỏ ở cuối file.

Ví dụ:

```
<?php
$f = fopen("vidu.txt", "r");
if(feof($f))
{
    echo "Đã kết thúc tập tin";
}
?>
```

- Duyệt và đọc từng dòng nội dung trong tập tin

Khi mở file chúng ta có thể đọc nội dung đã được mở theo từng dòng bằng hàm **fgets().** Hàm sẽ trả về là chuỗi có độ dài xác định, mặc định độ dài là 1024.

Cú pháp:

fgets($f [, int độ dài])

Ví dụ:

```
<?php
$f = fopen("vidu.txt", "r");
while(!feof($f))
{
	$noidung =fgets($f);
	echo $noidung. "</br>";
}
fclose($f);
?>
```

- Duyệt và đọc từ ký tự trong tập tin

Để đọc nội dung tập tin theo từng ký tự 1 ta sử dụng hàm **fgetc()**

Cú pháp:

fgetc($f)

Ví dụ:

```
<?php
$f = fopen("vidu.txt", "r");
while(!feof($f))
{
	$noidung =fgetc($f);
	echo $noidung;
}
fclose($f);
?>
```

- Đọc toàn bộ nội dung tập tin

Để đọc toàn bộ nội dung tập tin ta sử dụng hàm **readfile().**

Cú pháp:

readfile(<đường dẫn và tên tập tin>);

```
<?php
$f = "vidu.txt";
echo readfile($f);
?>
```

d. Định dạng tập tin.

Trước khi ghi chuỗi vào file, chúng ta cần phải định dạng lại chuỗi đó theo nhu cầu xuất dữ liệu trở lại khi đọc file.

Cách thức định dạng là do chúng ta tự thiết lập. Tuy nhiên, có một số định dạng được quy định sẵn như sau:

+ \t : nhảy tab

+ \n : xuống dòng

e. Ghi nội dung tập tin

Để ghi nội dung vào tập tin ta sử dụng hàm fwrite()

Cú pháp:

fwrite(<tập tin>, <nội dung [, <độ dài>]>);

Ví dụ:

f. Đóng tập tin.

Để đóng tập tin đã mở ta sử dụng hàm fclose()

Cú pháp:

fclose($f)

Trong đó $f là tên biến nhận giá trị trả về của hàm fopen()

g. Kiểm tra sự tồn tại của tập tin

Để mở file ta thường gặp hai trạng thái, file đó đã có hoặc file đó không tồn tại.

Để kiểm tra sự tồn tại của file ta sử dụng hàm file_exists() hoặc is_file()

Cú pháp:

file_exists(<tập tin>)

Trong đó: Hàm trả về kết quả TRUE hoặc FALSE, nếu tồn tại trả về giá trị là TRUE, ngược lại trả về giá trị FALSE

Ví dụ:

```
<?php
```

```
$f = "vidu.txt";
if (!file_exists($f))
{
	echo "không ton tai file";
	exit;
} else {
	echo readfile($f);
}
?>
```

h. Kiểm tra kích thước file

Để kiểm tra kích thước file ta sử dụng hàm filesize()

Cú pháp:

filesize(<đường dẫn và tên của file>);

Vi dụ: Kiểm tra kích thước tập tin

```
<?php
$f = "vidu.txt";
echo "Size :".filesize($f);
?>
```

k. Xóa tập tin

Để xóa tập tin ta sử dụng hàm unlink()

Cú pháp:

unlink(<tập tin>)

Ví dụ: Xóa tập tin vidu.txt

```
<?php
$f = "vidu.txt";
if (!unlink($f))
{
	echo "Khong xoa dc tap tin $f";
} else {
	echo "da xoa tap tin $f";
}
?>
```

2. Thư mục

a. Tạo thư mục

Để tạo thư mục ta sử dụng hàm mkdir()

Cú pháp:

mkdir(<tên thư mục>);

Ví dụ: Tạo thư mục

```
<?php
mkdir("DIENLOI");
?>
```

b. Kiểm tra thư mục

Để kiểm tra thư mục ta sử dụng hàm is_dir(), trả về TRUE nếu tồn tại thư mục, ngược lại là FASLE.

Cú pháp:

is_dir(<thư mục>)

Ví dụ: Kiểm tra sự tồn tại của thư mục

```
<?php
if(!is_dir("DIENLOI"))
{
	echo "Khong ton tai thu muc";
}
else
{
	echo "Ton tai thu muc";
}
?>
```

c. Mở thư mục

Để mở thư mục ta sử dụng hàm opendir(). Kết quả trả về sẽ là nguồn(chứa các thư mục và tập tin) của thư mục nếu thư mục mở thành công, ngược lại trả về giá trị False.

Cú pháp:

opendir(<tên thư mục>)

Ví dụ: Mở thư mục

```
<?php
```

```
$dir = opendir("Vidu_php");
?>
```

d. Đóng thư mục

Khi đã dùng xong thư mục, cần phải đóng thư mục bằng hàm closedir()

Cú pháp:

closedir(<tên thư mục>)

Ví dụ: Đóng thư mục

```
<?php
closedir("Vidu_php");
?>
```

e. Duyệt thư mục

Chúng ta duyệt thư mục bằng cách sử dụng vào lặp kết hợp với hàm readdir()

Cú pháp:

readdir(<tên thư mục>)

Ví dụ:

```
<?php
$dir =opendir("Vidu_php");
while(($file = readdir($dir))==true)
{
 echo $file."</br>";
}
closedir($dir)
?>
```

III. Upload tập tin lên server.

1. Giới thiệu

Trong hầu hết các ứng dụng web, thông thường người dùng có thể upload file lên server. Đối với ứng dụng web xây dựng bằng ngôn ngữ PHP chúng ta cũng có thể xây dựng cho người dùng upload file lên server.

2. Các bước upload file

Bưới 1: Tạo form upload file:

```
<form action ="upload_file.php" method ="POST" enctype ="multipart/form-data">
<label for ="file"> Tên file </label>
```

```
<input type ="file" name ="file_upload" id ="file" />
</br>
<input type ="submit" name ="submit" value ="Upload file" />
</form>
```

Chú ý:

+ Đối với form để upload file thì trên thẻ form chúng ta bổ sung thêm thuộc tính enctype ="multipart/form-data".

+ Method được sử dụng theo phương thức POST

+ Nếu muốn quyết định kích cỡ tối đa của tập tin upload thì trong thẻ input FileField upload chúng ta bổ sung thêm thuộc tính value = "kích thước tối đa" – đơn vị tính là byte, lúc này thuộc tính name của file field có giá trị là "MAX_FILE_SIZE".

Bước 2: Viết code thực hiện việc upload file

<?php

```
if($_FILES["file_upload"]["error"]>0)
{
  echo "Lỗi của file ".$_FILES["file_upload"]["error"]."</br>";
}
else
{
      echo "Upload:".$_FILES["file_upload"]["name"]."</br>";
      echo "Type:".$_FILES["file_upload"]["type"]."</br>";
      echo "Upload:".($_FILES["file_upload"]["size"]/1024)."Kb</br>";
      echo "Temp file: ".$_FILES["tmp_file"]."</br>";
      if (file_exists("upload/".$_FILES["file_upload"]["name"]))
      {
            echo $_FILES["file_upload"]["name"]."Đa ton tai";
      }
      else
      {
      move_uploaded_file($_FILES["file_upload"]["tmp_name"],"Upload/".$_FILES["file_
upload"]["name"]);
            echo "Lưu trữ "."Upload/".$_FILES["file_upload"]["name"];
      }
}
?>
```

Bằng cách sử dụng biến $_FILES[tên_đk_field] [tên thuộc tính] ta có thể upload file từ máy client sang server.

Trong đó, tham số thứ nhất là tên điều khiển filefield trên form, tham số thứ hai là một trong những thuộc tính như: name(tên file), type(loại file), size(kích thước byte – tính theo byte), tmp_name(tên tạm của file), error(lỗi)…

Với đoạn code trên, trước tiên ta kiểm tra file upload có bị lỗi hay không, nếu có lỗi thì thông báo lỗi, ngược lại in ra các thuộc tính của file upload. Sau đó kiểm tra xem trong thư mục upload đã có file này hay chưa, nếu đã có thì thông báo, ngược lại thì sử dụng hàm move_uploaded_file(file_tạm, nơi lưu trữ upload) để di chuyển thư mục về thư mục lưu trữ file trên server.

IV. PHP Cookies

1. Khái niệm

Cookie được sử dụng để xác định thông tin của người dùng. Cookie là một file nhỏ được server lưu trữ xuống từng máy tính của người dùng. Mỗi khi máy tính này yêu cầu một trang tới trình duyệt, nó cũng sẽ gửi theo cookie. Với PHP ta có thể tạo ra và sử dụng giá trị của biến cookie.

2. Khai báo cookie

Dùng hàm setcookie() để khai báo một biến cookie.

Chú ý: Hàm này đặt trên thẻ <html>

Cú pháp:

setcookie(name, value, expire [,path, domain]);

Trong đó:

+ Name: tên biến cookie

+ value : giá trị

+ expire: thời gian giới hạn dành cho cookie – đơn vị tính là giây. Nếu thời gian không được thiết lập trong hàm setcookie(), biến cookie sẽ còn hiệu lực cho đến khi người dùng xóa tập tin cookie.

+ path: đường dẫn

+ domain: tên miền của website.

Ví dụ:

```
setcookie("username","ho die n loi", time()+3600);
```

Chú ý: Giá trị của biến sẽ được tự động mã hóa khi gửi cookie đi, tự động giải mã khi nhận cookie về .

3. Sử dụng cookie

Khi người dùng muốn sử dụng biến cookie đã đăng ký chúng ta dùng biến $_COOKIE để đọc giá trị biến cookie.

Cú pháp:

$giá_trị = $_COOKIE[“tên biến cookie”];

Ví dụ:

```
$username = $_COOKIE["username"]; // ho dien loi
```

4. Hủy cookie

Khi hủy biến cookie chúng ta cần kiểm tra lại thời gian giới hạn dành cho biến cookie này(được thiết lập khi chúng ta tạo ra biến cookie). Sau đó chúng ta sử dụng hàm setcookie() để hủy bằng cách đặt gia trị cho biến cookie bằng "" và thời gian =- thời gian giới hạn.

Cú pháp:

setcookie(name, "" , time() – thời gian giới hạn);

Ví dụ:

```
setcookie("username","", time()-3600);
```

V. PHP Sessions

1. Khái niệm

Khi làm việc với 1 ứng dụng, chúng ta sẽ mở ứng dụng lên, làm việc và đóng ứng dụng lại. Máy tính sẽ biết được ta là ai, biết chúng ta mở ứng dụng và đóng ứng dụng lại. Nhưng trên trình duyệt web không biết ta là ai và chúng ta làm gì bởi vì giao thức HTTP không duy trì trạng thái.

Session giải quyết vấn đề này bằng cách cho phép chúng ta lưu trữ thông tin người dùng trên trình duyệt web. Tuy nhiên thông tin sesion chỉ tạm thời và sẽ bị xóa sau khi người dùng rời khỏi ứng dụng web. Nếu cần, chúng ta có thể lưu trữ trong CSDL.

2. Cách thức hoạt động

Session làm việc bằng cách tạo ra một địa chỉ duy nhất (UID) cho mỗi người sử dụng. UID có giá trị là một chuỗi số ngẫu nhiên. UID có thể được lưu trữ trong cookie hoặc được truyền lên URL.

Ngoài UID, bạn có thể khởi tạo và sử dụng một số biến session do người dùng khai báo, tất cả các session này có giá trị cho mỗi người sử dụng khi họ truy cập đến ứng dụng web.

3. Khởi động Session

Trước khi lưu trữ thông tin người dùng vào session, chúng ta phải khởi động session.

Chú ý: Hàm khởi động session phải đặt phía trên thẻ <html>

Cú pháp:

```
session_start();
```

hàm này sẽ đăng ký session với trình duyệt, cho phép chúng ta bắt đầu lưu trữ thông tin người dùng và đặt UID.

4. Đặt ký Session

Chúng ta dùng biến $_SESSION nhận và lưu trữ giá trị của biến session.

Cú pháp:

```
$_SESSION["ten_bien_session"] ="giá tri";
```

Ví dụ:

5. Sử dụng Session

Khi muốn sử dụng các biến session hoặc giá trị lưu trữ trong biến session đã đăng ký chúng ta cũng dùng biến $_SESSION để đọc giá trị biến session.

Cú pháp:

$gia_tri=$_SESSION["ten_bien_session"];

Ví dụ: Tạo form đăng nhập thành tài khoản thành viên

6. Hủy biến Session

a. Hủy toàn bộ các biến session

Khi chúng ta không dùng đến các biến session nữa chúng ta có thể hủy toàn bộ các biến session mà chúng ta đã đăng ký bằng cách sử dụng hàm session_destroy()

Cú pháp:

```
session_destroy();
```

b. Hủy một biến session

Khi chúng ta không cần dùng biến session nào thì chúng ta có thể dùng hàm unset() để hủy bỏ biến session đó.

Cú pháp:

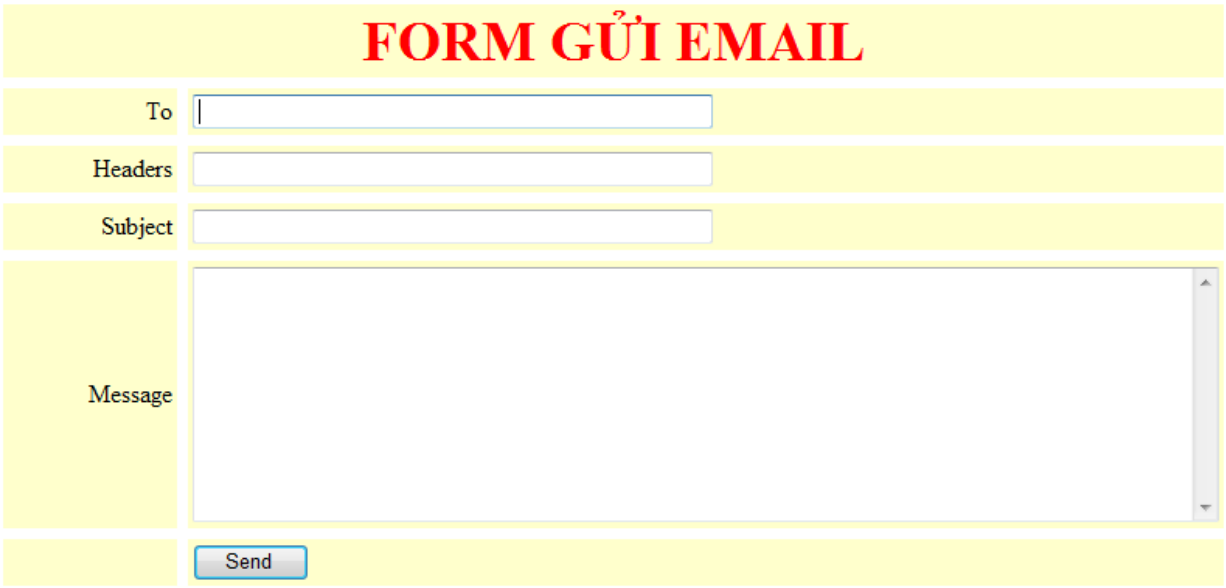
unset($_SESSION[“tên biến session”]);

VI. Gửi E-mail trong PHP

PHP cho phép người dùng tạo ra form gửi mail đến địa chỉ mail xác định. Để gửi mail sử dụng hàm mail(to,subject,message,headers,parameters với các tham số sau:

To:	Địa chỉ người nhận
Subject:	Tiêu đề của nội dung thư
Message:	Nội dung thư
Headers	Tiêu đề bổ sung: Người gửi email
Parameters:	

Ví dụ: Form Mail



Ví dụ: Lấy thông tin từ Form

```
<?php
$to = $_POST['to'];
$headers =$_POST['headers'];
$subject=$_POST['subject'];
$message =$_POST['message'];
$headers ="Form: $headers";
mail($to,$subject,$message,$headers);
?>
```

Chương 8: CƠ SỞ DỮ LIỆU MYSQL

I. Tổng quan

1. Giới thiệu CSDL

a. Khái niệm

CDSL là một tập hợp dữ liệu được lưu trữ một cách có tổ chức nhằm giúp việc xem, tìm kiếm và lấy thông tin được nhanh chóng và chính xác, giúp giảm công sức và thời gian quả lý thông tin cần thiết.

b. Chức năng.

- Lưu trữ

Dữ liệu được lưu trữ trên đĩa và người dùng có thể chuyển đổi dữ liệu từ CSDL này sang CSDL khác.

Tùy theo quy mô của ứng dụng mà chúng ta có thể chọn CSDL lớn hay nhỏ. Nếu quy mô nhỏ thì chúng ta chọn Access, MySQL, … nếu quy mô lớn thì chúng ta có thể chọn SQL Server, Oracle, DB2,…

- Truy cập

Tùy thuộc vào mục đích và yêu cầu của người sử dụng mà có những mức độ truy cập khác nhau: cục bộ, chia sẽ, truy cập dữ liệu giữa các CSDL khác nhau.

- Tổ chức

Cách tổ chức CSDL, tùy thuộc vào mô hình CSDL, cách phân tích và thiết kế CSDL và các đặc điểm riêng của từng ứng dụng.

- Xử lý

Xử lý dữ liệu là việc sử dụng các truy vấn cùng các phép toán để truy xuất các kết quả theo yêu cầu của người dùng.

c. Các loại CSDL

- CSDL phân cấp(Hierachical Database)

CSDL phân cấp có cấu trúc cây, dữ liệu được tổ chức dưới dạng tập tin trên đĩa.

Ưu điểm: Tốc độ truy cập nhanh vì chúng có quan hệ trức tiếp với nhau.

Khuyết điểm: không dùng cho các ứng dụng có quan hệ phức tạp.

- CSDL hướng đối tượng(Object Oriented Database)

CSDL hướng đối tượng là CSDL mà trong đó một bảng dữ liệu có thể được khai báo như một field của bảng dữ liệu khác.

- CSDL quan hệ(Relation Database)

CSDL quan hệ là CSDL mà các bảng dữ liệu có quan hệ với các bảng khác thông qua các mối quan hệ.

d. Các đối tượng chính của CSDL

Tuy có rất nhiều CSDL khác nhau nhưng trong môn học này chúng ta chỉ tìm hiểu về CSDL quan hệ.

- Bảng dữ liệu(table)

Bảng dữ liệu là thành phần trung tâm của CSDL, được dùng để lưu trữ thông tin của CSDL.

Cách thiết kế các bảng dữ liệu có vai trò rất quan trọng vì nó quyết định tính hiệu quả trong việc lưu trữ thông tin.

Trong một CSDL có nhiều bảng, mỗi bảng dùng để lưu trữ một nhóm thông tin khác nhau.

Cấu trúc của bảng dữ liệu gồm hai thành phần dòng và cột

Cột: là một khối dữ liệu trong bảng, có cùng loại dữ liệu.

Mỗi cột có các thông tin chính sau:

+ Tên cột: dùng để phân biệt với các cột khác trong bảng, do vậy nó có tính duy nhất, tên cột không dùng các ký tự đặc biệt

+ Kiểu dữ liệu của cột: xác định loại kiểu dữ liệu nào được phép lưu trữ trong cột

Dòng: là tập hợp các thông tin của tất cả các cột trong bảng

- Quan hệ(relation)

Là thành phần được dùng để tạo mối liên kết giữa các bảng dữ liệu với nhau nhằm đảm bảo tính nhất quán, đúng đắn của dữ liệu trong CSDL.

e. Hệ quản trị CSDL

Hầu hết các CSDL đều dựa vào một hệ quản trị CSDL để quản lý các dữ liệu được lưu trữ bên trong các CSDL đó và làm cho CSDL dễ dàng đến được với người dùng khi cần truy cập các thông tin khác nhau.

Một hệ quản trị CSDL tối thiểu phải có khả năng lưu trữ dữ liệu và cho phép dữ liệu có thể trao đổi với các CSDL khác.

Tuy nhiên, hầu hết các hệ quản trị CSDL có nhiều tính năng hơn:

- Quản lý dữ liệu.

- Duy trì bảo vệ

- Duy trì dữ liệu

- Quản lý các giao dịch.

…

f. SQL (Structure Query Language)

SQL là một ngôn ngữ cho phép thực hiện các thao tác rút trích, tính toán, cập nhật trên các dữ liệu được lưu trữ trong CSDL.

2. CSDL MySQL

a. Giới thiệu

CSDL MySQL là tập hợp các đối tượng: bảng, bảng ảo … cho phép người dùng lưu trữ và xuất các thông tin đã được tổ chức và lưu trữ bên trong đó.

b. Đặc điểm

- MySQL được sử dụng cho các ứng dụng web có quy mô vừa và nhỏ.

- Người dùng có thể sử dụng giao diện đồ họa hay dùng dòng lệnh để thực hiện các thao tác trên CSDL.

c. Các tập tin vật lý lưu trữ CSDL

Mỗi bảng trong CSDL được tạo ra sẽ được lưu trữ dưới 3 tập tin vật lý:

- .frm: lưu cấu trúc của bảng

- .MYD: lưu nội dung của bảng

- .MYI: lưu chỉ mục của bảng.

Các tập tin dữ liệu này sẽ được tự động lưu trữ trong thư mục:

Wamp\mysql\data\tên_CSDL.

d. Quy tắc đặt tên cho CSDL, bảng, chỉ mục, cột và định danh

- Chiều dài của tên

Loại	Chiều dài tối đa (byte)	Chiều dài tối đa (ký tự không dấu)
Database	64	64
Table	64	64
Index	64	64
Column	64	64
Alias	255	255

- Quy tắc đặt tên

+ Tên không kết thúc bằng khoảng trắng.

+ Tên CSDL không có các ký tự '/', '\', '.', hoặc các ký tự không cho phép khi đặt tên cho thư mục (\, /, :, *, ", <, >)

+ Tên bảng không có các ký tự '/', '\', '.', hoặc các ký tự không cho phép khi đặt tên cho tập tin (\, /, :, *, ", <, >, |)

+ Chiều dài của tên tối đa là 64 ký tự không dấu. Nếu chúng ta sử dụng các ký tự đa byte thì chiều dài sẽ dựa trên tổng số byte của tất cả các ký tự được dùng.

e. Tạo CSDL

Có hai cách để tạo một CSDL là dùng giao diện đồ họa hoặc dùng dòng lệnh.

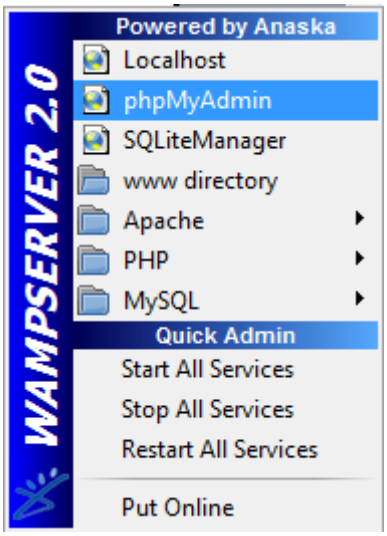
Các thuộc tính của CSDL

+ Tên CSDL: phải duy nhất trong hệ quản trị CSDL.

+ Vị trí lưu trữ: khi tạo mới một CSDL hệ thống sẽ tự động tạo ra một thư mục có tên là tên CSDL và được lưu trữ tại thư mục wamp\mysql\data\

Ví dụ: CSDL khoacntt

Bước 1: khởi động phpMyadmin



Bước 2: Trong màn hình giao diện đồ họa, nhập tên CSDL vào create new database và chọn các thông tin khác (nếu cần)

Bước 3: Nhấn Create để hoàn thành việc tạo CSDL.

- Sử dụng câu lệnh SQL

Cú pháp lệnh SQL tạo CSDL:

```
CREATE DATABSE name_database
[[DEFAULT] CHARACTER SET <character set name>]
[[DEFAULT] COLLATE <collation name>]
```

Trong đó:

+ CHARACTER SET : xác định bộ ký tự mặc định cho CSDL mới

+ COLLATE: Xác định bộ collation

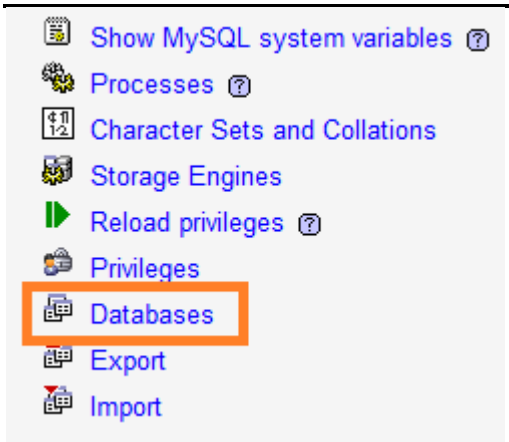
+ Character set name: Tên của một bộ mã bao gồm các ký tự, ký tự số, và biểu tượng để lưu trữ thông tin trong CSDL.

+ Collation name: tên một bộ mã tùy theo từng khu vực dựa trên bộ mã chuẩn character set name.

Cách thực hiện:

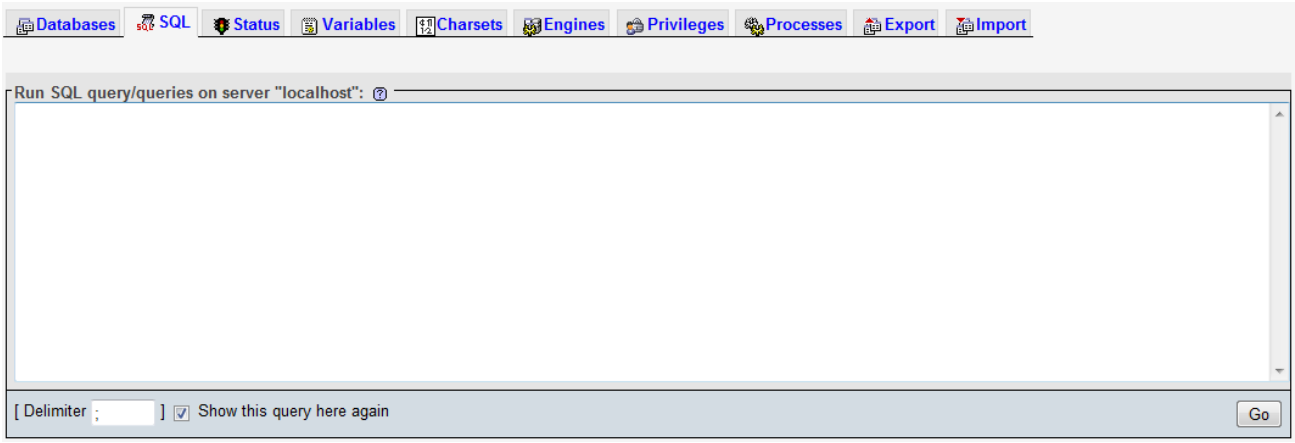
Bước 1: Khởi động phpMyAdmin

Bước 2: Chọn database



Bước 3: Chọn SQL

Bước 4: Viết lệnh SQL



Bước 5: Nhấn Go để kết thúc việc tạo CSDL

Ví dụ: Tạo CSDL khoacntt

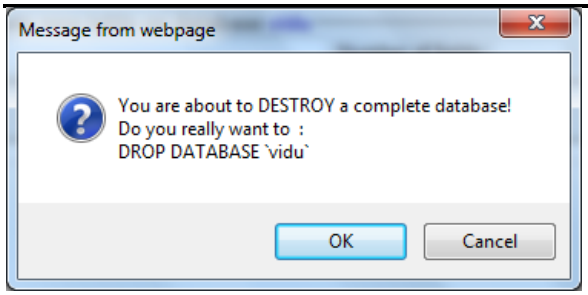
```
CREATE DATABASE `khoacntt1` ;
```

f. Xóa CSDL

- Xóa bằng giao diện đồ họa

Bước 1: Chọn CSDL cần xóa - nhấn Drop

Bước 2: Xác định lại việc xóa CSDL, sau đó chọn OK để xóa



- Sử dụng lệnh SQL

Cú pháp lệnh:

```
DROP DATABASE name_database
```

Ví dụ: Xóa CSDL khoacntt

DROP DATABASE `khoacntt`

II. Bảng(Table)

1. Khái niệm

Bảng trong MySQL dùng để lưu trữ thông tin của những đối tượng, thực thể trong thế giới thực muốn được lưu trữ vào trong máy tính.

Các thông tin trong bảng sẽ được tổ chức theo dạng dòng và cột.

2. Thuộc tính

a. Tên bảng

Tên bảng do người dùng đặt tên, tên bảng phải duy nhất trong một CSDL.

b. Các thuộc tính của cột trong bảng

- Tên cột: do người dùng đặt và tên cột là duy nhất trong bảng.

- Kiểu dữ liệu: Xác định kiểu dữ liệu lưu trữ trong cột, có các kiểu dữ liệu sau:

Kiểu số nguyên:

Kiểu dữ liệu	**Kích thước**	**Miền giá trị**
Tinyint	1 byte	-127 – 128 hay 0..255
Smallint	2 byte	-32768 – 32767 hay 0..65535
Mediumint	3 byte	-8388608 – 838860 hay 0..16777215
Int	4 byte	-2^{31} – 2^{31}-1 hay 0..2^{32}-1
Bigint	8 byte	-2^{63} – 2^{63}-1 hay 0..2^{64}-1

Kiểu dữ liệu true/false

Kiểu dữ liệu	**Kích thước**	**Miền giá trị**
Bool/ bloolean	1 byte	Có giá trị là true hoặc false

Kiểu số thập phân: decimal và numeric

Decimal và numeric là những kiểu dữ liệu được dùng để lưu trữ các giá trị số cụ thể. Giá trị của decimal và numeric được lưu trữ với một định dạng nhị phân.

Cú pháp:

Decimal(M[,N])

Trong đó:

+ M: tổng ký số.

+ N: số ký số thập phân, nếu N=0 được hiểu là không ký số thập phân và tương đương Decimal(M)

Các kiểu dữ liệu số thực

Kiểu dữ liệu	**Kích thước**	**Miền giá trị**
Float	4 bytes	3.402823466E-38 – 1.175493451E+38
Double	8 bytes	1.7976931348623157E-308 – 2.2250738585072014E+308

Kiểu dữ liệu ngày giờ

Kiểu dữ liệu	**Kích thước**	**Diễn giải**
Date		
Datetime		
Time		
Year[(2|4)]		
Timestamp[(kích cỡ định dạng)]		

Kiểu dữ liệu chuỗi

Kiểu dữ liệu	**Kích thước**	**Diễn giải**
Char	1÷ 255	Chuỗi cố định
Varchar	1÷ 255	Chuỗi động
TinyBlob	1÷ 255	Kiểu đối tượng nhị phân cỡ 255 ký tự
Tinytext	1÷ 255	Kiểu đối tượng chuỗi kích cỡ 255 ký tự
Blob	1÷ 65535	Kiểu blob
Text	1÷ 65535	Kiểu dạng văn bản cỡ 65535 ký tự
MediumBlob	1÷ 16777215(byte)	
Mediumtext	1÷ 16777215(ký tự)	

longBlob	$1 \div 2^{32}$-1(byte)	
longtext	$1 \div 2^{32}$-1(ký tự)	

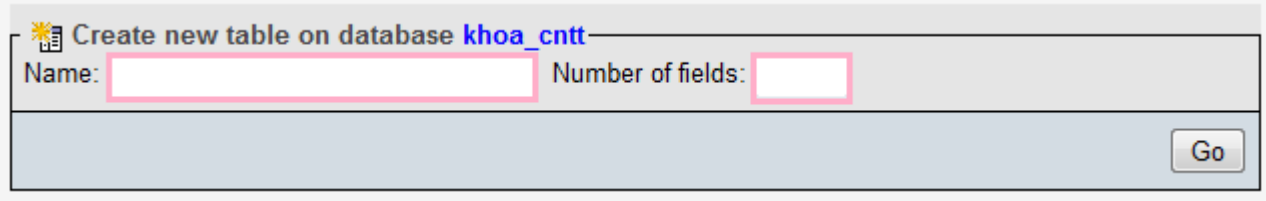
3. Thao tác với bảng

a. Tạo bảng

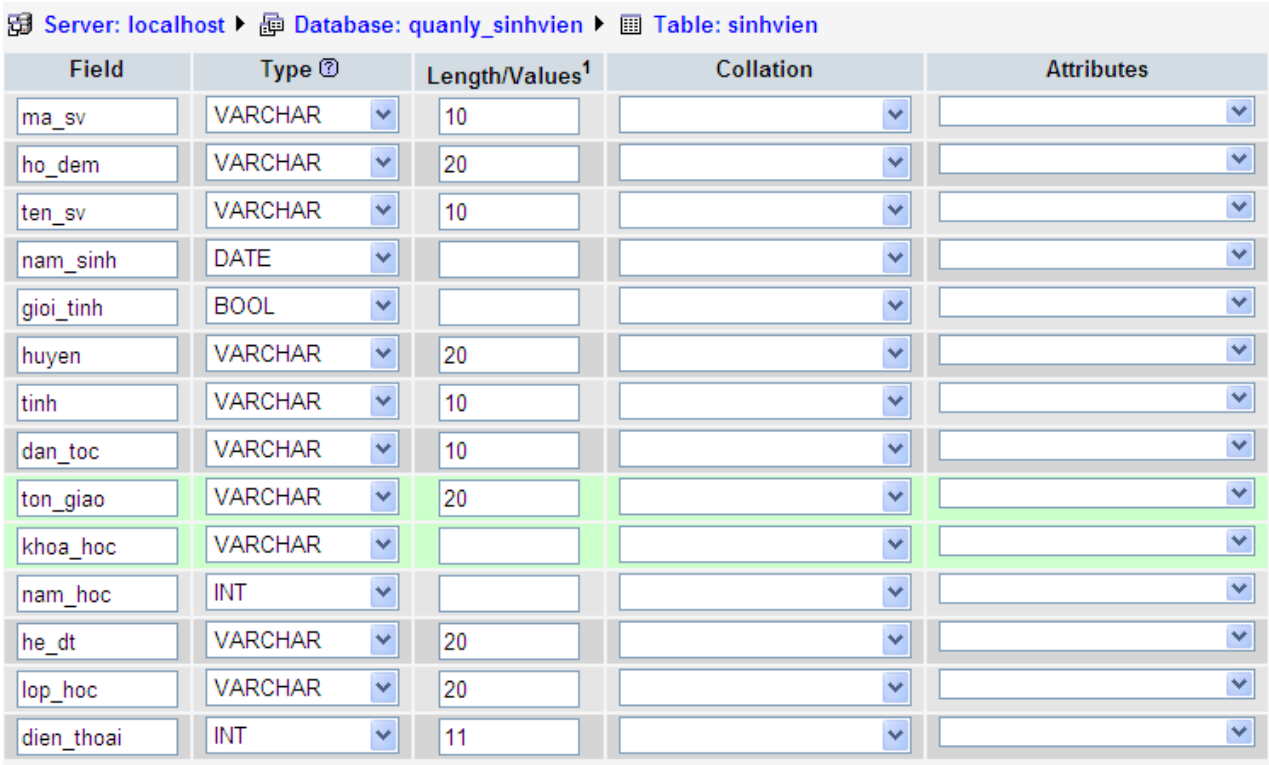
- Cách tạo bảng bằng giao diện đồ họa:

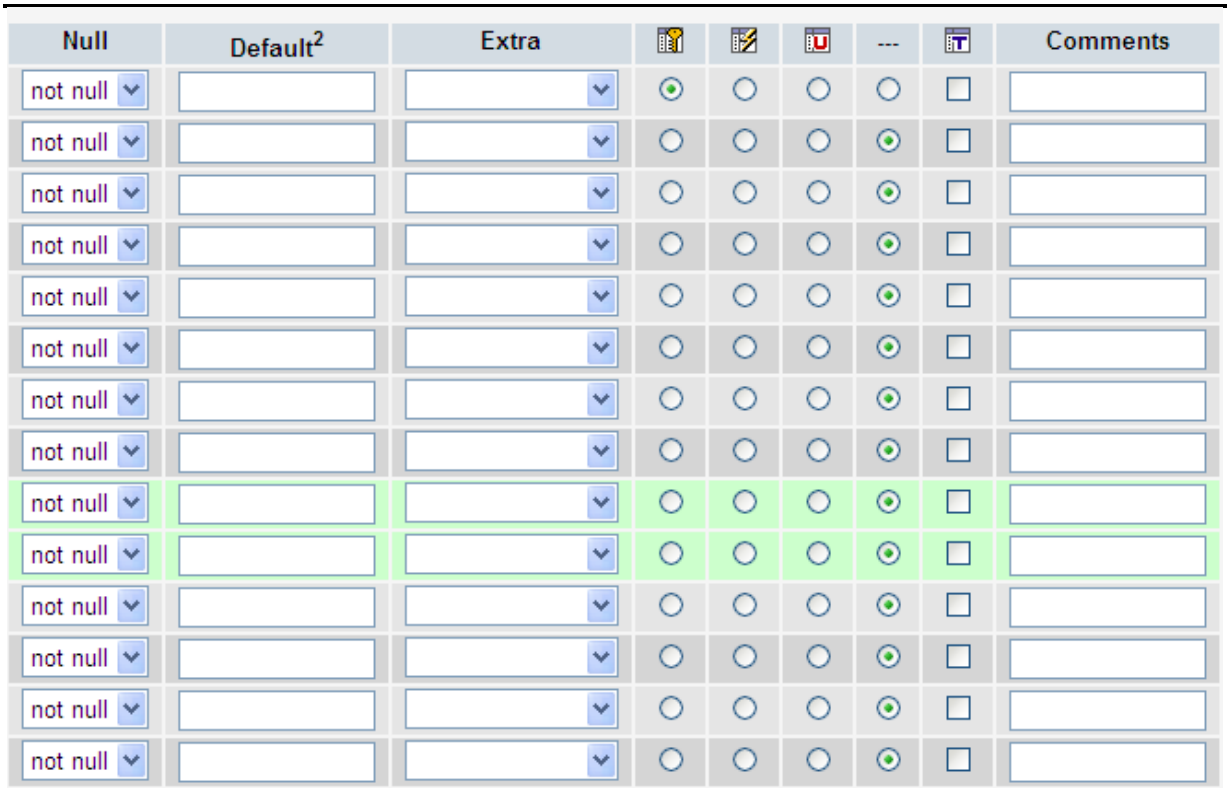
Bước 1: Chọn CSDL để tạo bảng

Bước 2: Nhập tên bảng vào trong Name và nhập số cột vào Number of fields sau đó nhấn Go



Bước 3: Nhập tên trường, chọn kiểu trường, độ dài của giá trị, ….





Bước 4: Nhấn vào Save để hoàn thành quá trình tạo bảng

- Tạo bảng bằng lệnh SQL

Cú pháp:

```
CREATE TABLE name_table
{
        Name_column_1 type_data[(size)] [<parameter>],
          Name_column_2 type_data[(size)] [<parameter>],
          …
}
```

Trong đó: Một số parameter sau:

+ NO NULL: không cho pháp dữ liệu trong cột để trống

+ DEFAULT giá_trị: cho phép cột có giá trị mặc định

+ PRIMARY KEY: thiết lập khóa chính của bảng

+ Auto_Increment: Xác định cột tăng giá trị tự động

Ví dụ: Tạo bảng sinhvien

```
CREATE TABLE `sinhvien` (
  `ma_sv` varchar(10) NOT NULL,
  `ho_dem` varchar(20) NOT NULL,
```

```
  `ten_sv` varchar(10) NOT NULL,
  `nam_sinh` date NOT NULL,
  `gioi_tinh` tinyint(1) NOT NULL,
  `huyen` varchar(20) NOT NULL,
  `tinh` varchar(10) NOT NULL,
  `dan_toc` varchar(10) NOT NULL,
  `ton_giao` varchar(20) NOT NULL,
  `khoa_hoc` varchar(10) NOT NULL,
  `nam_hoc` int(11) NOT NULL,
  `he_dt` varchar(20) NOT NULL,
  `lop_hoc` varchar(20) NOT NULL,
  `dien_thoai` int(11) NOT NULL,
  PRIMARY KEY  (`ma_sv`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

b. Thay đổi cấu trúc bảng

Trong trường hợp ta muốn thêm một hay nhiều cột vào bảng đã có, ta sẽ dùng câu lệnh ALTER TABLE.

Cú pháp:

```
ALTER TABLE <name_table>
ADD <name_column>  <type_data>[(size) ][…]
```

Chú ý: tên cột mới thêm vào phải khác với tên cột đã có trong bảng.

Ví dụ: Thêm vào bảng khoa một trường có tên là dien_thoai sau trường giao_vu

```
ALTER TABLE `khoa` ADD `dien_thoai` VARCHAR( 11 ) NOT NULL AFTER `giao_vu` ;
```

c. Sửa đổi kiểu dữ liệu của cột

Khi chúng ta muốn sửa đổi kiểu dữ liệu cho cột đã có chúng ta có thể dùng lệnh ALTER TABLE

Cú pháp:

```
ALTER TABLE <name_table>
CHANGE <name_column_old>  <name_column_new> type_data_new [(size)]
```

Ví dụ: Thay đổi kích thước của trường dien_thoai

```
ALTER TABLE `khoa` CHANGE `dien_thoai` `dien_thoai` VARCHAR( 12 )
CHARACTER SET latin1 COLLATE latin1_swedish_ci NOT NULL
```

e. Hủy cột trong bảng

Khi không cần sử dụng cột trong bảng chúng ta sử dụng Cú pháp ALTER TABLE để hủy bỏ cột. Tuy nhiên, khi cột bị xóa thì dữ liệu bên trong cột cũng sẽ bị xóa mà không thể phục hồi được. Do đó cần phải cẩn thận khi hủy bỏ cột.

Cú pháp:

```
ALTER TABLE <name_table>
DROP  COLUMN <name_column>,…
```

f. Xóa bảng

Khi chúng ta không cần bảng hoặc cấu trúc bảng không phù hợp chúng ta có thể xóa bảng.

Cú pháp:

```
DROP TABLE <name_table>
```

III. Bảng ảo

1. Khái niệm

View bắt đầu được sử dụng từ phiên bản MySQL server 5.0.

View là một cách khác để hiện thị CSDL. Một view là một bảng ảo được lưu trữ trong CSDL nhưng không thật sự chứa dữ liệu. Thay vào đó, view là một đối tượng mà bên trong nó chỉ có một câu lệnh SELECT dùng để chọn lọc một cột, các dòng trong các bảng CSDL để người dùng có thể xem và truy cập.

2. Tạo bảng ảo

Tạo bảng ảo bằng lệnh CREATE VIEW

Cú pháp:

```
CREATE VIEW name_view [(name_column_view)]
AS
      command SELECT
[WITH CHECK OPTION]
```

Trong đó:

- Tên các cột trong view: là các tên được đặt tương ứng với các cột hay biểu thức tính toán trong câu lệnh SELECT.

- Câu lệnh SELECT: câu lệnh truy vấn, chọn lựa dữ liệu từ trong một hay trong nhiều bảng có liên kết với nhau.

- WITH CHECK OPTION: dùng để ngăn cản các thao tác cập nhật dữ liệu tác động vào bảng ảo có làm ảnh hưởng đến dữ liệu trong các bảng ảo được tạo ra bằng cách lấy nguồn dữ liệu từ bảng ảo này.

Ví dụ: Tạo một bảng ảo dùng để hiển thị thông tin tất cả các cột của bảng sinhvien, với mã sinh viên là "09Tin0012"

```
CREATE VIEW sinhvien_view
AS
SELECT * FROM sinhvien WHERE ma_sv= "09Tin0012"
```

Sau khi chúng ta muốn xem kết quả của view đã tạo, chúng ta sử dụng lệnh SELECT.

Cú pháp:

```
SELECT * FROM name_view
```

Ví dụ: Xem kết quả của view có tên sinhvien_view

```
SELECT * FROM sinhvien_view
```

Khi sử dụng bảng ảo để thực hiện tính toán thống kê dữ liệu, chúng ta có thể đặt tên cho các cột theo hai cách:

Cách 1: Đặt tên cột ngay sau câu lệnh CREATE VIEW

Ví dụ: Tạo một bảng ảo có tên là sinhvien_view_sum dùng để đếm số sinh viên. Dữ liệu hiển thị gồm các cột: ma_sv, tong_sv

```
CREATE VIEW sinhvien_view_sum (ma_sv, sum_sv)
AS
SELECT ma_sv, count(ma_sv) FROM sinhvien GROUP BY  ma_sv
```

Cách 2: Đặt tên cho cột tính toán ngay trong câu lệnh SELECT

Ví dụ: Tạo một bảng ảo có tên là sinhvien_view_sum dùng để đếm số sinh viên. Dữ liệu hiển thị gồm các cột: ma_sv, tong_sv

```
CREATE VIEW sinhvien_view_sum (ma_sv, sum_sv)
AS
SELECT ma_sv, count(ma_sv) as sum_sv FROM sinhvien GROUP BY  ma_sv
```

3. Cập nhật nội dung bảng ảo

Để sử đổi nội dung có trong bảng ảo ta sử dụng Cú pháp ALTER VIEW, Cú pháp này cũng tương tự như CREATE VIEW

Cú pháp:

```
ALTER VIEW name_view [(name_column_view)]
AS
    Câu lệnh SELECT mới
[WITH CHECK OPTION]
```

Ví dụ: Từ bảng ảo có tên là sinhvien_view, cho biết sinh viên có điểm tổng kết cao nhất của năm học

```
ALTER VIEW sinhvien_view
AS
SELECT ma_sv, count(ma_sv) as sum_sv, max(diemtk) as diem_max FROM sinhvien
GROUP BY as ma_sv
```

4. Xóa bảng ảo

Khi chúng ta không cần sử dụng bảng ảo nữa, chúng ta có thể xóa bỏ bảng ảo. Khi xóa bảng ảo, dữ liệu trong bảng nguồn không ảnh hưởng.

Cú pháp:

```
DROP VIEW name_view
```

Ví dụ:

IV. Toán tử

1. Khái niệm

MySQL cung cấp cho chúng ta các toán tử như: toán tử số học, toán tử so sánh, toán tử logic.

Các toán tử này được kết hợp vào bên trong các mệnh đề WHERE, HAVING, IF, CASE,…

2. Toán tử số học

Dùng để tính toán các phép tính: cộng, trừ, nhân, chia, chia lấy phần dư. Giá trị được đem tính toán phải là kiểu số.

Khi có nhiều phép tính thì chúng ta nên đưa từng biểu thức tính toán vào trong dấu ngoặc đơn () để việc tính toán đó được tường minh.

Toán tử toán học:

+	Cộng
-	Trừ
*	Nhân
/	Chia
%	Chia lấy phần dư

Chú ý: Toán tử toán học cho phép sử dụng các kiểu dữ liệu số, tuy nhiên đối với phép tính chia lấy phần dư thì chúng ta chỉ có thể sử dụng kiểu số nguyên.

3. Toán tử so sánh

Dùng để thực hiện các phép tính so sánh như: bằng, lớn hơn, nhỏ hơn, khác,… cho các biểu thức cần so sánh. Kết quả trả về của phép so sánh là đúng hoặc sai.

Toán tử so sánh được sử dụng cho nhiều kiểu dữ liệu khác nhau như kiểu số, kiểu chuỗi…

Toán tử so sánh:

=	So sánh bằng
< = >	So sánh bằng cả khi hai giá trị đem so sánh đều là NULL
<>, !=	So sánh khác
<	So sánh nhỏ hơn
<=	So sánh nhỏ hơn hoặc bằng
>	So sánh lớn hơn
>=	So sánh lớn hơn hoặc bằng

4. Toán tử logic

Để kết hợp các biểu thức so sánh đơn lẻ thành một biểu thức chung.

Toán tử logic:

AND, &&	Và
OR, ||	Hoặc
XOR	Nếu hai biểu thức cùng đúng thì trả về giá trị false ngược lại true
NOT, !	Phủ định

V. Phát biểu SQL

1. Câu lệnh SELECT

a. Truy vấn đơn giản SELECT … FROM

Câu lệnh này giúp chúng ta chọn ra dữ liệu của các cột có trong một bảng.

Cú pháp:

```
SELECT list_column
FROM name_table
```

b. Truy vấn có sắp xếp dữ liệu

Câu lệnh SELECT … FROM kết hợp với mệnh đề ORDER BY giúp chúng ta lấy dữ liệu của các cột bên trong bảng đồng thì sắp xếp lại dữ liệu theo thứ tự tăng dần hay giảm dần.

Cú pháp:

```
SELECT list_column
FROM name_table
ORDER BY  name_column_sort [DESC,…]
```

c. Truy vấn có điều kiện WHERE

Câu lệnh SELECT … FROM kết hợp với mệnh đề điều WHERE giúp chúng ta lọc các dòng dữ liệu bên trong bảng, dữ liệu này phải thỏa mãn điều kiện đưa ra trong mệnh đề WHERE.

Cú pháp:

```
SELECT list_column
FROM name_table
WHERE conditonal
[ORDER BY  name_column_sort [DESC,…]]
```

Các phép toán thường dùng trong điều kiện lọc

- Các phép so sánh
  - >, >= : so sánh lớn hơn, lớn hơn hoặc bằng
  - <, <= : so sánh nhỏ hơn, nhỏ hơn hoặc bằng
  - = : so sánh bằng
  - !=, <> : so sánh khác

- Các phép toán học
  - and : phép và
  - or : phép hoặc
  - not : phép phủ định
  - not in : phép phủ định tập hợp
  - between : kết quả phụ thuộc vào miền giá trị
  - like : phép toán so sánh gần giống, sử dung % để thay thế ký tự
  - not like : phép phủ định so sánh gần giống
  - in : phép so sánh trong một tập hợp

d. Nhóm dữ liệu GROUP BY

Lệnh SELECT … FROM kết hợp với mệnh đề GROUP BY giúp chúng ta nhóm dữ liệu của các dòng dữ liệu bên trong bảng và sử dụng thêm các hàm thống kê đi kèm để tính toán dữ liệu có tính chất thống kê.

Cú pháp:

```
SELECT list_column
```

```
FROM name_table
[WHERE conditional]
GROUP BY list_column_group
[ORDER BY  name_column_sort [DESC,…]]
```

e. Điều kiện lọc nhóm HAVING

Cú pháp:

```
SELECT list_column
FROM name_table
[WHERE conditional]
GROUP BY list_column_group
HAVING conditional
[ORDER BY  name_column_sort [DESC,…]]
```

f. Giới hạn mẫu tin LIMIT

Cú pháp:

```
SELECT list_column
FROM name_table
[WHERE conditonal]
[GROUP BY list_columns_group]
[HAVING conditonal]
[ORDER BY  name_list_sort [DESC,…]]
LIMIT n,m
```

2. Truy vấn con

Truy vấn con là một câu lệnh select được lồng vào trong các câu lệnh truy vấn khác nhằm thực hiện các truy vấn tính toán phức tạp.

Chú ý: Khi dùng truy vấn con cần tuân theo các quy tắc sau:

+ Truy vấn con phải đặt trong dấu ngoặc đơn ()

+ Truy vấn con chỉ có thể tham chiếu đến một cột hoặc một biểu thức.

Kết quả trả về của truy vấn con có thể là một giá trị hoặc một danh sách các giá trị.

a. Truy vấn con trả về giá trị

Truy vấn con trả về một giá trị là truy vấn mà kết quả trả về của nó là một giá trị duy nhất.

Ví dụ:

b. Truy vấn con trả về danh sách các giá trị

Truy vấn con trả về danh sách các giá trị là truy vấn con mà kết qảu trả về là tập hợp các giá trị.

Toán tử IN hoặc NOT IN thường được dùng trong trường hợp này vì nó so sánh một phần tử có thuộc (hay không thuộc) tập hợp các giá trị hay không.

Ví dụ

c. Làm việc với các toán tử so sánh

Các toán tử so sánh thương được sử dụng trong truy vấn con có thể là: >, >=, <, <=, =, <>.

Chú ý: Thông thường các toán tử so sánh được sử dụng khi truy vấn con trả về một giá trị.

d. Làm việc với toán tử truy vấn con

Các toán tử truy vấn con thường hay sử dụng là: ANY, SOME, ALL, IN, NOT IN, EXISTS, NOT EXISTS.

Chú ý: Thông thường các toán tử truy vấn con được sử dụng khi dùng truy vấn cón trả về tập hợp các giá trị.

Quy tắc:

IN ⇔ ANY

NOT IN ⇔ ALL

3. Câu lệnh thêm dữ liệu

Câu lệnh INSERT INTO cho phép chúng ta thêm mới một hay nhiều dòng dữ liệu vào bên trong một bảng.

a. Giá trị trực tiếp

Khi chúng ta có giá trị trực tiếp cần thêm vào một bảng thì chúng ta sử dụng câu lệnh INSERT.

Cú pháp:

```
INSERT INTO name_table [(list_columns)]
VALUES (list_values)
```

b. Lấy từ nguồn dữ liệu

Trong trường hợp chúng ta muốn lấy dữ liệu từ các bảng khác để thêm vào bảng thì chúng ta kết hợp giữa INSERT và SELECT.

Cú pháp:

```
INSERT INTO name_table [(list_columns_table)]
SELECT list_columns_values
FROM table_source
WHERE conditional
```

4. Câu lệnh cập nhật dữ liệu

Đôi khi chúng ta có nhu cầu thay đổi giá trị của dữ liệu bên trong bảng khi chúng không còn phù hợp nữa. Câu lệnh UPDATE cho phép chúng ta cập nhật dữ liệu đã tồn tại bên trong bảng.

Chú ý: Chúng ta cần cân nhắc khi cập nhật dữ liệu bởi vì dữ liệu khi cập nhật thì không thể khôi phục lại giá trị ban đầu được nữa.

a. Giá trị trực tiếp

Khi chúng ta muốn cập nhật giá trị trực tiếp hay một biểu thức có giá trị trả về cho mẫu tin bên trong bảng, chúng ta cần dùng câu lệnh UPDATE

Cú pháp:

```
UPDATE name_table
SET name_column = value (or expression)
WHERE conditional_update
```

b. Lấy dữ liệu từ các bảng khác

Khi chúng ta muốn lấy dữ liệu từ các bảng khác để cập nhật vào bảng thì chúng ta kết hợp giữa UPDATE và SELECT

```
UPDATE name_table
SET name_column = (SELECT … FROM … WHERE …)
WHERE conditional_update
```

5. Câu lệnh xóa dữ liệu

Khi dữ liệu trong bảng không còn cần sử dụng nữa chúng ta có thể huy bỏ các dòng dữ liệu này. Câu lệnh DELETE cho phép chúng ta xóa dữ liệu trong bảng.

a. Câu lệnh xóa dữ liệu đơn giản

b. Câu lệnh xóa dữ liệu có điều kiện được lấy từ bảng khác.

6. Sử dụng mệnh đề UNION trong truy vấn

Mệnh đề UNION dùng để kết nối dữ liệu của các câu lệnh truy vấn lại với nhau:

Cú pháp:

```
SELECT danh sách các cột 1
FROM tên bảng 1
[WHERE …]
[GROUP BY … [HAVING…]]
UNION
SELECT danh sách các cột 2
FROM tên bảng 2
[WHERE …]
[GROUP BY … [HAVING…]]
[ORDER BY…]
```

Chú ý:

+ Với truy vấn sử dụng UNION thì danh sách các cột trong các câu truy vấn phải tương ứng với nhau về số lượng, thứ tự và kiểu dữ liệu của các cột.

+ Khi dùng UNION, việc đặt tiêu đề cột được thực hiện ngay truy vấn đầu tiên.

+ Với UNION có thể kết hợp nhiều truy vấn với nhau.

Ví dụ:

7. Truy vấn dữ liệu từ nhiều bảng

Khi muốn liên kết các bảng có quan hệ với nhau để lấy ra dữ liệu chung chúng ta kết hợp lệnh SELECT … FROM với mệnh đề JOIN.

Khi sử dung JOIN để nối các bảng chúng ta cần phải lưu ý những bảng này phải có các cột liên hệ với nhau và thứ tự quan hệ chúng ta chỉ định giữa các bảng cũng sẽ làm ảnh hưởng tới kết quả truy vấn.

a. INNER JOIN

Khi kết nối các bảng dùng INNER JOIN, ta chỉ định việc so sánh giá trị trong các cột của các bảng là tương đương – dữ liệu đều có ở cả hai bảng.

Kết quả sau khi thực hiện truy vấn kết nối INNER JOIN là các mẫu tin thỏa điều kiện quan hệ ở cả hai bảng.

Cú pháp:

```
SELECT list_column
FROM name_table
INNER JOIN name_table_link ON conditional_link
[WHERE conditional]
[ORDER BY list_column_sort [DESC]]
```

Ví dụ:

b. LEFT JOIN, RIGHT JOIN

Khi kết nối các bảng dùng LEFT|RIGHT JOIN, ta chỉ định việc so sánh giá trị trong các cột của các bảng được ưu tiên cho mối quan hệ bên nhánh trái | phải. Việc đổi thứ tự ưu tiên này sẽ làm ảnh hưởng tới kết quả truy vấn.

Cú pháp:

```
SELECT list_column
FROM name_table
LEFT|RIGHT name_table_link
ON conditional_link
[WHERE conditional]
[ORDER BY list_column_sort [DESC]]
```

c. Mệnh đề liên kết dữ liệu nhiều bảng

Cũng với SELECT … FROM với JOIN, chúng ta có thể kết hợp nhiều bảng dữ liệu trong một câu lệnh truy vấn. Một bảng có thể liên kết với một hay nhiều bảng khác nhau trong cùng một câu truy vấn.

Cú pháp:

```
SELECT list_column
FROM name_table_1
INNER LEFT|RIGHT name_table_2
ON conditional_link_2
INNER LEFT|RIGHT name_table_3
ON conditional_link_2
…
[WHERE conditional]
[ORDER BY list_column_sort [DESC]]
```

8. Sử dụng hàm trong SQL

a. Các hàm cấu trúc điều khiển

- Hàm IF

Cú pháp:

IF(biểu_thức_so_sánh, biểu_thức_1, biểu_thức_2)

Kiểm tra biểu thức so sánh đúng thì kết quả trả về là một biểu thức 1, ngược lại kết quả trả về là biểu thức 2

Ví dụ:

- Hàm IFNULL

Cú pháp:

- Hàm NULLIF

Cú pháp:

- Hàm CASE

Cú pháp:

b. Các hàm chuyển đổi kiểu dữ liệu

- Hàm CAST
- Hàm CONVERT

c. Hàm xử lý chuỗi

- Hàm CHAR_LENGHT, hàm CHARACTER_LENGHT VÀ LENGTH
- Hàm CONCAT và hàm CONCAT_WS
- Hàm LOWER và hàm UPPER
- Hàm LEFT, hàm RIGHT, hàm MID và hàm SUBSTRING
- Hàm REPEAT
- Hàm REVERSE
- Hàm REPLACE
- Hàm ENCODE và hàm DECODE
- Hàm SPACE
- Hàm STRCMP

d. Các hàm sử lý số

- Hàm ABS()

- Hàm CEILING()/ hàm CEIL()
- Hàm FLOOR()
- Hàm MOD()
- Hàm PI()
- Hàm POW() và hàm POWER()
- Hàm ROUND()
- Hàm SQRT
- Hàm SIGN()
- Hàm RAND()

e. Các hàm xử lý thời gian

- Hàm ADDDATE/ DATE_ADD/ SUBDATE()/ DATE_SUB()
- Hàm CURDATE / CURRENT_DATE / CURTIME/ CURRENT_TIME/ NOW()
- Hàm DATE()/ MONTH() MONTHNAME() / YEAR()
- Hàm DAY()/DAYOFMONTH/ DAYNAME/ DAYOFWEEK/ DAYOFYEAR
- Hàm SECOND/ MINUTE/ HOUR/ TIME
- Hàm DATEDIFF/ TIMEDIFF

9. Import và export dữ liệu

a. Import dữ liệu

Nhập dữ liệu từ bên ngoài vào database trong MySQL

b. Export dữ liệu

Xuất dữ liệu từ database trong MySQL ra tập tin thuộc một trong các dạng sau:

- SQL
- LaTex
- Microsoft Excel 2000
- Microsoft Word 2000
- CSV for MS Excel
- CSV
- XML

Chương 9: PHP&MYSQL

I. Kết nối CSDL

1. Tạo kết nối

Trước khi chúng ta truy cập và làm việc với dữ liệu trong CSDL, chúng ta cần phải tạo kết nối đến CSDL.

Để thực hiện được công việc này, chúng ta sử dụng hàm **mysql_connect()**

Cú pháp:

```
mysql_connect(servername, username, password);
```

Trong đó:

+ servername: tham số tùy chọn, xác định server cần phải kết nối tới. Giá trị mặc định là localhost.

+ username: tham số tùy chọn, xác định tên người dùng đăng nhập vào hệ thống

+ password: tham số tùy chọn, xác định mật khẩu người dùng. Giá trị mặc định là "".

Chú ý: Hàm này có nhiều hơn 3 tham số, tuy nhiên những tham số trên cần thiết và rất quan trọng.

Ví dụ: Tạo kết nối

```
<?php
// tạo kết nối và lưu vào biến $conn
$conn =  mysql_connect("localhost","root","");
// kiểm tra kết nối
if(!$conn)
{
      die ("Can not connect database ".mysql_error());
      exit;
}
?>
```

2. Chọn CSDL

Sau khi đã tạo được kết nối, chúng ta cần phải chọn một CSDL để làm việc. Để thực hiện công việc này, chúng ta sử dụng hàm **mysql_select_db**().

Kết quả trả về của hàm này là TRUE nếu chọn CSDL thành công, ngược lại giá trị trả về là FALSE.

Cú pháp:

mysql_select_db(database, connection)

Trong đó:

+ database: tham số bắt buộc, xác định tên CSDL cần làm việc

+ connection: tham số tuy chọn, xác định kết nối. Nếu không xác định thì kết nối cuối cùng được mở bởi hàm **mysql_connect()** hoặc hàm **mysql_pconnect**() sẽ được sử dụng.

Ví dụ: Kết nối đến CSDL khoacntt

```
<?php
$conn =  mysql_connect("localhost","root","");
if(!$conn)
{
	echo "Can not connect database";
	exit;
}
// chọn CSDL khoacntt
$select_db = mysql_select_db("khoacntt",$conn);
// kiểm tra CSDL
if (!$select_db)
{
	die("Can not connect database".mysql_error());
}
?>
```

3. Truy vấn dữ liệu

Để thực hiện việc truy vấn dữ liệu, chúng ta dùng hàm **mysql_query()**.

Hàm **mysql_query()** sẽ trả về kết quả của cấu lệnh truy vấn nếu thực hiện thành công, ngược lại sẽ trả về FALSE.

Cú pháp:

mysql_query(query, connection) ;

Trong đó:

+ query: tham số bắt buộc, là câu lệnh truy vấn được gửi đi.

+ connection: tham số tùy chọn, xác định kết nối. Nếu không xác định thì kết nối cuối cùng được mở bởi hàm **mysql_connect()** hoặc **mysql_pconnect**() sẽ được sử dụng.

Ví dụ: Thực hiện truy vấn dữ liệu trong bảng sinhvien trong CSDL khoacntt.

```
<?php
```

```
$conn = mysql_connect("localhost","root","");
if (!$conn)
{
	echo "Can not connection database";
	exit;
}
mysql_select_db('khoacntt',$conn);
$sql ="SELECT * FROM sinhvien";
$result= mysql_query($sql);
while($r = mysql_fetch_array($result))
{
	echo "MaSV: ".$r['ma_sv']."  Ten sv: ".$r['ten_sv']."</br>";
}
?>
```

4. Thông báo lỗi

Trong quá trình làm việc với CSDL lỗi có thể phát sinh. Do đó, chúng ta cần thông báo các lỗi phát sinh này.

Để thực hiện công việc này, chúng ta sử dụng hàm **mysql_error()**.

Hàm này có kết quả trả về là câu thông báo lỗi nếu có lỗi phát sinh, ngược lại kết quả trả về sẽ là một chuỗi rỗng “”.

Cú pháp:

```
mysql_error(connection)
```

Trong đó:

connection: tham số tùy chọn, xác định kết quả kết nối. Nếu không kết nối thì kết quả cuối cùng được mở bởi hàm **mysql_connect()** hoặc **mysql_pconnect**() sẽ được sử dụng.

Ghi chú: Chúng ta thường kết hợp hàm **mysql_error()** với hàm **die**() hoặc hàm **exit**() để vừa thông báo lỗi vừa kết thúc công việc.

Ví dụ: Thông báo lỗi nếu không tạo kết nối

```
<?php
$conn = mysql_connect("localhost","root","");
if (!$conn)
{
	echo "Can not connection database";
	exit;
```

```
}
?>
```

5. Đóng kết nối

Sau khi đã làm việc xong với CSDL, chúng ta cần đóng kết nối đã mở.

Để thực hiện công việc này, chúng ta sử dụng hàm **mysql_close()**.

Hàm này có kết quả trả về là TRUE nếu đóng kết nối thành công, ngược lại sẽ trả về giá trị là FALSE nếu thất bại.

Cú pháp:

```
mysql_close(connection);
```

+ connection: tham số tùy chọn, xác định kết nối. Nếu không xác định thì kết nối cuối cùng được mở bởi hàm **mysql_connect() mysql_pconnect**() sẽ được sử dụng.

Ví dụ: Đóng kết nối

```
<?php
$conn = mysql_connect("localhost","root","");
……..
mysql_close($conn);
?>
```

II. Làm việc với CSDL MySQL

1. Đếm số lượng mẫu tin

Thông thường, chúng ta cần biết số lượng mẫu tin được truy vấn dữ liệu trước khi xử lý các công việc tiếp theo.

Để thực hiện công việc này, chúng ta sử dụng hàm **mysql_num_rows()**.

Hàm này có kết quả trả về là số lượng mẫu tin nếu thành công, ngược lại kết quả trả về là FALSE nếu thất bại.

Cú pháp:

```
mysql_num_rows(data);
```

Trong đó:

+ data: là tham số bắt buộc. Xác định con trỏ dữ liệu. Con trỏ dữ liệu là kết quả trả về của hàm **mysql_query()**

Ví dụ: Đếm số mẫu tin trong bảng sinhvien

```
<?php
```

```
$conn = mysql_connect("localhost","root","");
if (!$conn)
{
      echo "Can not connection database";
      exit;
}
mysql_select_db('khoacntt',$conn);
$sql ="SELECT * FROM sinhvien";
$result= mysql_query($sql);
echo "Số mẫu tin: ". mysql_num_rows($result);
mysql_close($conn);
?>
```

2. Hiển thị dữ liệu

a. Duyệt dữ liệu

Có nhiều cách để duyệt dữ liệu: duyệt dữ liệu theo dạng mỗi mẫu tin là một dòng, duyệt theo dạng mỗi mẫu tin là một mảng, duyệt theo dạng mỗi mẫu tin là một đối tượng.

Cách 1: Duyệt dữ liệu theo dạng mỗi mẫu tin là một dòng bằng hàm mysql_fetch_row().

Hàm này trả về một mảng (có chỉ số) chứa giá trị của một dòng dữ liệu với mỗi phần tử là nội dung của một cột.

Sau đó truy xuất bằng cách gọi từng phần tử của mảng $row[0], $row[1] ,$row[2],…

Cú pháp:

```
mysql_fetch_row(data)
```

+ data: là tham số bắt buộc. Xác định số con trỏ dữ liệu. Con trỏ dữ liệu là kết quả trả về của hàm **mysql_query()**.

Ví dụ: Duyệt dữ liệu trong bảng sinhvien sử dụng hàm **mysql_fetch_row()**

```
<?php
$conn =  mysql_connect("localhost","root","");
if (!$conn)
  {
  die('Could not connect: ' . mysql_error());
  }
mysql_select_db("khoacntt", $conn);
$result = mysql_query("SELECT * FROM sinhvien");
for($i=1;$i<=mysql_num_rows($result);$i++)
```

```
{
print_r(mysql_fetch_row($result));
echo "</br>";
}
mysql_close($conn);
?>
```

Cách 2: Duyệt theo dạng mỗi mẫu tin là một mảng bằng hàm mysql_fetch_array()

Hàm này trả về một mảng (có chỉ số chuỗi) chứa giá trị của một dòng dữ liệu với mỗi phần tử là nội dung của một cột.

Sau đó truy xuất bằng cách gọi từng phần tử của mảng $row[“tên cột 1”], $row[“tên cột 2”], $row[“tên cột 3”],…

Cú pháp:

```
mysql_fetch_array(data)
```

+ data: là tham số bắt buộc. Xác định số con trỏ dữ liệu. Con trỏ dữ liệu là kết quả trả về của hàm **mysql_query()**.

Ví dụ: Duyệt dữ liệu trong bảng sinhvien sử dụng hàm **mysql_fetch_array()**

```
<?php
$conn =  mysql_connect("localhost","root","");
if (!$conn)
  {
  die('Could not connect: ' . mysql_error());
  }
mysql_select_db("khoacntt", $conn);
$result = mysql_query("SELECT * FROM sinhvien");
while($row = mysql_fetch_array($result))
  {
  echo $row['ma_sv'] . " " . $row['ten_sv'];
  echo "<br />";
  }
mysql_close($conn);
?>
```

Cách 3: Duyệt theo dạng mỗi mẫu tin là một đối tượng bằng hàm mysql_fetch_object()

Hàm này có kết quả trả về là một mẫu tin trong bộ các mẫu tin như một đối tượng.

Sau đó truy xuất bằng cách gọi từng thuộc tính của đối tượng $tên_đôi_tượng ->tên cột 1, $tên_đôi_tượng ->tên cột 2, $tên_đôi_tượng ->tên cột 3,…

Cú pháp:

```
mysql_fetch_object(data)
```

+ data: là tham số bắt buộc. Xác định số con trỏ dữ liệu. Con trỏ dữ liệu là kết quả trả về của hàm **mysql_query()**.

Ví dụ: Duyệt dữ liệu trong bảng sinhvien sử dụng hàm **mysql_fetch_object()**

```
<?php
$conn =  mysql_connect("localhost","root","");
if (!$conn)
  {
  die('Could not connect: ' . mysql_error());
  }
mysql_select_db("khoacntt", $conn);
$result = mysql_query("SELECT * FROM sinhvien");
for($i=1;$i<=mysql_num_rows($result);$i++)
{
        $ob =mysql_fetch_object($result);
        echo $ob->ma_sv.", ".$ob->ten_sv."</br>";
}
mysql_close($conn);
?>
```

b. Hiển thị dữ liệu không định dạng

Khi chúng ta chỉ có nhu cầu hiện thị dữ liệu mà không cần định dạng thì cách đơn giản nhất là in dữ liệu theo dạng bảng.

Ví dụ: Hiển thị thông tin sinh viên trong bảng sinhvien

```
<p > <h1 align="center">THÔNG TIN SINH VIÊN</h1>
<table width="1000" border="1" cellspacing="2" cellpadding="2" align="center">
  <tr>
    <td width='100px'>Mã SV</td>
        <td width='250px'>Tên SV</td>
    <td width='80px'>Ngay sinh</td>
        <td width='70px'>Gioi tinh</td>
        <td width='300px'>Quen quan</td>
    <td width='100px'>Nam hoc</td>
        <td width='100px'>Lop hoc</td>
  </tr>
  <?php
  $conn =  mysql_connect("localhost","root","");
        if (!$conn)
```

```
        {
                die('Could not connect: ' . mysql_error());
        }
  mysql_select_db("khoacntt", $conn);
  $result = mysql_query("SELECT * FROM sinhvien");
  while($row = mysql_fetch_array($result))
  {
  echo "<tr>";
  echo "<td width='100px'>".$row[0]."</td>";
  echo "<td width='260px'>".$row[1]."</td>";
  echo "<td width='80px'>".$row[2]."</td>";
  echo "<td width='70px'>".$row[3]."</td>";
  echo "<td width='300px'>".$row[4]."</td>";
  echo "<td width='100px'>".$row[5]."</td>";
  echo "<td width='100px'>".$row[6]."</td>";
  echo "</tr>";
  }
  mysql_close($conn);
  ?>
</table>
</p>
```

c. Hiển thị dữ liệu có yêu cầu

Khi xây dựng một ứng dụng web nếu chúng ta hiển thị thông tin không có định dạng thì sẽ làm cho người xem cảm thấy nhàm chán và khó khăn trong quá trình đọc và tìm kiếm dữ liệu, chính vì vậy mà phát sinh các yêu cầu hiển thị dữ liệu khác nhau.

- Định dạng hiển thị dữ liệu

Yêu câu đơn giản nhất của một trang web là dữ liệu hiển thị phải được định dạng giúp cho người dùng dễ dàng đọc thông tin.

Các định dạng thông thường là định dạng các cột, dòng và kích thước, màu sắc, kiểu chữ, … cho nội dung hiển thị.

Chú ý: Để thực hiện được các yêu cầu về định dạng, chúng ta cần phải xem xét và tính toán các nội dung cần hiển thị.

Ví dụ:

- Tạo các cột tùy biến

Đôi khi, nội dung của một số cột trong CSDL khi hiển thị sẽ gây cho người dùng khó hiểu, hoặc chúng ta phải có dòng giải thích cho những thông tin đó. Chính vì vậy mà nhu cầu hiển

thị dữ liệu một cách rõ ràng, chi tiết, dễ hiểu trở thành một yêu cầu tất yếu khi thiết kế trang web.

Ví dụ:

- Phân trang

Đối với những bảng có số lượng mẫu tin lớn khi hiển thị dữ liệu sẽ làm cho người dùng cảm thấy khó xem. Phân trang dữ liệu giúp cho việc xem và tìm kiếm thông tin dễ dàng, nhanh chóng và thuận tiện.

Ví dụ:

- Liên kết trang có chuỗi tham số.

* Chuỗi URL có cấu trúc định dạng như sau:

http://<host>[:<port>][path][?string_parameter]

Trong đó:

string_parameter có Cú pháp như sau:

[?parameter_1 = value_1 [&parameter_2 = value_2[&…]]]

Chú ý: Trong trường hợp có nhiều tham số, cặp [parameter_1 = value_1] phân cánh nhau bằng dấu &

Ví dụ:

Để nhận giá trị truyền qua tham số này ta dùng biến $_REQUEST[“parameter”]

Ví dụ:

d. Chuyển đổi giá trị thời gian

Định dạng date time trong MySQL có dạng năm-tháng-ngày.

Ví dụ: Khi nhập ngày – tháng – năm ở PHP như sau: 25th January 2009 thì khi lưu trữ vào CSDL MySQL chúng sẽ được định nghĩa như sau: 2009-01-25

Khi nhập ngày tháng năm từ PHP vào MySQL ta có thể dùng hàm date() đã học hoặc sử dụng hàm DATE_FROMAT() trong MySQL

Để định dạng ngày tháng năm YYYY-MM-DD của MySQL sang định dạng MM-DD-YYYY có thể dùng hàm DATE_FORMAT() với Cú pháp như sau:

```
SELECT DATE_FORMAT (column_date ‘%M-%D-%Y’)
Hay
SELECT DATE_FORMAT (column_date, %M %D %Y)
```

Trong đó: %M và %D là định dạng tháng và ngày với 2 ký số, %Y là định dạng năm với 4 ký số.

Chú ý: Chúng ta cũng có thể căn cứ vào bảng định dạng thời gian sau để định dạng theo yêu câu:

Ký tự	Mô tả
%M	Tên tháng trong năm(January, February…)
%W	Tên thứ trong tuần (Sunday, Monday,…)
%D	Ngày trong tháng theo dạng 1^{st}, 2^{nd}, 3^{th}…
%Y	Năm với 4 ký số(2010)
%y	Năm với 2 ký số(10)
%m	Tháng với hai ký số (01-12)
%d	Ngày trong tháng với hai ký số (01-31)
%H	Giờ trong ngày với hai ký số(00-23)
%h	Giờ trong ngày với hai ký số(00-12)
%i	Phút trong giờ với hai ký số(01-59)
%r	12 giờ với định dạng (hh:mm:ss [AM|PM])
%T	12 giờ với định dạng (hh:mm:ss)
%S	Giây trong phút có hai ký số (00-59)
%p	AM hay PM

3. Lưu trữ dữ liệu mới vào CSDL

Để lưu thông tin từ trong PHP vào trong MySQL, chúng ta sẽ sử dụng hàm **mysql_query()** kết hợp với câu lệnh truy vấn INSERT INTO. Hàm này được dùng để gửi một truy vấn(hiển thị thông tin, thêm mới, xóa, cập nhật) tới một kết nối MySQL

Cú pháp:

```
INSERT INTO table_name
VALUES (value1, value2, value3,...)
Hoặc
INSERT INTO table_name (column1, column2, column3,...)
VALUES (value1, value2, value3,...)
```

Chú ý: Cú pháp INSERT INTO dùng để thêm một mẫu tin mới vào bảng trong CSDL

Ví dụ: Tạo form nhập thông tin của một mẫu tin(form_insert.php)

Các tên tương ứng

- masv
- tensv
- d, m, y
- gt
- quequan
- namhoc
- lophoc

Ví dụ: Trang lấy thông tin từ form (insert_sinhvien.php)

```
<?php
$masv = $_POST['masv'];
$tensv = $_POST['tensv'];
$d = $_POST['d'];
$m = $_POST['m'];
$y = $_POST['y'];
$gt = $_POST['gt'];
$quequan = $_POST['quequan'];
$namhoc = $_POST['namhoc'];
$lophoc = $_POST['lophoc'];
// xu ly dl ngay thang nam
if (checkdate($m,$d,$y))
{
	$ngaysinh =$d."/".$m."/".$y;
}
// kiem tra xem trong csdl co masv nay chua
function kt($masv)
{
	$conn = mysql_connect("localhost","root","");
	mysql_select_db("khoacntt",$conn);
	$sql="SELECT ma_sv FROM sinhvien WHERE ma_sv='$masv'";
	$result=mysql_query($sql);
	$kt1=true;
	while ($row = mysql_fetch_array($result))
	{
		$kt1=false;
```

```
        }
        return $kt1;
        mysql_close($conn);
}
if (kt($masv)&& checkdate($m,$d,$y))
{
        $conn = mysql_connect("localhost","root","");
        mysql_select_db("khoacntt",$conn);
        $sql = "INSERT INTO sinhvien
VALUES('$masv','$tensv','$ngaysinh','$gt','$quequan','$namhoc','$lophoc')";
        mysql_query($sql);
}
?>
```

4. Cập nhật dữ liệu

Để cập nhật thông tin từ PHP vào MySQL, chúng ta sẽ sử dụng hàm **mysql_query()** kết hợp với câu lệnh truy vấn UPDATE. Hàm này được dùng để gửi một truy vấn (hiển thị thông tin, thêm mới, xóa, cập nhật) tới một kết nối MySQL

Cú pháp:

```
UPDATE table_name
SET column1=value, column2=value2,...
WHERE some_column=some_value
```

Ví dụ: Form cập nhật dữ liệu bảng sinh viên

- Mã sinh viên được đọc từ CSDL với tên listbox là masv.
- Tên sinh viên mới có tên textfield là tensv

Ví dụ: Trang lấy thông tin từ form cập nhật dữ liệu

```
<?php
$conn = mysql_connect("localhost","root","");
mysql_select_db("khoacntt", $conn);
if(strlen($_POST['tensv']))
{
```

```
        $tensv = trim($_POST['tensv']);
        $masv = trim($_POST['masv']);
        mysql_query("UPDATE sinhvien SET ten_sv = '$tensv' WHERE ma_sv = '$masv'");
}
mysql_close($conn);
?>
```

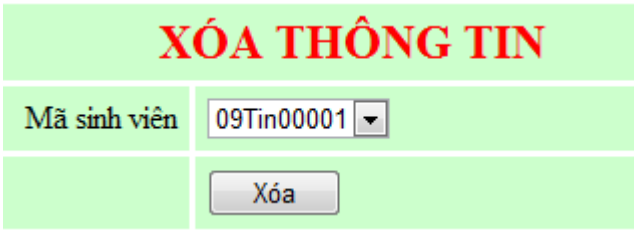
5. Xóa dữ liệu

Để xóa thông tin trong CSDL thông qua PHP, chúng ta sẽ sử dụng hàm **mysql_query()** kết hợp với câu lệnh truy vấn DELETE. Hàm này được dùng để gửi một truy vấn (hiển thị thông tin, thêm mới, xóa, cập nhật) tới một kết nối MySQL

Cú pháp:

```
DELETE FROM table_name
WHERE some_column = some_value
```

Ví dụ: Form xóa dữ liệu trong bảng sinhvien



- Chọn mã sinh viên cần xóa có tên listbox là masv, value được lấy trong CSDL

Ví dụ: Lấy thông tin từ form và xóa dữ liệu trong bảng sinh viên

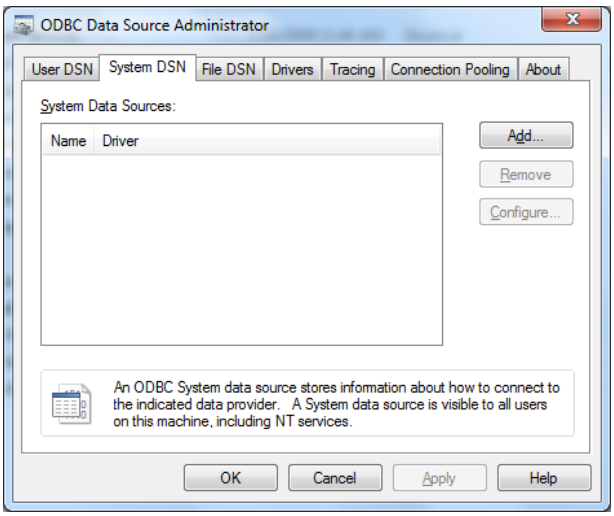
```
<?php
$conn = mysql_connect("localhost","root","");
mysql_select_db("khoacntt", $conn);
if(strlen($_POST['masv']))
{
        $masv = trim($_POST['masv']);
        mysql_query("DELETE FROM sinhvien WHERE ma_sv = '$masv'");
}
mysql_close($conn);
?>
```

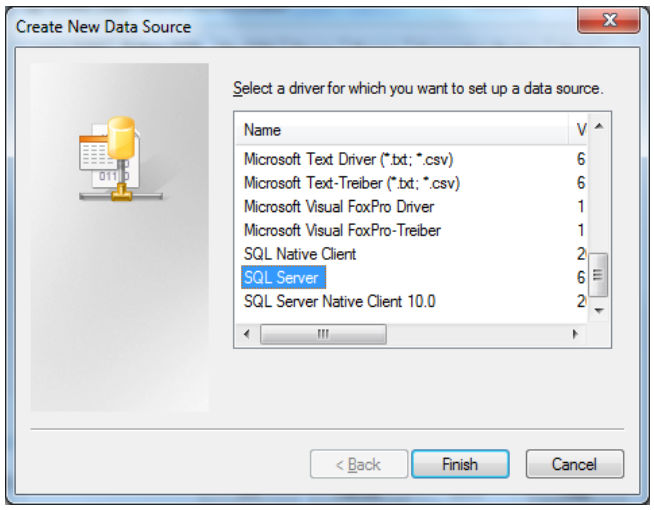
III. PHP kết hợp với các CSDL SQL Server

Bước 1: Mở Administrative Tools trong control Panel

Bước 2: Trong cửa sổ Administrative Tools chọn biểu tượng Data Sources(OBBC)

Bước 3: Trong cửa sổ ODBC Data Sources Administrator chọn tab System DSN

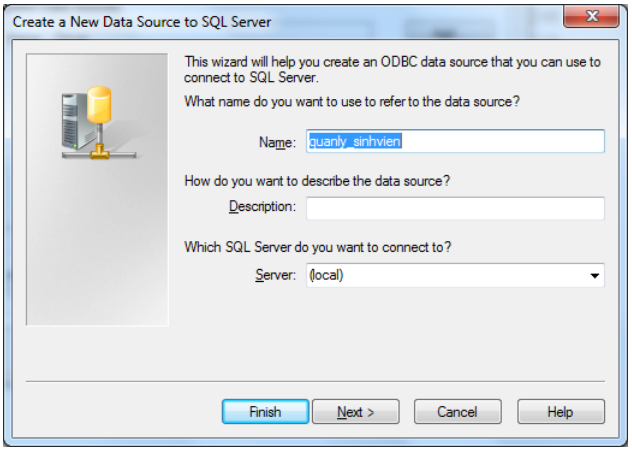


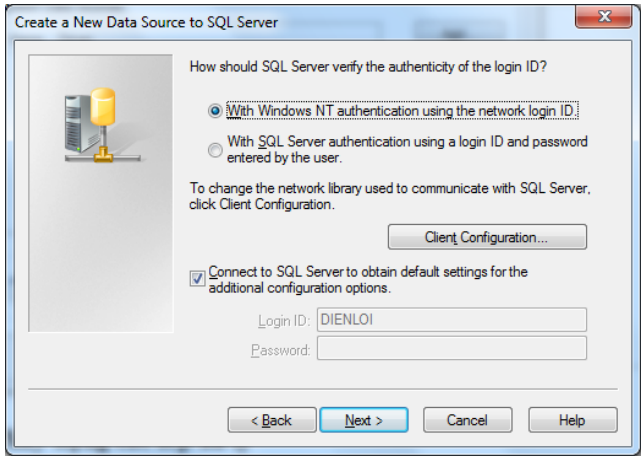


Bước 4: Nhấn vào nút Add chọn SQL Server chọn Finsh

Bước 5: Đặt tên cho Data Source (DSN) trong mục Name và chọn server cần kết nối tới trong mục server => nhấn nút next

Xác định cách thức kết nối tới SQL Server theo tài khoản của Window NT hay tài khoản của SQL Server => Nhấn nút next





Bước 6: Lựa chọn CSDL cần kết nối của SQL Server trong mục Change Default Database to => nhấn nút Next sau đó nhấn nút Test Data Source để kiểm tra việc kết nối với nguồn dữ liệu có thành công hay không => nhấn nút OK

Bước 7: Hoàn thành việc kết nối CSDL => nhấn nút OK

IV. Xây dựng các lớp xử lý

1. Một số phương thức trong lớp xử lý bảng

Lớp xử lý bảng dùng để xử lý các công việc liên quan tới kết nối, chọn CSDL và làm việc với CSDL,…

```
class connect_database
{
```

```
    // khai báo các thuộc tính
    // xây dựng các phương thức lớp xử lý bảng
}
```

a. Khai báo thuộc tính

Trong lớp xử lý bảng có một số thuộc tính thường xuyên được sử dụng nên chúng ta cần phải khai báo:

```
var $_sql ="";
var $_connection ="";
var $_result ="";
```

Trong đó:
+ $_sql : chứa nội dung của câu lệnh truy vấn
+ $_connection: chứa kết quả của hàm kết nối mysql_connect()
+ $_result: chứa kết quả của hàm mysql_query()

b. Kết nối CSDL

Trong hàm khởi tạo của lớp connect _database khai báo các thông tin kết nối đến CSDL.

```
{
    // Tạo và kiểm tra kết nối
    $this ->_connection =@mysql_connect("localhost","root","");
    // chọn CSDL
    $db ="name_database";
    mysql_select_db($db, $this->connection)
}
```

c. Gán giá trị cho thuộc tính $_sql

Hàm này sẽ gán giá trị cho thuộc tính $sql của lớp xử lý bảng

```
function setquery($sql)
{
    $this->_sql =$sql;
}
```

d. Lấy toàn bộ các dòng dữ liệu trong bảng

Hàm này có kết quả trả về là biến con trỏ chứa kết quả là các dòng dữ liệu trong bảng

```
function query()
{
$this->_result =mysql_query($this->_sql,$this->_connection);
return $this->_result;
```

```
}
```

Sau đó chuyển kết quả của biến $_result (gọi hàm query()) này vào mảng các dòng dữ liệu bằng hàm loadAllRow()

```
function loadAllRow()
{
	if(!($result = $this->query()))
	{
		return null;
	}$array =array();
	while ($row =mysql_fetch_assoc($result))
	{
		$array[] =$row;
	}
	mysql_free_result($result);
	return $array;
}
```

e. Đóng kết nối

Hàm này được dùng để đóng kết nối đang được mở

```
function close_connect()
{
	mysql_close($this->_connection);
}
```

2. Xây dựng lớp xử lý nghiệp vụ

Lớp xây dựng nghiệp vụ có các thành phần riêng của lớp đó. Lớp này kế thừa từ lớp xử lý bảng và có các hàm đọc, thêm, cập nhật, xóa dữ liệu.

```
class process_class extends connect_database
{
	// các thuộc tính
	// các phương thức
}
```

a. Các phương thức thường sử dụng

- Đọc dữ liệu

```
function doc_ds()
{
	$this ->setQuery("SELECT * FROM name_table");
```

```
        $result = $this->LoadAllRow();
        $this->close_connect();
        $return $result;
}
```

- Thêm dữ liệu

```
function them_moi(danh sach tham so)
{
        $this ->setQuery("INSERT INTO name_table VALUES (gia tri) ");
        $result = $this->query();
        $this->close_connect();
        $return $result;
}
```

- Cập nhật dữ liệu

```
function cap_nhat(danh sach tham so)
{
        $this ->setQuery("UPDATE name_table SET name_column=value, ...");
        $result = $this->query();
        $this->close_connect();
        $return $result;
}
```

- Xóa dữ liệu

```
function xoa(danh sach tham so)
{
        $this ->setQuery("DELETE FROM name_table WHERE ...");
        $result = $this->query();
        $this->close_connect();
        $return $result;
}
```

b. Các phương thức riêng cho các lớp

Ngoài các phương thức thường dùng được nêu ở trên, mỗi lớp xử lý nghiệp vụ còn có những phương thức đặc trưng, riêng biệt khác. Tùy theo yêu cầu mà chúng ta sẽ xây dựng các phương thức này.

Mục lục

Chương 1: Quy trình thiết kế website 1
I. Các khái niệm cơ bản 6
1. HTML (Hypertext Markup Language) – Ngôn ngữ đánh dấu siêu văn bản 6
2. Ngôn ngữ lập trình Web 6
3. WebServer – trình chủ Web 6
4. Database server – Trình chủ CSDL 6
5. Web browser-Trình duyệt Web 6
6. URL (Uniform Resource Locator)- Tài nguyên trên Internet 6
7. HTTP (Hypertext Transfer Protocol)- Giao thức truyền siêu văn bản 7
8. Cơ chế Web 7
II. Quy trình thiết kế website 8
1. Xác định mục đích, yêu cầu của website 8
2. Xác định độc giả 8
3. Thiết kế giao diện Website 9
a. Xác định kiểu chữ, màu sắc 9
b. Xác định các kỹ thuật, công cụ thiết kế 9
c. Cắt đoạn, tóm lược thông tin 9
d. Xác định cấu trúc WebSite 9
4. Các thành phần cơ bản của Website 11
a. Trang chủ (HomePage) 11
b. Hệ thống Menu, Logo, định danh 11
c. Các trang thành viên 11
III. Một số nguyên tắc khi phát triển website. 11
Chương 2: Giới thiệu về ngôn ngữ HTML 15
I. Khái niệm cơ bản về html 15
1. HTML là gì? 15
2. Thẻ HTML 15
3. Cần gì để tạo một trang web 15
II. Các thẻ định cấu trúc tài liệu 15
1. Thẻ html 15
2. Thẻ head 16
3. Thẻ title 16
4. Thẻ body 16
III. Các thẻ định dạng khối 17
1. Thẻ định dạng khối văn bản <p> 17
2. Các thẻ định dạng đề mục h1/h2/h3/h4/h5/h6 17
3. Thẻ xuống dòng
 18
4. Thẻ pre và thẻ 18
IV. Các thẻ định dạng danh sách 18
V. Các thẻ định dạng ký tự 19
1. Các thẻ định dạng in ký tự 19
2. Căn lề văn bản trong trang Web 20
3. Các ký tự đặc biệt 20
4. Sử dụng màu sắc trong thiết kế các trang Web 20
5. Chọn kiểu chữ cho văn bản 22
6. Khái niệm văn bản siêu liên kết 22
7. Địa chỉ tương đối 23
8. Kết nối mailto 24

9. Vẽ một đường thẳng nằm ngang 24
VI. Các thẻ chèn âm thanh, hình ảnh 25
1. Giới thiệu 25
2. Đưa âm thanh vào một tài liệu HTML 26
3. Chèn một hình ảnh, một đoạn video vào tài liệu HTML 26
VII. Các thẻ định dạng bảng biểu 27
VIII. FORM 28
1.Form 29
2. Hộp nhập văn bản 1 dòng (Oneline Textbox) 29
3. Radio Button 30
4. Checkbox 30
5. Nút lệnh (Button) 30
6. Combo Box (Drop-down menu) 31
7. Listbox 31
8. Hộp nhập văn bản nhiều dòng (TextArea) 32
IX. Một số thẻ đặc biệt 34
1. Thẻ <meta> 34
2. Thẻ <marquee> 35
3. Thẻ <style> 36
4. Thẻ <link> 36
5. Thẻ <script> 37
Chương 3: Thiết kế CSS 38
I. Giới thiệu về CSS 38
II. Cú pháp 39
1. Định dạng thuộc tính thẻ html 39
2. Định dạng một kiểu mới 40
3. Định dạng ngay trong thẻ html 41
III. Sử dụng css trong tài liệu HTML 41
1. CSS được khai báo trong một tập tin riêng 41
2. Định dạng ngay trên tài liệu html 41
IV. Một số thuộc tính thường dùng 42
1. Định kiểu nền 42
a. Màu nền 42
b. Ảnh nền 42
2. Định kiểu chữ 44
a. Màu chữ 44
b. Canh lề: 44
c. Trang trí chữ 44
d. Chuyển đổi chữ hoa/thường 45
e. Thuộc tính letter-spacing: 45
3. Định kiểu font 45
a. Tên font (font-family) 45
b. Kiểu font (font style) 46
c. Cỡ font (font size) 47
d. Thuộc tính font-weight: 48
4. CSS Link 49
5. Định kiểu danh sách 49
6. Định kiểu bảng 51
a. Border: 51

b. Width: 53
c. Height: 53
d. Text-align: 54
e. Vertical-align: 54
f. Padding: 55
g. Background-color: 55
h. Color: 55
7. Thuộc tính Id và class của thẻ 56
a. Thuộc tính Id 56
b. Thuộc tính Class 57
8. Mô hình hộp 58
a. Thuộc tính margin: 59
b. Thuộc tính padding 59
c. Border 61
d. Thuộc tính Width và Height 61
e. Thuộc tính float và clear 62
Chương 4: Giới thiệu ngôn ngữ kịch bản Javascript 65
I. Giới thiệu về Javascript 65
II. Ngôn ngữ javascript 65
1. Chèn mã lệnh javascript vào trong tài liệu HTML 65
a. Chèn mã lệnh trên vùng <body> 65
b. Chèn mã lệnh trên vùng <head> 65
c. Chèn mã lệnh trực tiếp vào trong các thẻ HTML 65
d. Chèn mã lệnh bằng một tập tin riêng trên vùng <head> 66
2. Lời chú thích 66
3. Biến và cách xuất thông tin lên trình duyệt 66
a. Biến và cách khai báo biến 66
b. Xuất thông tin lên trình duyệt web 67
4. Các phép toán 67
5. Câu lệnh rẽ nhánh If...Else 69
6. Câu lệnh lựa chọn Switch 71
7. Định nghĩa hàm 72
8. Hộp thông báo 72
9. Câu lệnh lặp For 74
10. Câu lệnh lặp While 75
11. Câu lệnh lặp For...In 76
12. Sự kiện trong Javascript 77
a. Sự kiện onLoad và onUnload 77
13. Câu lệnh Try...Catch 77
14. Câu lệnh Throw 78
15. Ký tự đặc biệt Text 79
III. Đối tượng trong javascrip 80
1. Đối tượng String 80
2. Đối tượng Date 81
3. Đối tượng Array 81
4. Đối tượng Math 82
Chương 5: Ngôn ngữ PHP 83
I. Tổng quan về PHP 83
1. Cú pháp PHP 83

2. Xuất giá trị ra trình duyệt 83
3. Lời chú thích 84
4. Biến trong PHP 84
a. Khai báo biến 84
b. Gán giá trị cho biến. 85
c. Phạm vi hoạt động của biến 85
5. Hằng 87
a. Khái báo hằng 87
b. Sử dụng hằng 88
6. Kiểu dữ liệu 88
a. Kiểu dữ liệu 88
b. Chuyển đổi kiểu dữ liệu 90
7. Các toán tử 90
a. Toán tử toán học 90
b. Toán tử nối chuỗi 90
c. Toán tử gán kết hợp 90
d. Toán tử so sánh 91
e. Toán tử logic 91
f. Toán tử @ 91
g. Tham chiếu & 92
8. Các hàm kiểm tra giá trị 92
a. Kiểm tra tồn tại isset() 92
b. Kiểm tra giá trị rỗng empty() 93
c. Kiểm tra giá trị số is_numeric() 94
d. Kiểm tra kiểu giá trị của tên biến 94
e. Xác định kiểu dữ liệu biến 96
II. Câu lệnh điều khiển 96
1. Câu lệnh rẽ nhánh If..Else 96
2. Câu lệnh lựa chọn switch 98
3. Câu lệnh lặp 98
a. Cấu trúc for/foreach 98
b. Cấu trúc while 99
c. Cấu trúc do ... while 99
4. Sử dụng break và continue trong cấu trúc lặp 100
a. Lệnh break 100
b. Lệnh continue 100
5. Kiểu mảng 101
a. Khái niệm mảng 101
b. Khai báo mảng và sử dụng mảng 101
c. Truy xuất phần tử mảng. 102
d. Các thao tác trên mảng 102
e. Một số hàm 103
III. Xây dựng hàm trong PHP 105
1. Hàm do người dùng định nghĩa 105
a. Khai báo hàm 105
b. Sử dụng hàm 106
2. Hàm trong thư viện hàm 107
a. Kiểu dữ liệu string 107
b. Kiểu dữ liệu số 111

c. Kiểu dữ liệu ngày, giờ ... 114
IV. Biểu mẫu form ... 116
1. Đặc điểm form ... 116
2. Biểu mẫu sử dụng phương thức $_POST ... 116
3. Biểu mẫu sử dụng phương thức $_GET ... 118
Chương 6: Hướng đối tượng trong PHP ... 119
I. Khái niệm ... 119
II. Tạo lớp ... 119
III. Sử dụng lớp ... 120
IV. Kế thừa ... 122
Chương 7: Tạo web động ... 123
I. Sử dụng tập tin dùng chung ... 123
1. REQUIRE ... 123
2. INCLUDE ... 125
II. Mở tập tin và thư mục ... 126
1. Tập tin ... 126
a. Chế độ mở tập tin ... 126
b. Mở tập tin ... 127
c. Đọc tập tin ... 127
d. Định dạng tập tin. ... 129
e. Ghi nội dung tập tin ... 129
f. Đóng tập tin. ... 129
g. Kiểm tra sự tồn tại của tập tin ... 129
h. Kiểm tra kích thước file ... 130
k. Xóa tập tin ... 130
2. Thư mục ... 131
a. Tạo thư mục ... 131
b. Kiểm tra thư mục ... 131
c. Mở thư mục ... 131
d. Đóng thư mục ... 132
e. Duyệt thư mục ... 132
III. Upload tập tin lên server. ... 132
1. Giới thiệu ... 132
2. Các bước upload file ... 132
IV. PHP Cookies ... 134
1. Khái niệm ... 134
2. Khai báo cookie ... 134
3. Sử dụng cookie ... 135
4. Hủy cookie ... 135
V. PHP Sessions ... 135
1. Khái niệm ... 135
2. Cách thức hoạt động ... 135
3. Khởi động Session ... 136
4. Đặt ký Session ... 136
5. Sử dụng Session ... 136
6. Hủy biến Session ... 136
a. Hủy toàn bộ các biến session ... 136
b. Hủy một biến session ... 136
VI. Gửi E-mail trong PHP ... 137

Ví dụ: Lấy thông tin từ Form 137
Chương 8: CƠ SỞ DỮ LIỆU MYSQL 138
I. Tổng quan 138
1. Giới thiệu CSDL 138
a. Khái niệm 138
b. Chức năng. 138
c. Các loại CSDL 138
d. Các đối tượng chính của CSDL 139
e. Hệ quản trị CSDL 139
f. SQL (Structure Query Language) 140
2. CSDL MySQL 140
a. Giới thiệu 140
b. Đặc điểm 140
c. Các tập tin vật lý lưu trữ CSDL 140
d. Quy tắc đặt tên cho CSDL, bảng, chỉ mục, cột và định danh 140
e. Tạo CSDL 141
f. Xóa CSDL 143
II. Bảng(Table) 144
1. Khái niệm 144
2. Thuộc tính 144
a. Tên bảng 144
b. Các thuộc tính của cột trong bảng 144
3. Thao tác với bảng 146
a. Tạo bảng 146
b. Thay đổi cấu trúc bảng 148
c. Sửa đổi kiểu dữ liệu của cột 148
e. Hủy cột trong bảng 149
f. Xóa bảng 149
III. Bảng ảo 149
1. Khái niệm 149
2. Tạo bảng ảo 149
3. Cập nhật nội dung bảng ảo 150
4. Xóa bảng ảo 151
IV. Toán tử 151
1. Khái niệm 151
2. Toán tử số học 151
3. Toán tử so sánh 152
4. Toán tử logic 152
V. Phát biểu SQL 152
1. Câu lệnh SELECT 152
a. Truy vấn đơn giản SELECT … FROM 152
b. Truy vấn có sắp xếp dữ liệu 152
c. Truy vấn có điều kiện WHERE 153
d. Nhóm dữ liệu GROUP BY 153
e. Điều kiện lọc nhóm HAVING 154
f. Giới hạn mẫu tin LIMIT 154
2. Truy vấn con 154
a. Truy vấn con trả về giá trị 155
b. Truy vấn con trả về danh sách các giá trị 155

c. Làm việc với các toán tử so sánh 155
d. Làm việc với toán tử truy vấn con 155
3. Câu lệnh thêm dữ liệu 155
a. Giá trị trực tiếp 155
b. Lấy từ nguồn dữ liệu 156
4. Câu lệnh cập nhật dữ liệu 156
a. Giá trị trực tiếp 156
b. Lấy dữ liệu từ các bảng khác 156
5. Câu lệnh xóa dữ liệu 156
a. Câu lệnh xóa dữ liệu đơn giản 157
b. Câu lệnh xóa dữ liệu có điều kiện được lấy từ bảng khác. 157
6. Sử dụng mệnh đề UNION trong truy vấn 157
7. Truy vấn dữ liệu từ nhiều bảng 157
a. INNER JOIN 157
b. LEFT JOIN, RIGHT JOIN 158
c. Mệnh đề liên kết dữ liệu nhiều bảng 158
8. Sử dụng hàm trong SQL 158
a. Các hàm cấu trúc điều khiển 158
b. Các hàm chuyển đổi kiểu dữ liệu 159
c. Hàm xử lý chuỗi 159
d. Các hàm sử lý số 159
e. Các hàm xử lý thời gian 160
9. Import và export dữ liệu 160
a. Import dữ liệu 160
b. Export dữ liệu 160
Chương 9: PHP&MYSQL 161
I. Kết nối CSDL 161
1. Tạo kết nối 161
2. Chọn CSDL 161
3. Truy vấn dữ liệu 162
4. Thông báo lỗi 163
5. Đóng kết nối 164
II. Làm việc với CSDL MySQL 164
1. Đếm số lượng mẫu tin 164
2. Hiển thị dữ liệu 165
a. Duyệt dữ liệu 165
b. Hiển thị dữ liệu không định dạng 167
c. Hiển thị dữ liệu có yêu cầu 168
d. Chuyển đổi giá trị thời gian 169
3. Lưu trữ dữ liệu mới vào CSDL 170
4. Cập nhật dữ liệu 172
5. Xóa dữ liệu 173
III. PHP kết hợp với các CSDL SQL Server 173
IV. Xây dựng các lớp xử lý 174
1. Một số phương thức trong lớp xử lý bảng 174
2. Xây dựng lớp xử lý nghiệp vụ 176
Mục lục 178

Bài học lập trình PHP cơ bản và ứng dụng

Mục lục

Bài 1: Giới thiệu, Cấu hình và cài đặt PHP 1
Bài 2: Kiến thức tổng quan về lập trình PHP 3
Bài 3: Toán tử và biểu thức trong PHP 5
Bài 4: Xử lý giá trị form trong PHP 9
Bài 6: Tổng quan về mảng và các hàm hỗ trợ trong PHP 12
Bài 7: Khái niệm cơ bản về Cookie và session trong PHP 14
Bài 8: Khái niệm về hàm và gọi lại hàm trong PHP 17
Bài 9: Cơ bản về ngôn ngữ SQL và Mysql 20
Bài 10: Kết hợp PHP và MYSQL trong ứng dụng website 23
Bài 11: Viết ứng dụng đăng nhập bằng PHP và MYSQL 25
Bài 12: Viết ứng dụng thêm thành viên bằng PHP và MYSQL 28
Bài 13: Viết ứng dụng quản lý thành viên bằng PHP và MYSQL 32
Bài 14: viết ứng dụng sửa xóa thành viên bằng PHP và MYSQL 35
Bài 15: Viết ứng dụng tạo mã xác nhận bằng PHP 40
Bài 16: Viết ứng dụng đếm số người online bằng php 44
Bài 17: Xây dựng hệ thống bình chọn bằng PHP và MYSQL 46
Bài 18: Xây dựng ứng dụng tạo trang upload nhiều hình ảnh 52
Bài 19: Viết ứng dụng tạo giỏ hàng - shopping cart - phần 1 57
Bài 20: Viết ứng dụng tạo giỏ hàng - shopping cart - phần 2 62

Bài 1: Giới thiệu, Cấu hình và cài đặt PHP

PHP (Hypertext Preprocessor) là ngôn ngữ script trên server được thiết kế để dễ dàng xây dựng các trang Web động. Mã PHP có thể thực thi trên Webserver để tạo ra mã HTML và xuất ra trình duyệt web theo yêu cầu của người sử dụng.

Ngôn ngữ PHP ra đời năm 1994 Rasmus Lerdorf sau đó được phát triển bởi nhiều người trải qua nhiều phiên bản. Phiên bản hiện tại là PHP 5 đã được công bố 7/2004.

Có nhiều lý do khiến cho việc sử dụng ngôn ngữ này chiếm ưu thế xin nêu ra đây một số lý do cơ bản :

- Mã nguồn mở (open source code)
- Miễn phí, download dễ dàng từ Internet.
- Ngôn ngữ rất dễ học, dễ viết.
- Mã nguồn không phải sửa lại nhiều khi viết chạy cho các hệ điều hành từ Windows, Linux, Unix
- Rất đơn giản trong việc kết nối với nhiều nguồn DBMS, ví dụ như : MySQL, Microsoft SQL Server 2000, Oracle, PostgreSQL, Adabas, dBase, Empress, FilePro, Informix, InterBase, mSQL, Solid, Sybase, Velocis và nhiều hệ thống CSDL thuộc Hệ Điều Hành Unix (Unix dbm) cùng bất cứ DBMS nào có sự hỗ trợ cơ chế ODBC (Open Database Connectivity) ví dụ như DB2 của IBM.

Cài đặt PHP, Apache, MYSQL

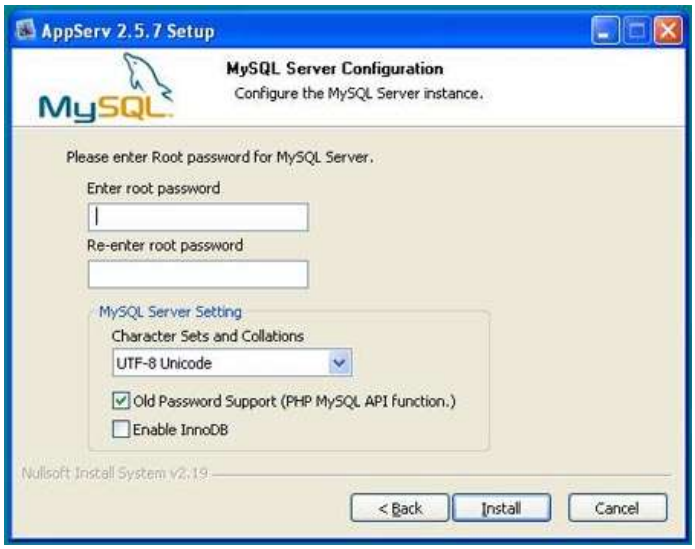
Để cài đặt PHP, các bạn có thể cài đặt từng gói riêng lẻ (PHP, Apache, Mysql). Tuy nhiên, tôi vẫn khuyến khích các bạn cài đặt dạng gói tích hợp. Sẽ tiện lợi cho việc sau này hơn rất nhiều.

Gói phần mềm mà tôi chọn là : Appserv - Win - 2.5.8 (Tên phần mềm là Appserv, dành cho phiên bản window, phiên bản 2.5.8).

Bước 1: Bạn cài đặt bình thường bằng cách nhấp vào file exe.

Bước 2: Phần mềm sẽ cho bạn chọn cần cài những module nào. Hãy giữ nguyên tất cả như hình dưới và nhấn next.

Bước 3: Trong giao diện dưới:



Server Name: bạn nhập vào: localhost
Email: Bạn nhập vào email của bạn:
Port: Mặc định là 80, tuy nhiên nếu máy bạn đã cài IIS thì có thể chỉnh thành 81 để chạy dịch vụ bên appserv.
Nhấn next để qua trang kế tiếp.

Bước 4: Trong giao diện bên dưới ta điền thông tin như sau:

Enter root password: Bạn gõ vào root

+ Echo "Thông tin";
+ Printf "Thông tin";
Thông tin bao gồm : biến, chuỗi, hoặc lệnh HTML ….

```php
<?php
Echo "Hello word";
Printf"<br><font color=red>Who Are You ?</font>";
?>
```

Nếu giữa hai chuỗi muốn liên kết với nhau ta sử dụng dấu "."

```php
<?php
Echo "Hello"."who are you ?";
?>
```

3- Khái niệm biến, hằng, chuỗi và các kiểu dữ liệu.

a) Biến trong PHP.

Biến được xem là vùng nhớ dữ liệu tạm thời. Và giá trị có thể thay đổi được. Biến được bắt đầu bằng ký hiệu "$". Và theo sau chúng là 1 từ, 1 cụm từ nhưng phải viết liền hoặc có gạch dưới.
1 biến được xem là hợp lệ khi nó thỏa các yếu tố :
+ Tên của biến phải bắt đầu bằng dấu gạch dưới và theo sau là các ký tự, số hay dấu gạch dưới.
+ Tên của biến không được phép trùng với các từ khóa của PHP.
Trong PHP để sử dụng 1 biến chúng ta thường phải khai báo trước, tuy nhiên đối với các lập trình viên khi sử dụng họ thường xử lý cùng một lúc các công việc, nghĩa là vừa khái báo vừa gán dữ liệu cho biến.
Bản thân biến cũng có thể gãn cho các kiểu dữ liệu khác. Và tùy theo ý định của người lập trình mong muốn trên chúng.
Một số ví dụ về biến :

```php
<?
$a= 100 // biến a ở đây có giá trị là 100.
$a= "PHP is easy" // Biến a ở đây có giá trị "PHP Is easy".
Biena=123 //Có lỗi vì bắt đầu 1 biến phải có dấu "$"
$123a="PHP" //Có lỗi vì phần tên bắt đầu của biến là dạng số.
?>
```

b) Khái niệm về hằng trong PHP.

Nếu biến là cái có thể thay đổi được thì ngược lại hằng là cái chúng ta không thể thay đổi được. Hằng trong PHP được định nghĩa bởi hàm define theo cú pháp: define (string tên_hằng, giá_trị_hằng).
Cũng giống với biến hằng được xem là hợp lệ thì chúng phải đáp ứng 1 số yếu tố :
+ Hằng không có dấu "$" ở trước tên.
+ Hằng có thể truy cập bất cứ vị trí nào trong mã lệnh
+ Hằng chỉ được phép gán giá trị duy nhất 1 lần.
+ Hằng thường viết bằng chữ in để phân biệt với biến
Ví dụ :

```php
<?
define ("C", "COMPANY");
define ("YELLOW", "#ffff00");
echo "Gia tri cua C la". C;
?>
```

c) Khái niệm về chuỗi:

Chuỗi là một nhóm các kỹ tự, số, khoảng trắng, dấu ngắt được đặt trong các dấu nháy.

Toán Tử	Lý giải	Ví dụ	Kết quả
+	Cộng hai số hạng	10+8	18
-	Trừ hai số hạng	10-8	2
*	Nhân hai số hạng	10*8	80
/	Chia hai số hạng	10/3	3.33333333
%	Trả về số dư	10%3	1

3- Toán tử so sánh:
Là toán tử được sử dụng để thực hiện các phép toán so sánh giữa hai số hạng. Chi tiết, xem bảng bên dưới.

Phép Toán	Tên	Giải Thích	Ví Dụ
==	Bằng	Hai số hạng bằng nhau	$a==5
!=	Không bằng	Hai số hạng không bằng nhau	$a != 5
===	Đồng nhất	Hai số bằng nhau và cùng kiểu	$a === 5
>	Lớn hơn	Vế trái lớn hơn vế phải	$a > 5
>=	Lớn hơn hoặc bằng	Vế trái lớn hơn hoặc bằng vế phải	$a >= 5
<	Nhỏ hơn	Vế trái nhỏ hơn vế phải	$a < 5
<=	Nhỏ hơn hoặc bằng	Vế trái nhỏ hơn hoặc bằng vế phải	$a <= 5

4- Toán tử logic:
Toán tử logic là các tổ hợp các giá trị boolean.
Ví dụ: toán tử or trở về true nếu toán tử trái hoặc toán tử phải là true.
True || false à true.
Ta có bảng các toán tử như sau:

Toán Tử	Tên	Trả về True Nếu	Ví dụ	Kết quả
||	Or	Vế trái hoặc vế phải là True	True || false	True
Or	Or	Vế trái hoặc vế phải là True	True || false	True
Xor	Xor	Vế trái hoặc vế phải là True nhưng không phải cả hai	True || True	False
&&	And	Vế trái và vế phải là true	True && false	False
And	And	Vế trái và vế phải là true	True && false	False
!	Not	không phải là true	!true	False

5- Toán tử kết hợp:
Khi tạo mã PHP, chúng ta sẽ thường nhận thấy cần phải tăng hoặc giảm lượng biến một số nguyên nào đó. Bạn sẽ thường thực hiện điều này khi chúng ta đếm 1 giá trị nào đó trong vòng lặp.

Phép Toán	Ví dụ	Lý Giải
++	$a++	Bằng với $a= $a + 1
--	$a--	Bằng với $a= $a – 1
+=	$a+=$b	Bằng với $a= $a + $b
-=	$a-=$b	Bằng với $a= $a - $b
*=	$a*=$b	Bằng với $a= $a * $b
/=	$a/=$b	Bằng với $a= $a / $b

B- Các biểu thức cơ bản trong PHP:

1- Biểu thức điều kiện:

Là biểu thức dùng kiểm tra 1 sự kiện. Nếu chúng thỏa điều kiện đó thì sẽ thực thi một hành động. Ngược lại sẽ là một hành động khác.

Cú pháp:

If(Điều kiện)
{
hành động
}

Ví dụ:

```php
<?php
$a=5;
$b=7;
if($a < $b)
{
    echo "Bien A co gia tri nho hon bien B";
}
else
{
    echo "Bien A co gia tri lon hon bien B";
}
?>
```

2- Vòng lặp trong PHP:

a- While()....

Phép lặp này yêu cầu phải thỏa mãn điều kiện thì mới thực thi được vòng lặp

Cú pháp:

While(điều kiện)
{
Hành động – thực thi
}

Ví dụ:

```php
<?php
$a=1;
while($a < 10)
{
    echo "gia tri $a la";
    $a++;
}
?>

```

b-Do....while():

Phép lặp này sẽ thực thi hành động ít nhất là một lần. Sau đó mới tiến hành kiểm tra điều kiện.

Cú pháp:

Do
{

Hành động thực thi
}while(điều kiện)
Ví dụ:

```php
<?php
$a=5;
do{
echo "day la gia tri cua $a";
$a++;
}while($a >6)
?>
```

c- For():
Phép lặp này là phép toán gộp các tham số. Giúp người lập trình giảm thiểu thời gian phải khai báo biến và các tham số khi thực thi việc lặp dữ liệu.
Cú pháp:
For(giá trị ; điều kiện ; biến tăng hoặc giảm)
{ Hành động }
Ví dụ:

```php
<?php
$a=2;
for($i=1; $i<=10; $i++)
{
    echo"$a x $i =".$a*$i."<br>";
}
?>

```

3- Biểu thức switch case:
Là biểu thức sử dụng để giảm thiểu quá trình xử lý dữ liệu nếu có quá nhiều phép toán if else.
Cú pháp:
Switch(biến)
{
Case giá trị 1: Hành động; Break;
……
Case giá trị N: Hành động; Break;
Default: Hành động; Break;
}
Ví dụ:

```php
<?php
$a=5;
switch($a)
{
    case 1: echo "day la gia tri $a"; break;
    case 2: echo "day la gia tri $a"; break;
    case 3: echo "day la gia tri $a"; break;
    case 4: echo "day la gia tri $a"; break;
    case 5: echo "<font color=red>day la gia tri $a</font>"; break;
    default: echo "Khong co gia tri phu hop"; break;

}
?>
```

C- Tổng kết:
Kết thúc bài học này, các bạn ít nhiều đã nắm được những thuộc tính cơ bản của các phép toán học trong PHP, ngoài ra chúng ta cũng từng bước hiểu được cú pháp của từng biểu thức.
Đối với các vòng lặp, chúng ta cần hiểu và vận chúng một cách linh hoạt và mềm dẻo, phân biệt được while và do…while. Sử dụng hàm switch trong trường hợp có quá nhiều giá trị if…else trả về.
D- Bài tập áp dụng:
Bài tập 1:
Viết 1 trang web có giá trị từ 1->20. Hãy xuất ra trình duyệt những số chẵn nằm trong khoảng 1->20 đó.
Bài tập 2:
Xây dựng 1 website thỏa yêu cầu xuất ra bảng cửu chương từ 2 ->10.

```
<html>
<form action = "processform.php" method = post>
Please type your name here : <BR>
<input type = text name = "username"><BR><BR>
<input type = submit value = "Submit Data">
</form>
</html>
```

Tạo file processform.php để xuất ra dữ liệu

```
<?
echo "welcome, " . $_POST['username'] . "!";
?>
```

Tổng Kết:

Kết thúc bài này các bạn đã nắm được kỹ thuật kiểm tra thông tin dựa trên PHP. Một trong những vấn đề không thể thiếu đối với bất kỳ một website động nào. Chúng cho ta kiểm tra tính hợp lệ của người sử dụng 1 cách dễ dàng bởi sự tùy biến trong các biểu thức của PHP.

Bài tập áp dụng :

Tạo 1 trang web với hộp thoại nhập liệu username và password. Nếu người sử dụng nhập thông tin username/password là admin/12345 thì xuất ra thông báo "welcome, admin" với kiểu chữ Tahoma, màu đỏ. Ngược lại nếu nhập sai thì xuất thông báo "Username hoặc password sai. Vui lòng nhập lại".

Đáp án:

Tạo trang login.html với nội dung sau:

```
<html>
<head>
<title>login page</title>
</head>
<body>
<form action="checklogin.php" method=POST>
Username <input type=text name="user" size=15>
<br>Password <input type=password name="pass" size=15>
<br><input type=submit name=submit value="Login">
</form>
</body>
</html>
```

Tiếp tục tạo trang checklogin.php với nội dung sau:

```
<?php
$username=$_POST['user'];
$password=$_POST['pass'];
if($username == "admin" && $password="12345")
{
    echo "<font color=red>Welcome to, ".$username."<font>";
}
else
{
    echo "<font color=red>Username hoac password khong chinh xac,
    vui long dang nhap lai<font>";
}
?>
```

Bài 5: Tìm hiểu quy trình làm việc trên file trong PHP

Một trong những tác vụ đặc biệt của PHP đó là cho phép xử lý dữ liệu trực tiếp thông qua quá trình nhận và đọc nội dung trên 1 file dữ liệu. Điều này giúp cho PHP trở nên tinh tế và dễ tùy biến hơn khi xử lý 1 lượng dữ liệu có quy mô lớn. Việc thao tác mở, đọc, ghi, đóng file này cũng có ý nghĩa tuần tự như bạn đang làm việc trực tiếp trên 1 file dữ liệu thực thụ.

1- Đóng, mở 1 file trong PHP:

Để mở 1 file ta sử dụng cú pháp sau: fopen("Đường dẫn", thuộc tính).
Trong đó Đường dẫn chính là đường dẫn tới file cần mở.
Thuộc tính bao gồm các quyền hạn cho phép thao tác trên file đó như thế nào.
Các thuộc tính cơ bản :

Mode	Diễn giải
r	Read only
r+	Read_Write
w	Write Only
w+	Write_Read. Nếu file này tồn tại, nội dung cũ sẽ bị xoá đi, còn nếu không tồn tại nó sẽ được tạo mới
a	Mở dưới dạng append dữ liệu, chỉ có write, nếu file tồn tại, sẽ ghi tiếp vào phần dưới của nội dung có sẵn, nếu file không tồn tại sẽ được tạo mới.
a+	Mở dưới dạng append dữ liệu (write và read), nếu file tồn tại, dữ liệu sẽ ghi tiếp vào phần bên dưới của nội dung cũ, nếu file không tồn tại sẽ được tạo mới
b	Mở dưới chế độ file binary

Ví dụ:

```
<?php
$fp=fopen("test.txt",r)or exit("khong tim thay file can mo");
?>
```

Tương tự như thế, để đóng 1 file ta có cú pháp như sau: fclose(file vừa mở)
Ví dụ:

```
<?php
$fp=fopen("test.txt",r)or exit("khong tim thay file can mo");
fclose($fp);
?>
```

Việc mở và đóng này không có ý nghĩa là chúng đã được đọc. Muốn đọc được nội dung của file chúng ta lại tiếp tục với thao tác lấy dữ liệu từ file nữa.

2- Đọc và ghi file trong PHP.

a) Đọc 1 file trong PHP

PHP cho ta nhiều sự lựa chọn trong việc đọc 1 file. Có nhiều hình thức hỗ trợ nhưng hiện nay 2 hình thức phổ biến nhất vẫn là đọc file theo từng dòng và đọc file theo từng ký tự.
- Đọc file theo từng dòng:
Cú pháp : fgets(file vừa mở).
Ví dụ:

```
<?php
$fp=fopen("test.txt",r)or exit("khong tim thay file can mo");
```

mảng luôn bắt đầu với giá trị là 0.

1- Định nghĩa mảng trong PHP:

Với cú pháp giống với java ta có cách gọi 1 mảng như sau:
$tên_biến= array("Kenny","Maria","Julia","Kenvin")
Như vậy khi gọi giá trị trong mảng ta có:
Echo $tên_biến[2]; // Giá trị sẽ cho là Julia.
Ngoài cách khai báo ở trên chúng ta cũng có thể khai báo bằng cách khác như sau:
$tên_biến[] = "Kenny";
$tên_biến[] = "Gillian";
$tên_biến[] = "Charlene";
$tên_biến[] = "Calvin"
Chúng ta không cần thêm ký tự vào. Vì PHP đã tự xử lý cho chúng ta phần đó.
Để thêm 1 phần tử vào mảng chúng ta cũng có thể sử dụng như sau:
$tên_biến[] = "Jiro";
Như vậy nếu chúng ta gọi giá trị: echo $tên_biến[4] thì giá trị sẽ là jiro. Vì nó được PHP thêm vào sau cùng và ngầm hiểu là giá trị tiếp theo.
Ví dụ:

```
<?php
$a=array("Kenny","Maria","Julia","Kenvin");
echo $a[2]; //Xuất ra giá trị Julia
?>
```

2- Khái niệm mảng kết hợp trong PHP

Là các mảng được tạo index bằng các chuỗi , chúng được gọi là các mảng kết hợp.
Ví dụ:

```
<?
$a= array (name => "Kenny", job => "Teacher", age=>"45", email =>
"webmaster@vietchuyen.com.vn")
?>
```

Như vậy để truy cập giá trị của mảng trên chúng ta không thể sử dụng cách cũ được.
Ví dụ: Để lấy giá trị của tuổi tôi không thể sử dụng cách trên : **echo $a[2].**
Vì chúng ta đã gán giá trị của index cho 1 tên gọi khác. Do vậy để lấy giá trị của tuổi ta sử dụng như sau: **echo $a[age]** // Kết quả sẽ cho ra 45.

```
<?
$tên_biến= array (name => "Kenny", job => "Teacher", age=>"45", email =>
"webmaster@vietchuyen.com.vn")
echo $a[age];
?>
```

3- Phép lặp trong mảng:

Cú pháp:
foreach($array as $temp)
{ Hành Động }
Trong đó $array là mảng mà ta muốn thực thi việc lặp dữ liệu, và $temp là một biến trong đó ta sẽ tạm thời lưu trữ mỗi phần tử.
Ví dụ:

```
<?php
$name= array("Kenny", "Jiro", "Calvin", "Gillian");
foreach ($name as $test)
{ echo "$test<br>"; }
?>
```

1- Tổng quan về cookie:

Cookie là 1 đoạn dữ liệu được ghi vào đĩa cứng hoặc bộ nhớ của máy người sử dụng. Nó được trình duyệt gởi ngược lên lại server mỗi khi browser tải 1 trang web từ server.
Những thông tin được lưu trữ trong cookie hoàn toàn phụ thuộc vào website trên server. Mỗi website có thể lưu trữ những thông tin khác nhau trong cookie, ví dụ thời điểm lần cuối ta ghé thăm website, đánh dấu ta đã login hay chưa, v.v...
Cookie được tạo ra bởi website và gởi tới browser, do vậy 2 website khác nhau (cho dù cùng host trên 1 server) sẽ có 2 cookie khác nhau gởi tới browser. Ngoài ra, mỗi browser quản lý và lưu trữ cookie theo cách riêng của mình, cho nên 2 browser cùng truy cập vào 1 website sẽ nhận được 2 cookie khác nhau.

a -Thiết lập cookie:

Để thiết lập cookie ta sử dụng cú pháp:
Setcookie("tên cookie","giá trị", thời gian sống)
Tên cookie là tên mà chúng ta đặt cho phiên làm việc.
Giá trị là thông số của tên cookie.
Ví dụ:

Setcookie("username","admin", time() +3600)
Như ví dụ trên ta thấy với tên là username và giá trị là admin, có thời gian sống là 1 giờ tính từ thời điểm thiết lập.
Chú ý: Kịch bản cookie phải đặt trên mọi giá trị trả về bao gồm thẻ HTML và lệnh echo.
Kịch bản cookie phải đặt trên mọi giá trị trả về bao gồm thẻ HTML và lệnh echo.

b - Sử dụng cookie:

Để sử dụng lại cookie vừa thiết lập, chúng ta sử dụng cú pháp:
Cú pháp: $_COOKIE["tên cookies"]
Tên cookie là tên mà chúng ta thiết lập phía trên.
Ví dụ:
Tạo trang cookie.php với nội dung sau:

```
<?php
setcookie("name","Kenny Huy",time() + 3600);
?>
<html>
<head>
<title>Test page 1</title></head>
<body>
<b><a href=cookie2.php>Click here</a></b>
</body>
</html>
```

Tiếp tục tạo trang cookie2.php với nội dung sau:

```
<html>
<head><title>Result Page</title></head>
<body>
<?php
echo "Ten cua ban la <b>".$_COOKIE['name']."</b>";
?>
</body>
</html>
```

c- Hủy Cookie:

Để hủy 1 cookie đã được tạo ta có thể dùng 1 trong 2 cách sau:
+ Cú pháp: setcookie("Tên cookie")

Gọi hàm setcookie với chỉ duy nhất tên cookie mà thôi
+ Dùng thời gian hết hạn cookie là thời điểm trong quá khứ.
Ví dụ: setcookie("name","Kenny Huy",time()-3600);
Ví dụ:
Tiếp tục tạo trang cookie3.php với nội dung sau:

```
<?php
setcookie("name","Kenny Huy",time()-360);
?>
<html>
<head>
<title>Test page 1</title></head>
<body>
<b><a href=cookie2.php>Click here</a></b>
</body>
</html>
```

2- Tổng quan về session:

Một cách khác quản lý người sử dụng là session. Session được hiểu là khoảng thời gian người sử dụng giao tiếp với 1 ứng dụng. Một session được bắt đầu khi người sử dụng truy cập vào ứng dụng lần đầu tiên, và kết thúc khi người sử dụng thoát khỏi ứng dụng. Mỗi session sẽ có được cấp một định danh (ID) khác nhau và nội dung được lưu trong thư mục thiết lập trong file php.ini (tham số session.save_path).

a- Thiết lập session:

Để thiết lập 1 session ta sử dụng cú pháp: session_start()
Đoạn code này phải được nằm trên các kịch bản HTML. Hoặc những lệnh echo, printf.
Để thiết lập 1 giá trị session, ngoài việc cho phép bắt đầu thực thi session. Chúng ta còn phải đăng ký 1 giá trị session. Để tiện cho việc gán giá trị cho session đó.
Ta có cú pháp sau: session_register("Name")
Ví dụ:

```
<?php
Session_start();
Session_register("username");
?>
```

b- Sử dụng giá trị của session:

Giống với cookie. Để sử dụng giá trị của session ta sử dụng mã lệnh sau:
Cú pháp: $_SESSION["name"]
$_SESSION["name"]
Với Name là tên mà chúng ta sử dụng hàm session_register("name") để khai báo.

Ví dụ:
Tạo trang session.php với nội dung sau:

```
<?php
session_start();
session_register("name");
$_SESSION["name"] = "Kenny Huy";
?>
<html>
<head>
<title>Test page 1</title></head>
<body>
<b><a href=session2.php>Click here</a></b>
</body>
</html>
```

Cú pháp:
function functiono_name()
{
//Lệnh thực thi
}
Tên hàm có thể là một tổ hợp bất kỳ những chứ cái, con số và dấu gạch dưới, nhưng phải bắt đầu từ chứ cái và dấu gạch dưới.

Ví dụ:
```
<?php
function name()
{
$name= array("Kenny", "Jiro", "Calvin", "Gillian");
foreach ($name as $test)
{ echo "$test<br>"; }
}
name();
?>
```

2- Hàm tự định nghĩa với các tham số:
Cú pháp:
function function_name($gt1,$gt2)
{
//Hành động
}

Ví dụ:

```
<?php
function indulieu($text)
{
echo "$text<br>";
}
indulieu("welcome");
indulieu("who are you ?");
?>
```

3- Hàm tự định nghĩa với giá trị trả về:
Cú pháp:
function function_name(Có hoặc không có đối số)
{
//Thuc thi
return giatri;
}

Ví dụ:
```
<?php
function tinhtong($a,$b)
{
$total=$a+$b;
return $total;
}
echo tinhtong(19,31)
```

Bài 9: Cơ bản về ngôn ngữ SQL và Mysql

Mysql là hệ quản trị dữ liệu miễn phí, được tích hợp sử dụng chung với apache, PHP. Chính yếu tố phát triển trong cộng đồng mã nguồn mở nên mysql đã qua rất nhiều sự hỗ trợ của những lập trình viên yêu thích mã nguồn mở. Mysql cũng có cùng một cách truy xuất và mã lệnh tương tự với ngôn ngữ SQL. Nhưng Mysql không bao quát toàn bộ những câu truy vấn cao cấp như SQL. Về bản chất Mysql chỉ đáp ứng việc truy xuất đơn giản trong quá trình vận hành của website nhưng hầu hết có thể giải quyết các bài toán trong PHP.

1- Cách khởi động và sử dụng MYSQL.

Chúng ta sử dụng command như sau:

Mysql –hname –uuser –ppass

Để truy cập vào cơ sở dữ liệu.
Hoặc sử dụng bộ appserv để vào nhanh hơn theo đường dẫn sau:
Start/ Appserv/ Mysql command Line client
Sau đó nhập password mà chúng ta đã đặt vào.

2- Những định nghĩa cơ bản:

a) Định nghĩa cơ sở dữ liệu, bảng, cột:

Cơ sở dữ liệu: là tên của cơ sở dữ liệu chúng ta muốn sử dụng
Bảng: Là 1 bảng giá trị nằm trong cơ sở dữ liệu.
Cột là 1 giá trị nằm trong bảng. Dùng để lưu trữ các trường dữ liệu.
Thuộc tính
Ví dụ:

Bảng user	
User_id	1234
First_name	Nguyen
Last_name	Hai Anh
Username	Kenny
Password	12345
Email	kenny@qhonline.info
Date	2007-12-12 16:25:30

Như vậy ta có thể hiểu như sau:
1 cơ sở dữ liệu có thể bao gồm nhiều bảng.
1 bảng có thể bao gồm nhiều cột
1 cột có thể có hoặc không có những thuộc tính.

b) Định nghĩa 1 số thuật ngữ:

NULL : Giá trị cho phép rỗng.
AUTO_INCREMENT : Cho phép giá trị tăng dần (tự động).
UNSIGNED : Phải là số nguyên dương
PRIMARY KEY : Cho phép nó là khóa chính trong bảng.

c)Loại dữ liệu trong Mysql:

Ở đây chúng tả chỉ giới thiệu 1 số loại thông dụng: 1 số dữ liệu khác có thể tham khảo trên trang chủ của mysql.

Kiểu dữ liệu	Mô Tả
Char	Định dạng text có chiều dài từ 0->255
Varchar	Định dạng text có chiều dài từ 0->255
Text	Định dạng text có chiều dài 0->65535
Longtext	Định dạng text có chiều dài 0->4294967215
INT	Định dạng số có chiều dài từ 0->4294967215
Float	Định dạng số thập phân có chiều dài nhỏ
Double	Định dạng số thập phân có chiều dài lớn
Date	Định dạng thời gian theo định dạng: YYYY-MM-DD
DateTime	Định dạng thời gian theo định dạng: YYYY-MM-DD HH:MM:SS

3- Những cú pháp cơ bản:

Cú pháp tạo 1 cơ sở dữ liệu:

CREATE DATABASE tên_cơ_sở_dữ_liệu;

Cú pháp sử dụng cơ sở dữ liệu: Use tên_database;

Cú pháp thoát khỏi cơ sở dữ liệu: Exit

Cú pháp tạo 1 bảng trong cơ sở dữ liệu:
CREATE TABLE user (<tên_cột> <mô_tả>,…,<tên_cột_n>…..<mô_tả_n>)

Ví dụ:
mysql> create table user(user_id INT(15) UNSIGNED NOT NULL AUTO_INCREMENT, username VARCHAR(255) NOT NULL, password CHAR(50) NOT NULL, email VARCHAR(200) NOT NULL, PRIMARY KEY (user_id));

Hiển thị có bao nhiều bảng: show tables;
Hiển thị có bao nhiêu cột trong bảng: show columns from table;
Thêm 1 cột vào bảng :

ALTER TABLE tên_bảng ADD <tên_cột> <thuộc_tính> AFTER <tên_cột>

Ví dụ:
mysql> alter table user add sex varchar(200) NOT NULL after email;

4- Thêm giá trị vào bảng:

Cú pháp:
INSERT INTO Tên_bảng(tên_cột) VALUES(Giá_trị_tương_ứng);

Ví dụ:
mysql> insert into user(username,password,email,sex,home) values("Lanna","12345","lanna@yahoo.com","F","www.abc.com");

5- Truy xuất dữ liệu:

Cú pháp:

SELECT tên_cột FROM Tên_bảng;

Ví dụ:
mysql> select user_id,username from user;

6- Truy xuất dữ liệu với điều kiện:
Cú pháp:

SELECT tên_cột FROM Tên_bảng WHERE điều kiện;
Ví dụ:

mysql> select user_id,username from user where user_id=2;

7- Truy cập dữ liệu và sắp xếp theo trình tự
Cú pháp:

SELECT tên_cột FROM Tên_bảng
WHERE điều kiện (có thể có where hoặc không)
ORDER BY Theo quy ước sắp xếp.

Trong đó quy ước sắp xếp bao gồm hai thông số là ASC (từ trên xuống dưới), DESC (từ dưới lên trên).

mysql> select user_id,username from user order by username ASC ;

8- Truy cập dữ liệu có giới hạn :
Cú pháp:

SELECT tên_cột FROM Tên_bảng
WHERE điều kiện (có thể có where hoặc không)
LIMIT vị trí bắt đầu, số record muốn lấy ra

Ví dụ:
mysql> select user_id,username from user order by username ASC limit 0,10 ;

9- Cập nhật dữ liệu trong bảng:
Cú pháp:

Update tên_bảng set tên_cột=Giá trị mới
WHERE (điều kiện).

Nếu không có ràng buộc điều kiện, chúng sẽ cập nhật toàn bộ giá trị mới của các record trong bảng.
Ví dụ:

mysql> update user set email="admin@qhonline.info" where user_id=1 ;

10- Xóa dữ liệu trong bảng:
Cú pháp:

DELETE FROM tên_bảng WHERE (điều kiện).

```
mysql>create database demo_mysql;
mysql> use demo_mysql;
mysql>create table user(id INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
username VARCHAR(50) NOT NULL, password CHAR(50) NOT NULL, PRIMARY
KEY(id));
```

Tạo trang test.php. Đầu tiên chúng ta sẽ kết nối cơ sở dữ liệu.

```
<?
$conn=mysql_connect("localhost", "root", "root") or die("can't connect database");
mysql_select_db("demo_mysql",$conn);
?>
```

Tiếp đến viết câu truy vấn lấy ra tất cả user từ database.

```
<?
$sql="select * from user";
$query=mysql_query($sql);
?>
```

Kiểm tra xem trong bảng dữ liệu đã tồn tại user nào chưa ?. Nếu chưa thì xuất ra thông báo lỗi, ngược lại thì đưa chúng vào mảng và lặp cho đến hết bảng dữ liệu.

```
<?
if(mysql_num_rows($query) == 0)
{
echo "Chua co du lieu";
}
else
{
while($row=mysql_fetch_array($query))
{
echo $row[username] ." - ".$row[password]."<br />";
}
}
?>
```

Và cuối cùng chúng ta đóng kết nối và kết thúc thao tác xử lý.

```
<?
mysql_close($conn);
?>
```

Và cuối cùng là file hoàn chỉnh của ứng dụng trên:

```
<?
$conn=mysql_connect("localhost", "root", "root") or die("can't connect database");
mysql_select_db("demo_mysql",$conn);
$sql="select * from user";
$query=mysql_query($sql);
if(mysql_num_rows($query) == 0)
{
echo "Chua co du lieu";
}
else
{
while($row=mysql_fetch_array($query))
{
```

print?

```
<?php
if(isset($_POST['ok']))
{
 if($_POST['username'] == NULL)
 {
  echo "Please enter your username<br />";
 }
 else
 {
  $u=$_POST['username'];
 }
 if($_POST['password'] == NULL)
 {
  echo "Please enter your password<br />";
 }
 else
 {
 $p=$_POST['password'];
 }
}
?>
```

Đoạn code ở trên sẽ kiểm tra xem người dùng có tiến hành nhấn nút đăng nhập hay không. Và nếu có thì chúng ta sẽ xét tiếp tình trạng người dùng có để trống các ô username và password hay không. Nếu có chúng ta sẽ thông báo lỗi ở bên trên form, để người sử dụng tiến hành nhập liệu.

Kế đến chúng ta kiểm tra xem có tồn tại hai biến $u và $p (chỉ khi người dùng đăng nhập thành công thì mới có thể tạo ra 2 biến đó). Tiếp đến chúng ta tiến hành kết nối cơ sở dữ liệu (xem lại bài **kết hợp PHP & MYSQL trong ứng dụng web**).

view source

print?

```
<?php
$conn=mysql_connect("localhost","root","root") or die("can't connect this database");
mysql_select_db("project",$conn);
?>
```

Và tiến hành kiểm tra xem username và password người sử dụng vừa nhập có trùng khớp với thông tin có trong cơ sở dữ liệu hay không ?. Nếu không thì chúng ta sẽ báo lỗi ngay. Ngược lại sẽ tiến hành lấy dữ liệu từ bảng và gán vào session. Để có thể quản lý phiên làm việc một cách hiệu quả trên mọi trang của khu vực admin (xem lại bài **khái niệm cơ bản về cookie và session**).

view source

print?

```
<?php
if($u && $p)
{
 $conn=mysql_connect("localhost","root","root") or die("can't connect this database");
 mysql_select_db("project",$conn);
```

```
06  $sql="select * from user where username='".$u."' and
password='".$p."'";
07  $query=mysql_query($sql);
08  if(mysql_num_rows($query) == 0)
09  {
10   echo "Username or password is not correct, please try again";
11  }
12  else
13  {
14   $row=mysql_fetch_array($query);
15   session_start();
16   session_register("userid");
17   session_register("level");
18   $_SESSION['userid'] = $row[id];
19   $_SESSION['level'] = $row[level];
20  }
21 }
22 ?>
```

Như vậy code hoàn chỉnh cho ứng dụng này là như sau:
view source
print?

```
01 <?php
02 if(isset($_POST['ok']))
03 {
04  if($_POST['username'] == NULL)
05  {
06   echo "Please enter your username<br />";
07  }
08  else
09  {
10   $u=$_POST['username'];
11  }
12  if($_POST['password'] == NULL)
13  {
14   echo "Please enter your password<br />";
15  }
16  else
17  {
18   $p=$_POST['password'];
19  }
20  if($u && $p)
21  {
22   $conn=mysql_connect("localhost","root","root") or die("can't connect
this database");
23   mysql_select_db("project",$conn);
24   $sql="select * from user where username='".$u."' and
password='".$p."'";
25   $query=mysql_query($sql);
26   if(mysql_num_rows($query) == 0)
```

```
10  exit();
11 }
12 ?>
```

Như vậy tại trang add_user.php chúng ta cũng đặt đoạn code trên và viết thêm 1 form thực thi công việc thêm thành viên.

view source
print?

```
01 <form action=add_user.php method=POST>
02 Level: <select name=level>
03 <option value=1>Member</option>
04 <option value=2>Admin </option>
05 </select><br />
06 Username: <input type=text name=username size=25><br />
07 Password: <input type=password name=password size=25> <br />
08 Re-Password: <input type=password name=re-password size=25><br />
09 <input type=submit name="adduser" value="Add New User">
10 </form>
```

Tương tự như những bài trước, chúng ta cũng lần lượt kiểm tra từng đối tượng. Trước hết cần xem thử người dùng đã nhấn nút Add New user chưa, sau đó lại kiểm tra xem người dùng đã nhập đầy đủ thông tin chưa. Cuối cùng là so sánh giữa password và re-password có giống nhau hay không. Tiếp theo chúng ta gán biến $l sẽ bằng với giá trị mà khi người dụng chọn level cho user đó. (xem lại **Bài 4: Xử lý giá trị form trong PHP**)

view source
print?

```
01 <?php
02 if(isset($_POST['adduser']))
03 {
04  if($_POST['username'] == "")
05  {
06   echo "Vui long nhap username<br />";
07  }
08  else
09  {
10   $u=$_POST['username'];
11  }
12  if($_POST['password'] != $_POST['re-password'])
13  {
14   echo "Password va re-password khong chinh xac<br />";
15  }
16  else
```

```
17  {
18   if($_POST['password'] == "" )
19   {
20    echo "Vui long nhap password<br />";
21   }
22   else
23   {
24    $p=$_POST['password'];
25   }
26  }
27  $l=$_POST['level'];
28 }
29 ?>
```

Kế đến chúng ta lại tiếp tục kiểm tra xem có tồn tại biến $u, $p và $l hay không (chỉ khi người dùng nhập liệu thì mới tồn tại cùng lúc 3 biến này). sau khi đã kiểm tra sự tồn tại của 3 biến này chúng ta tiếp tục **dùng PHP kết nối cơ sở dữ liệu** để kiểm tra xem username mà người dùng vừa chọn có trùng với username khác trong cơ sở dữ liệu chưa. nếu trùng thì phải xuất thông báo lỗi yêu cầu họ chọn username khác còn nếu không thì ta lại tiến hành thêm thành viên mới này vào CSDL.

view source
print?

```
01 <?php
02 if($u & $p & $l)
03 {
04  $conn=mysql_connect("localhost","root","root") or die("can't connect
   this database");
05  mysql_select_db("project",$conn);
06  $sql="select * from user where username='".$u."'";
07  $query=mysql_query($sql);
08  if(mysql_num_rows($query) != "" )
09  {
10   echo "Username nay da ton tai roi<br />";
11  }
12  else
13  {
14   $sql2="insert into user(username,password,level)
   values('".$u."','".$p."','".$l."')";
15   $query2=mysql_query($sql2);
16   echo "Da them thanh vien moi thanh cong";
17  }
18 }
19 ?>
```

Tới đây ta đã kết thúc thao tác thêm một thành viên mới vào cơ sở dữ liệu. Sau đây là toàn bộ code trong ứng dụng này.

view source
print?

```
01 <?php
02 session_start();
03 if(isset($_SESSION['userid']) && $_SESSION['level'] == 2)
04 {
```

```
05 if(isset($_POST['adduser']))
06 {
07  if($_POST['username'] == "")
08  {
09   echo "Vui long nhap username<br />";
10  }
11  else
12  {
13   $u=$_POST['username'];
14  }
15  if($_POST['password'] != $_POST['re-password'])
16  {
17   echo "Password va re-password khong chinh xac<br />";
18  }
19  else
20  {
21   if($_POST['password'] == "" )
22   {
23    echo "Vui long nhap password<br />";
24   }
25   else
26   {
27    $p=$_POST['password'];
28   }
29  }
30  $l=$_POST['level'];
31  if($u & $p & $l)
32  {
33   $conn=mysql_connect("localhost","root","root") or die("can't connect
this database");
34   mysql_select_db("project",$conn);
35   $sql="select * from user where username='".$u."'";
36   $query=mysql_query($sql);
37   if(mysql_num_rows($query) != "" )
38   {
39    echo "Username nay da ton tai roi<br />";
40   }
41   else
42   {
43    $sql2="insert into user(username,password,level)
values('".$u."','".$p."','".$l."')";
44    $query2=mysql_query($sql2);
45    echo "Da them thanh vien moi thanh cong";
46   }
47  }
48 }
49 }
50 ?>
51 <form action=add_user.php method=POST>
```

được đếm trên mỗi user khi lặp, username là tên truy cập của họ, level là cấp bậc của user (1 là member và 2 là admin), edit là cột chứa các link chỉnh sửa user, del là cột xóa các user.
view source
print?

```
1 <table align=center width=400 border=1>
2 <tr>
3 <td>STT</td>
4 <td>Username</td>
5 <td>Level</td>
6 <td>Edit</td>
7 <td>Del</td>
8 </tr>
```

Tiếp theo, chúng ta sẽ kết nối CSDL để tiến hành lặp các user.
view source
print?

```
01 <?php
02 $conn=mysql_connect("localhost","root","root") or die("can't connect this database");
03 mysql_select_db("project",$conn);
04 $sql="select * from user order by id DESC";
05 $query=mysql_query($sql);
06 if(mysql_num_rows($query) == "")
07 {
08 echo "<tr><td colspan=5 align=center>Chua co username nao</td></tr>";
09 }
10 ?>
```

Đoạn code ở trên tiến hành lựa chọn tất cả user có trong database. Đồng thời kiểm tra xem trong database có tồn tại user nào không. Nếu không sẽ xuất ra thông báo "chưa có username nào". Ngược lại khi có tồn tại user chúng ta sẽ xử lý tiếp như sau:
view source
print?

```
01 <?php
02 $stt=0;
03 while($row=mysql_fetch_array($query))
04 {
05 $stt++;
06 echo "<tr>";
07 echo "<td>$stt</td>";
08 echo "<td>$row[username]</td>";
09 if($row[level] == "1")
10 {
11 echo "<td>Member</td>";
12 }
13 else
14 {
15 echo "<td>Admin</td>";
16 }
17 echo "<td><a href=edit_user.php?userid=$row[id]>Edit</a></td>";
18 echo "<td><a href=del_user.php?userid=$row[id]>Del</a></td>";
19 echo "</tr>";
```

```
20 ?>
```

Ở đây ta dùng biến $stt để làm bộ đếm cho số lượng user. Như vậy, cứ mỗi lần lặp 1 user thì bộ đếm này sẽ tự động tăng lên 1 đơn vị. Sau đó, chúng ta đưa dữ liệu vào mảng và truyền các giá trị trên vào các cột dữ liệu trong bảng.

STT	Username	Level	Edit	Del
1	jacky	Admin	Edit	Del
2	test	Member	Edit	Del
3	abc	Member	Edit	Del
4	admin	Admin	Edit	Del

http://www.qhonline.info

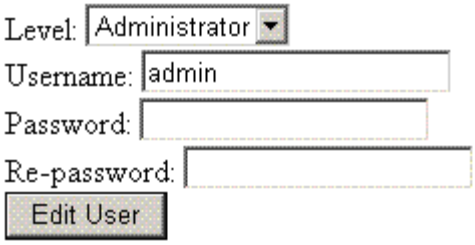
Tới đây ta đã kết thúc thao tác quản lý các user đang tồn tại trong cơ sở dữ liệu. Sau đây là toàn bộ code trong ứng dụng này.

view source

print?

```
01 <?php
02 sesison_start();
03 if(isset($_SESSION['userid']) && $_SESSION['level'] == 2)
04 {
05 ?>
06 <table align=center width=400 border=1>
07 <tr>
08 <td>STT</td>
09 <td>Username</td>
10 <td>Level</td>
11 <td>Edit</td>
12 <td>Del</td>
13 </tr>
14 <?
15 $conn=mysql_connect("localhost","root","root") or die("can't connect this database");
16 mysql_select_db("project",$conn);
17 $sql="select * from user order by id DESC";
18 $query=mysql_query($sql);
19 if(mysql_num_rows($query) == "")
20 {
21 echo "<tr><td colspan=5 align=center>Chua co username nao</td></tr>";
22 }
23 else
24 {
25 $stt=0;
26 while($row=mysql_fetch_array($query))
27 {
28 $stt++;
29 echo "<tr>";
```

Vì là trang chỉnh sửa thành viên, nên nội dung của chúng có phần sẽ giống với phần thêm thành viên, chỉ khác là các ô nhập liệu giờ đây đã có dữ liệu. Dữ liệu này chúng ta tiến hành lấy từ cơ sở dữ liệu thông qua biến truyền mà ở trang quản lý đã gửi edit_user.php?userid=$row[id]



http://www.qhonline.info

Như vậy để lấy được giá trị từ liên kết này chúng ta sẽ sử dụng biến $_GET['userid']. Sau khi đã có được giá trị này, việc còn lại của bạn là lấy thông tin của id này từ cơ sở dữ liệu và đưa vào form để người dùng có thể chỉnh sửa.

Vậy ta có.
view source
print?

```
01 <?php
02 $sql="select * from user where id='".$id."'";
03 $query=mysql_query($sql);
04 $row=mysql_fetch_array($query);
05 ?>
06 <form action="edit_user.php?userid=<?=$id?>" method=post>
07 Level: <select name=level>
08 <option value=1 <? if($row[level] == 1) echo "selected"; ?>)
>Member</option>
09 <option value=2 <? if($row[level] == 2) echo "selected";
?>>Administrator</option>
10 </select><br />
11 Username: <input type=text name=user size=20 value="<?=$row[username]?>"
/><br />
12 Password: <input type=password name=pass size=20 /> <br />
13 Re-password: <input type=password name=repass size=20 /><br />
14 <input type=submit name=ok value="Edit User" />
15 </form>
```

Theo như đoạn code ở trên phần value chúng ta muốn đưa giá trị vào thì buộc phải sử dụng cú pháp gọn <?=$ten?> hoặc cũng có thể sử dụng bằng cách <? echo $ten; ?>.. Ở đây chúng ta chỉ đưa ra giá trị trên text box username, còn lại mật khẩu do có thể bị mã hóa nên không nên hiển thị ra. Phần level do lưu trong cơ sở dữ liệu là những con số 1,2 nên khi đưa ra chúng ta phải so sánh, nếu trong bảng lưu là 1 thì ở chỗ có giá trị bằng 1 sẽ thêm chữ selected. Để mặc định lựa chọn khi người dùng nhấn vào nút chỉnh sửa.

Tiếp tới người dùng sẽ nhấn nút edit. Việc còn lại của chúng ta là kiểm tra dữ liệu khi họ tiến hành chỉnh sửa dữ liệu. Điểm khác biệt ở đây là chúng ta phải cân nhắc khi viết trang chỉnh sửa. Bởi không phải ai vào chỉnh sửa cũng đều sửa cả thông tin username, password, level. Có khi họ chỉ sửa mỗi level hoặc có khi họ chỉ sửa mỗi password.

Vì thế ta phải xét các trường hợp để giải quyết.

Trướng hợp 1: Người dùng chỉ chỉnh sửa username, level mà không chỉnh sửa password. Khi đó username, level bắt buộc đều đã có dữ liệu nên ta không cần kiểm tra chúng rỗng hay không. Nhưng với password chúng ta không thể bắt người lập trình nhập liệu. Và họ hoàn toàn có thể để rỗng.

Trường hợp 2: Người dùng chỉnh sửa username, level ,password. Khi đó ta xem password có dữ liệu và ta tiến hành cập nhật.

Để giải quyết bài toán trên thỏa 2 trường hợp ta sẽ xét phương pháp sau:

Ta kiểm tra password và repassword có giống nhau hay không ?. Nếu chúng khác nhau nghĩa là người dùng nhập liệu nhưng nhập sai. Vậy phải cảnh báo lỗi cho họ biết là họ đã nhập sai. Còn ngược lại nếu password và repassword giống nhau thì sẽ nảy sinh 2 trường hợp con.

+ Một là password và repassword đều có dữ liệu

+ Hai là password và repassword không có dữ liệu.

Khi đó ở trường hợp con 1 chúng ta sẽ khởi tạo biến $p và ghi nhận thông tin họ nhập liệu. Và ở trường hợp 2 chúng ta không ghi nhận thông tin người nhập liệu vì họ không có ý định chỉnh sửa mật khẩu. Có thể viết đoạn code đơn giản như sau:
view source
print?

```
01 <?php
02 if($_POST['pass'] != $_POST['repass'])
03 {
04  echo "Password and re-password is not correct";
05 }
06 else
07 {
08  if($_POST['pass'] != NULL)
09  {
10   $p=$_POST['pass'];
11  }
12 }
13 ?>
```

Tiếp tới chúng ta sẽ phân loại dữ liệu cập nhật. Khi có username,password, level thì ta cập nhật tất cả thông tin của họ bằng cú pháp SQL. (xem lại **bài 9: ngôn ngữ SQL và MYSQL cơ bản**). Ngược lại khi không tồn tại password thì ta chỉ cập nhật username và level.

Một điều nữa chúng ta cần quan tâm đó là đối với lệnh update và delete, ta phải truyền 1 id cụ thể để tránh việc chúng xóa hoặc sửa dữ liệu toàn bộ trong bảng user. Với id tôi truyền nhận từ bên ngoài ta sẽ có.
view source
print?

```
01 <?php
02 if($u && $p && $l )
03 {
04  $sql="update user set username='".$u."', password='".$p."',
```

```
   level='".$l."' where id='".$id."'";
05  mysql_query($sql);
06  header("location:mana_user.php");
07  exit();
08 }
09 else
10 {
11  if($u && $l)
12  {
13    $sql="update user set username='".$u."', level='".$l."' where
id='".$id."'";
14    mysql_query($sql);
15    header("location:mana_user.php");
16    exit();
17  }
18 }
19 ?>
```

Sau khi sửa thành công một thành viên chúng ta sẽ đưa người dùng trở về với trang quản lý (xem bài 13 để biết cách **xây dựng trang quản lý thành viên** như thế nào)

Nội dung của toàn bộ code ở trên như sau:
view source
print?

```
01 <?php
02 $conn=mysql_connect("localhost","root","root") or die("can't connect this
database");
03 mysql_select_db("project",$conn);
04 $id=$_GET['userid'];
05 if(isset($_POST['ok']))
06 {
07  if($_POST['user'] == NULL)
08  {
09   echo "Please enter your username";
10  }
11  else
12  {
13   $u=$_POST['user'];
14  }
15  if($_POST['pass'] != $_POST['repass'])
16  {
17   echo "Password and re-password is not correct";
18  }
19  else
20  {
21   if($_POST['pass'] != NULL)
22   {
23    $p=$_POST['pass'];
24   }
25  }
```

```
26 $l = $_POST['level'];
27 if($u && $p && $l )
28 {
29  $sql="update user set username='".$u."', password='".$p."',
level='".$l."' where id='".$id."'";
30  mysql_query($sql);
31  header("location:mana_user.php");
32  exit();
33 }
34 else
35 {
36  if($u && $l)
37  {
38   $sql="update user set username='".$u."', level='".$l."' where
id='".$id."'";
39   mysql_query($sql);
40   header("location:mana_user.php");
41   exit();
42  }
43 }
44 }
45 $sql="select * from user where id='".$id."'";
46 $query=mysql_query($sql);
47 $row=mysql_fetch_array($query);
48 ?>
49 <form action="edit_user.php?userid=<?=$id?>" method=post>
50 Level: <select name=level>
51 <option value=1 <? if($row[level] == 1) echo "selected"; ?>)
>Member</option>
52 <option value=2 <? if($row[level] == 2) echo "selected";
?>>Administrator</option>
53 </select><br />
54 Username: <input type=text name=user size=20 value="<?=$row[username]?>"
/><br />
55 Password: <input type=password name=pass size=20 /> <br />
56 Re-password: <input type=password name=repass size=20 /><br />
57 <input type=submit name=ok value="Edit User" />
58 </form>
```

B- Xây dựng trang xóa thành viên:

Đối với trang xóa dữ liệu, chúng ta cũng không cần phải xử lý quá nhiều. Bởi nhiệm vụ của chúng chỉ đơn giản là xóa đi những dòng trong bảng.

Như vậy cũng như trang edit chúng ta nhận giá trị từ nội dung liên kết ở trang quản lý đã gửi là del_user.php?userid=$row[id].

Và thực thi lệnh kết nối csdl để xử lý nội dung này.

view source
print?

```
1 <?php
```

```
20 </table>
21 </form>
```

Chúng ta chú ý phần hình ảnh, tại đây tôi truyền đường dẫn chính là liên kết tới trang random_image.php. Trang này sẽ thực thi công việc tạo ra những bức hình có dãy số ngẫu nhiên để phần nhập liệu của người sử dụng tham chiếu tới.

Tiếp theo, chúng ta khởi tạo file random_image.php để lấy ra dãy số ngẫu nhiên và phát sinh chúng ngay trên tấm hình cho người truy cập nhập liệu.

Để làm được điều đó. Chúng ta sẽ khởi tạo session và lưu dãy số nhập liệu đó vào session của mình, sau đó so khớp với trang form bên kia.(xem lại **bài 7: Khái niệm cơ bản về cookie và session trong PHP**)

Đầu tiên chúng ta sử dụng hàm md5 và ran để mã hóa các ký tự bao gồm số và chữ cái. Khi sử dụng hàm md5() ký tự phát sinh sẽ lên tới 32 ký tự. Và chúng ta chỉ lấy duy nhất 5 ký tự từ chuỗi mã hóa đó bằng hàm substr. Tiếp tục ta lưu đoạn mã hóa này trong session cụ thể $_SESSION['security_code'], để tại trang form ta sẽ sử dụng so sánh với phần nhập liệu của người sử dụng.

Vậy nên đoạn code dưới sẽ giải quyết các tình huống này.
view source
print?

```
1 <?php
2 $md5_hash = md5(rand(0,999));
3 $security_code = substr($md5_hash, 15, 5);
4 $_SESSION["security_code"] = $security_code;
5 ?>
```

Kế đến ta khởi tạo tấm hình với chiều rộng và chiều cao mà ta thiết lập thông qua hàm ImageCreate() . Và tiếp tục khai báo 2 màu chính là trắng và đen bằng hàm ImageColorAllocate($image, red, green, blue). Hàm này sẽ tạo ra một màu sắc từ hệ màu RGB trên tấm hình mà chúng ta vừa khởi tạo.

Tiếp tục ta đổ background của tấm hình sẽ là màu đen và chữ xuất hiện trên tấm hình sẽ là màu trắng bằng hàm ImageFill() và hàm ImageString(). Hàm ImageString có một số đối số cơ bản như sau: ImageString($image, 5, 30, 6, $security_code, $white); Trong đó:

+ $image là hình mà chúng ta khởi tạo.

+ 5: là font-size mà chúng ta quy ước cho ký tự xuất hiện trên hình.

+ 30: là khoảng cách bên trái của tấm hình.

+ 6 : là khoảng cách từ trên của tấm hình.

+ $security_code: là đoạn code sau khi chúng ta cắt ra ở trên.

+ $white: là màu sắc mà chúng ta đã sử dụng hàm ImageColorAllocate() ở trên để khởi tạo ra màu trắng.

view source
print?

```
1 <?php
2 $width = 100;
3 $height = 30;
4 $image = ImageCreate($width, $height);
5 $white = ImageColorAllocate($image, 255, 255, 255);
6 $black = ImageColorAllocate($image, 0, 0, 0);
7 ImageFill($image, 0, 0, $black);
8 ImageString($image, 5, 30, 6, $security_code, $white);
9 ?>
```

Kế tới chúng ta sẽ sử dụng lệnh header để trả nội dung này trở về dữ liệu dạng hình. Và tiến hành khởi tạo định dạng cho file hình mà chúng ta vừa tạo là JPG đồng thời giải phóng hình tạm được sử dụng để khợi tạo ra tấm hình này bằng đoạn code sau:

view source
print?

```
1 <?php
2 header("Content-Type: image/jpeg");
3 ImageJpeg($image);
4 ImageDestroy($image);
5 ?>
```

Để dễ quản lý chúng ta sẽ đặt tất cả đoạn code trên vào trong 1 hàm để dễ sử dụng và quản lý chúng tốt hơn. Vậy toàn bộ đoạn code trong trang random_image.php này như sau.

view source
print?

```
01 <?php
02 session_start();
03 function create_image()
04 {
05  $md5_hash = md5(rand(0,999));
06  $security_code = substr($md5_hash, 15, 5);
07  $_SESSION["security_code"] = $security_code;
08  $width = 100;
09  $height = 30;
10  $image = ImageCreate($width, $height);
11  $white = ImageColorAllocate($image, 255, 255, 255);
12  $black = ImageColorAllocate($image, 0, 0, 0);
13  ImageFill($image, 0, 0, $black);
14  ImageString($image, 5, 30, 6, $security_code, $white);
15  header("Content-Type: image/jpeg");
16  ImageJpeg($image);
17  ImageDestroy($image);
18 }
19 create_image() ;
20 exit();
21 ?>
```

Tiếp tục, trong trang form chúng ta phải khởi tạo session để chúng nhận được các giá trị mà ta đã khởi tạo và sử dụng ở trang random_image, kế tới ta kiểm tra xem người dùng có nhấn submit chưa. Nếu có ta sẽ tiếp tục kiểm tra xem người dùng nhập vào textbox có đúng là dãy số hiển thị trên hình ảnh hay không. Và xuất ra thông báo tương ứng với nội dung của người nhập liệu. (xem lại **bài 4: Xử lý giá trị form trong PHP**)

Sau đây là toàn bộ code của trang form.php.
view source
print?

```
01 <?php
02 session_start();
03 if(isset($_POST['ok']))
04 {
05  if($_POST['txtCaptcha'] == NULL)
06  {
07   echo "Please enter your code";
08  }
09  else
10  {
11   if($_POST['txtCaptcha'] == $_SESSION['security_code'])
12   {
13    echo "ma lenh hop le";
14   }
15   else
16   {
17    echo "Ma lenh khong hop le";
18   }
19  }
20 }
21 ?>
22 <form action="form.php" method=post>
23 <table>
24  <tr>
25   <td align="left">
26    <label for="captcha">Captcha</label>
27   </td>
28   <td>
29    <input type="text" name="txtCaptcha" maxlength="10" size="32" />
30   </td>
31   <td>
32    <img src="random_image.php" />
33   </td>
34  </tr>
35  <tr>
36   <td> </td>
37   <td>
38    <input type=submit name=ok value="Check" />
39   </td>
40  </tr>
```

Ví dụ: tôi viếng thăm website đó là 7h.

như vậy hệ thống sẽ ghi thông tin lúc đó là 7h.
Nếu sau 1 thời gian tôi không làm gì, hoặc tôi không truy cập website đó nữa thì hệ thống sẽ không ghi nhận thông tin mới. Như thế nếu bây giờ 8h và trừ đi 15 phút tôi quy ước, rõ ràng là thời gian mới hiện tại là 7h45, Thời gian này vẫn lớn hơn thời điểm lưu thông tin 7h (7h45 > 7h). Do vậy, nếu chúng thỏa điều kiện đó thì chúng ta chỉ việc xóa đi các record trong cơ sở dữ liệu là xong.
Vậy ta có code kết nối CSDL như sau
:(xem lại **bài 10: kết hợp PHP và MYSQL trong ứng dụng**)
view source
print?

```
1 <?php
2 $conn=mysql_connect("localhost","root","root") or die("can't connect");
3 mysql_select_db("online",$conn);
4 ?>
```

Sau đó ta tiến hành ghi nhận thông tin người dùng vào CSDL.
view source
print?

```
1 <?php
2 $sql="insert into useronline(tgtmp,ip,local)
  values('$tg','$REMOTE_ADDR','$PHP_SELF')";
3 $query=mysql_query($sql);
4 ?>
```

$REMOTE_ADDR là biến môi trường dùng để lấy ra IP của người truy cập.

$PHP_SELF là biến môi trường dùng để lấy ra đường dẫn mà người dùng đang truy cập.

Tiếp đến ta tiến hành xóa record khi thời gian thực lớn hơn thời gian trong cơ sở dữ liệu.
view source
print?

```
1 <?php
2 $sql="delete from useronline where tgtmp < $tgnew";
3 $query=mysql_query($sql);
4 ?>
```

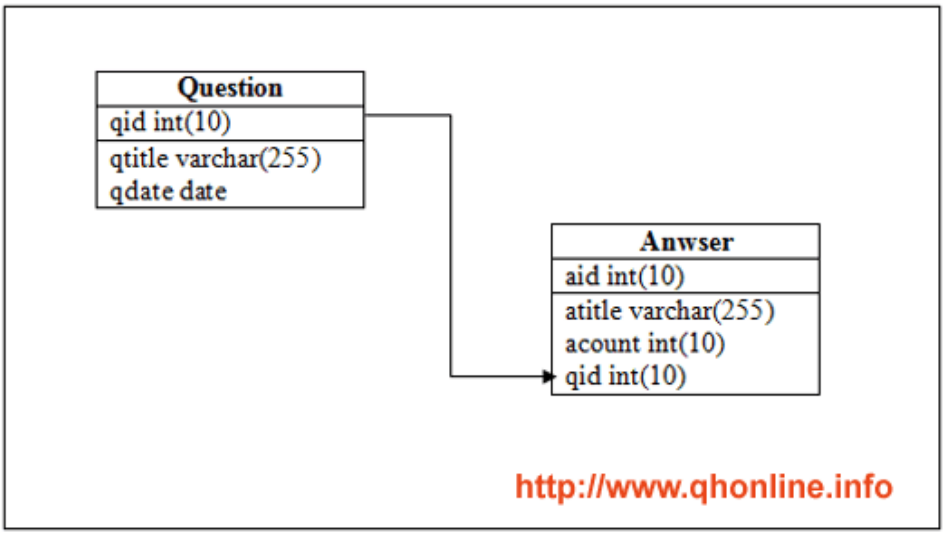
Tiếp tục là công việc hiển thị thông tin ra bên ngoài bằng cách liệt kê các record có trong database.
view source
print?

```
1 <?php
2 $sql="SELECT DISTINCT ip FROM useronline WHERE file='$PHP_SELF'";
3 $query=mysql_query($sql);
4 $user = mysql_num_rows($query);
5 ?>
```

DISTINCT là cú pháp cho phép liệt kê các dòng record mà không cho phép chúng có dữ liệu trùng lặp như cú pháp select bình thường.

Vậy chúng ta sẽ liệt kê tất cả những ip của những ai đang truy cập trên trang useronline.php.

Phần việc cuối cùng còn lại là chúng ta sẽ xuất thông tin ấy ra trình duyệt.



Vậy ta có cú pháp tạo 2 bảng như sau.

Bảng câu hỏi (question)
view source
print?

```
1 mysql> create table question(qid INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
2 -> qtitle VARCHAR(255) NOT NULL,
3 -> qdate DATE NOT NULL DEFAULT '0000-00-00',
4 -> PRIMARY KEY(qid));
```

Bảng câu trả lời (anwser)
view source
print?

```
1 mysql> create table answer(aid INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
2 -> qid INT(10) UNSIGNED NOT NULL,
3 -> atitle VARCHAR(255) NOT NULL,
4 -> acount INT(10) NOT NULL DEFAULT '0',
5 -> PRIMARY KEY(aid));
```

Nhập liệu bằng CSDL ta có:
view source
print?

```
1 mysql> Insert into question(qtitle,qdate) values("Ban thay qhonline the
  nao ?","
2 2009-10-10");
3 mysql> Insert into answer(qid,atitle,acount) values("1","Nhin rat dep",0);
4 mysql> Insert into answer(qid,atitle,acount) values("1","Nhin Dep",0);
5 mysql> Insert into answer(qid,atitle,acount) values("1","Nhin Cung
  duoc",0);
6 mysql> Insert into answer(qid,atitle,acount) values("1","Nhin qua xau",0);
```

B- Xây dựng trang bình chọn poll.php

Kết nối cơ sở dữ liệu:
view source
print?

```
1 <?php
2 $conn=mysql_connect("localhost","root","root") or die("can not connect
  database");
```

```
3 mysql_select_db("poll_exam",$conn);
4 ?>
```

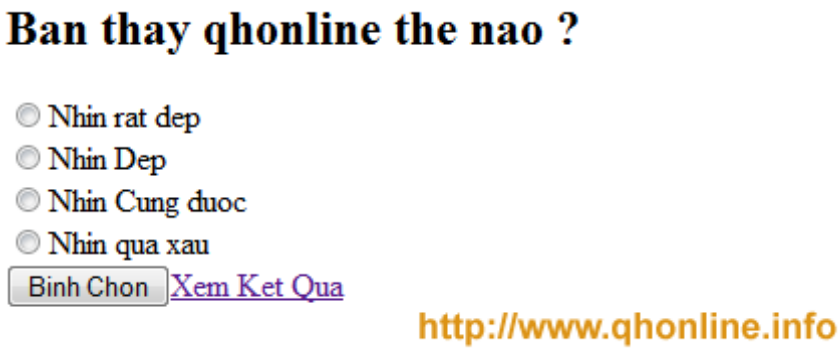
Lựa chọn câu hỏi có trong cơ sở dữ liệu để liệt kê ra trên website, sau đó ta lại tiếp tục liệt kê các câu trả lời của câu hỏi đó bằng cách sử dụng dấu chọn lựa (radio). Như vậy chúng ta sẽ phải chạy cùng lúc 2 câu truy vấn lồng nhau trong suốt quá trình truy xuất.

view source

print?

```
01 <?php
02 $sql="select * from question order by qid desc";
03 $query=mysql_query($sql);
04 if(mysql_num_rows($query) > 0)
05 {
06  $row=mysql_fetch_array($query);
07  $qid=$row[qid];
08  echo "<form action="poll.php?questionid=$qid" method="post" >";
09  echo "<h2>$row[qtitle]</h2>";
10  $sql2="select * from answer where qid='".$qid."' order by aid";
11  $query2=mysql_query($sql2);
12  if(mysql_num_rows($query2) > 0)
13  {
14   while($row2=mysql_fetch_array($query2)){
15   echo "<input type=radio name=answer value=$row2[aid]>$row2[atitle]<br />";
16   }
17  }
18  echo "<input type=submit name=ok value="Binh Chon">";
19  echo "<a href=result.php?questionid=$qid>Xem Ket Qua</a>";
20  echo "</form>";
21 }
22 ?>
```

Và màn hình sẽ xuất ra như hình bên dưới



Và khi người dùng nhất nút Bình chọn, chúng ta sẽ gọi lại chính trang đó để xử lý dữ liệu mà người dùng vừa lựa chọn.

Vậy ta phải sử dụng lệnh isset() để kiểm tra xem người dùng có nhấn nút bình chọn không, tiếp đến ta lấy ra id mà người dùng vừa tiến hành chọn ở form bên dưới.

Cuối cùng ta cập nhật dữ liệu bằng cách lấy số trong cơ sở dữ liệu cộng tiếp cho 1 đơn vị nữa. Và đưa người dùng sang trang kết quả.

Vì trong đoạn code này có lệnh header nên chúng ta phải đặt quá trình xử lý này ở trên form. Nếu không sẽ bị lỗi dữ liệu khi gởi.

view source
print?

```
01 <?php
02 if(isset($_POST['ok']))
03 {
04  $id=$_POST['answer'];
05  $qid=$_GET['questionid'];
06  $sql3="update answer set acount=acount + 1 where aid='".$id."'";
07  mysql_query($sql3);
08  header("location: result.php?questionid=$qid");
09  exit();
10 }
11 ?>
```

Vậy toàn bộ code của trang poll.php này như sau.

view source
print?

```
01 <?php
02 $conn=mysql_connect("localhost","root","root") or die("can not connect database");
03 mysql_select_db("poll_exam",$conn);
04 if(isset($_POST['ok']))
05 {
06  $id=$_POST['answer'];
07  $qid=$_GET['questionid'];
08  $sql3="update answer set acount=acount + 1 where aid='".$id."'";
09  mysql_query($sql3);
10  header("location: result.php?questionid=$qid");
11  exit();
12 }
13 $sql="select * from question order by qid desc";
14 $query=mysql_query($sql);
15 if(mysql_num_rows($query) > 0)
16 {
17  $row=mysql_fetch_array($query);
18  $qid=$row[qid];
19  echo "<form action="poll.php?questionid=$qid" method="post" >";
20  echo "<h2>$row[qtitle]</h2>";
21  $sql2="select * from answer where qid='".$qid."' order by aid";
22  $query2=mysql_query($sql2);
23  if(mysql_num_rows($query2) > 0)
24  {
25   while($row2=mysql_fetch_array($query2)){
26   echo "<input type=radio name=answer value=$row2[aid]>$row2[atitle]<br />";
27   }
28  }
```

```
29  echo "<input type=submit name=ok value="Binh Chon">";
30  echo "<a href=result.php?questionid=$qid>Xem Ket Qua</a>";
31  echo "</form>";
32 }
33 ?>
```

C- Xây dựng trang kết quả bình chọn result.php

Ở trang này, ta sẽ lấy giá trị question id của người bình chọn từ trang poll.php và tính toán xem tỷ lệ phần trăm của các câu hỏi ấy như thế nào.

Đầu tiên, ta lấy giá trị từ poll.php?questionid=1. vậy 1 là giá trị ta sẽ có được từ $_GET['questionid']. Tiếp đến ta lại dùng giá trị vừa lấy được này để liệt kê thông tin câu hỏi và tính toán giá trị.

Để tính toán giá trị, ta cần biết tổng số bình chọn trong toàn bộ các câu trả lời hiện nay là bao nhiêu. Bằng câu lệnh SUM ta có thể làm được điều đó.

view source
print?

```
1 $sql2="select qid, SUM(acount) as total from answer group by qid having
  qid='".$qid."'";
```

Câu lệnh này cho phép ta lấy ra tổng số bình chọn trong cơ sở dữ liệu của câu hỏi mà ta truyền vào. Để SUM được acount ta cần phải GROUP(gom nhóm) chúng ta lại theo mã câu hỏi mà chúng ta cần tìm. having là mệnh đề theo sau có ý nghĩa tương đương với where, nó thường được sử dụng theo GROUP.

Như vậy để lấy ra tổng số phiếu bình chọn ta chỉ việc thực thi câu truy vấn này, và lấy tên cột tạm là total.

view source
print?

```
1 <?php
2 $sql2="select qid, SUM(acount) as total from answer group by qid having
  qid='".$qid."'";
3 $query2=mysql_query($sql2);
4 $row2=mysql_fetch_array($query2);
5 $total=$row2[total];
6 ?>
```

kế tới, ta lại tính toán số phiếu của từng câu hỏi trong cơ sở dữ liệu, dựa vào tổng số câu hỏi này.

Ví dụ: Tổng số câu hỏi là 10
câu 1 là 3
câu 2 là 4
câu 3 là 2
câu 4 là 1
vậy suy ra phần trăm trên tổng số 10 của 4 câu này được tính bằng cách lấy. số liệu của từng câu chia cho tổng số câu hỏi và nhân cho 100. (3/10)*100 = 30%

Để làm tròn kết quả ta sử dụng hàm round(). Giúp dữ liệu đưa về sự đồng bộ và rõ ràng nhất.

Vậy code xử lý của chúng ta như sau:

view source
print?

```
01 <?php
02 $sql3="select * from answer where qid='".$qid."' order by aid";
03 $query3=mysql_query($sql3);
04 if(mysql_num_rows($query3) > 0)
05 {
06  while($row3=mysql_fetch_array($query3)){
07  $percent=round(($row3[acount]/$total)*100,2);
08  echo "<h4 style="color:red; font:12px verdana; ">$row3[atitle] : $row3[acount] ($percent %)</h4>";
09  }
10 }
11 ?>
```

Kết quả khi xuất ra sẽ có hình như sau:

Vậy toàn bộ mã nguồn của trang result.php này như sau:
view source
print?

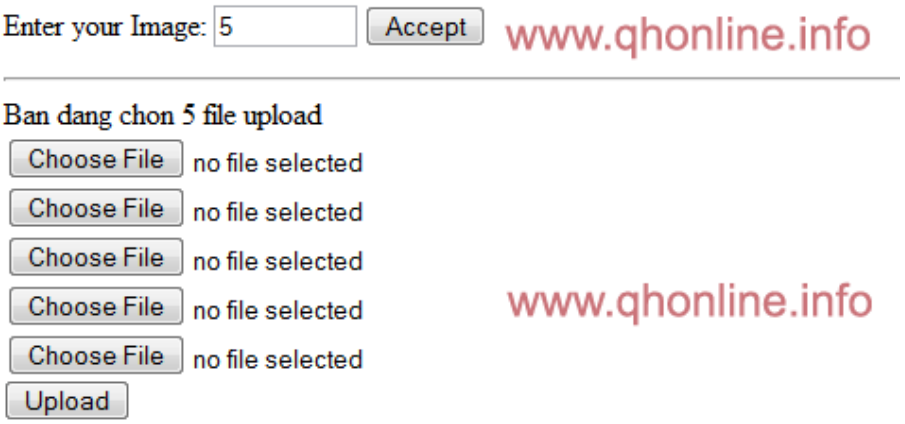
```
01 <?php
02 $conn=mysql_connect("localhost","root","root") or die("can not connect database");
03 mysql_select_db("poll_exam",$conn);
04 if(isset($_GET['questionid']))
05 {
06  $qid=$_GET['questionid'];
07  $sql="select * from question where qid='".$qid."'";
08  $query=mysql_query($sql);
09  $row=mysql_fetch_array($query);
10  echo "<h2>$row[qtitle]</h2>";
11  $sql2="select qid, SUM(acount) as total from answer group by qid having qid='".$qid."'";
12  $query2=mysql_query($sql2);
13  $row2=mysql_fetch_array($query2);
14  $total=$row2[total];
15  $sql3="select * from answer where qid='".$qid."' order by aid";
16  $query3=mysql_query($sql3);
17  if(mysql_num_rows($query3) > 0){
18   while($row3=mysql_fetch_array($query3)){
19   $percent=round(($row3[acount]/$total)*100,2);
20   echo "<h4 style="color:red; font:12px verdana; ">$row3[atitle] : $row3[acount] ($percent %)</h4>";
```

```
05              echo "<hr />";
06              echo "Ban dang chon $num file upload<br />";
07              echo "<form action='doupload.php?file=$num' method='post'
enctype='multipart/form-data'>";
08              for($i=1; $i <= $num; $i++)
09              {
10                   echo "<input type='file' name='img[]' /><br />";
11              }
12              echo "<input type='submit' name='ok_upload' value='Upload'
/>";
13              echo "</form>";
14 }
15 ?>
```

Đoạn code này sử dụng vòng lặp for với điều kiện là biến $i sẽ nhỏ hơn số file mà họ nhập vào. Đồng thời, để việc upload thành công tại form ta cũng cần thêm đoạn code :

enctype='multipart/form-data'.

Số lượng field sẽ lặp theo yêu cầu của người dùng. Do vậy. tại name của thể input, ta sẽ sử dụng mảng để lưu thông tin. Sử dụng name='img[]', có nghĩa là mỗi lần lặp và mỗi lần người dùng chọn file thì mảng img sẽ thêm 1 khóa tương ứng. (xem lại **bài 6: tổng quan về mảng về các hàm hỗ trợ**)
Và cuối cùng là nút submit thực thi công việc upload và gọi tới trang doupload.php?file=$num (chỉ rõ số file cụ thể do người dùng nhập tại trang doupload).



Vậy toàn bộ code của trang upload này sẽ như sau:
view source
print?

```
01 <form action="upload.php" method="post">
02 Enter your Image: <input type="text" name="txtnum" value="<?php echo
$_POST['txtnum']; ?>" size="10" />
03 <input type="submit" name="ok_num" value="Accept" />
04 </form>
05 <?php
06 if(isset($_POST['ok_num']))
07 {
08              $num=$_POST['txtnum'];
09              echo "<hr />";
10              echo "Ban dang chon $num file upload<br />";
11              echo "<form action='doupload.php?file=$num' method='post'
```

```
   enctype='multipart/form-data'>";
12              for($i=1; $i <= $num; $i++)
13              {
14                   echo "<input type='file' name='img[]' /><br />";
15              }
16              echo "<input type='submit' name='ok_upload' value='Upload'
    />";
17              echo "</form>";
18 }
19 ?>
```

Tại trang thực hiện nhiệm vụ doupload.php ta cần kiểm tra xem, người dùng có nhấn nút upload từ trang bên kia hay không ?. Nếu có ta sẽ xử lý công đoạn upload ấy, nếu không ta cần thông báo lỗi cho người dùng và yêu cầu họ chọn file.

view source
print?

```
01 <<?php
02 if(isset($_POST['ok_upload']))
03 {
04       //thực thi upload
05 }
06 else
07 {
08       echo "Vui long chon hinh truoc khi truy cap vao trang nay";
09 }
10 ?>
```

Như vậy, khi người dùng nhấn upload, ta sẽ lấy được tham số là biến $num từ liên kết $_GET[‘file'] mà ta đã thiết lập trên phần form action của file upload ở trang trước.

Biến $num này dùng để thực thi việc upload và tuần tự lấy các thông tin của file.

Để upload file từ máy lên hệ thống ta cần sử dụng hàm:

move_uploaded_file(tên_file_tạm,đường dẫn tới hệ thống).

Có thể diễn giải nôm na công việc của hàm này như sau: Ngay sau khi người dùng nhấn upload, hệ thống sẽ tạo ra 5 tham số ứng với từng file. Cụ thể là tên tạm (tmp_name), tên gốc (name), kích thước (size), định dạng mime (type) và lỗi (error) nếu có. Và khác với kiểu nhập liệu thông thường, khi chúng ta sử dụng <input type=file name=ten> thì lúc này sẽ phát sinh một biến môi trường mới là $_FILES[‘ten'] và dĩ nhiên nó sẽ đi kèm với 5 tham số cụ thể ở trên.

Ví dụ: $_FILES[‘ten'][‘name'] //Lấy ra tên gốc của file.

Vậy ứng với cú pháp upload ở trên ta sẽ có:

move_uploaded_file($_FILES['img']['tmp_name'],"data/".$_FILES['img']['name']);

Cú pháp này sẽ tự động upload hình ảnh lên thư mục data trên hệ thống. Vậy ta cần phải tạo thư mục data ngang cấp với 2 file upload và doupload. Nếu là host thật, bạn cần CHMOD thư mục là 777 (cho phép thư mục có toàn quyền) thì mới thực thi upload file được.

Trong tình huống của chúng ta, vì là upload một lúc nhiều file, nên lúc này ta phải sử dụng tham số mảng tại trang upload ta đã khởi tạo.

Vậy ứng với biến $num ta nhận từ upload ở trên ta sẽ triển khai code đơn giản như sau:
view source
print?

```
1 <?php
2 for($i=0; $i< $num; $i++)
3            {
4  move_uploaded_file($_FILES['img']['tmp_name'][$i],"data/".$_FILES['img']
['name'][$i]);
5  $url="data/".$_FILES['img']['name'][$i];
6  $name=$_FILES['img']['name'][$i];
7 }
8 ?>
```

Với biến $i là số lượng tăng lên theo vòng lặp. Vì mảng luôn bắt đầu bởi khóa là 0 nên ta cần phải thiết lập khởi tạo biến $i với giá trị là 0. Biến $url và $name là đường dẫn hình ảnh và tên gốc của tấm hình được sử dụng để lưu vào CSDL.

Kế tới, ta cần lưu thông tin hình ảnh tương ứng vào cơ sở dữ liệu. Để làm được điều này, ta cần kết nối CSDL với cú pháp sau:
(xem lại **bài 10: kết hợp php & mysql trong ứng dụng website**).
view source
print?

```
1 <?php
2 $conn=mysql_connect("localhost","root","root") or die("can't connect your
database");
3 mysql_select_db("images",$conn);
4 ?>
```

Như vậy mỗi khi upload thành công một hình ảnh, ta phải thêm hình ảnh đó vào CSDL như sau:
view source
print?

```
1 <?php
2 $sql="insert into images(img_url,img_name) values('$url','$name')";
3 mysql_query($sql);
4 ?>
```

Sau khi thêm vào CSDL ta cần hiển thị thông tin và hình ảnh vừa upload để cho người dùng biết họ đã upload thành công.

Khi đó ta sử dụng một biến $site để lưu đường dẫn gốc hiển thị trên textbox.

Demo Images Script - Copyright by QHOnline.Info

Upload Thanh cong file **akira.jpg**

www.qhonline.info

Images URL: http://www.qhonline.info/data/akira.jpg

Upload Thanh cong file **fullhouse7.jpg**

www.qhonline.info

Images URL: http://www.qhonline.info/data/fullhouse7.jp

Upload Thanh cong file **fullhouse8.jpg**

www.qhonline.info

Images URL: http://www.qhonline.info/data/fullhouse8.jp

Vậy hoàn chỉnh code cho file doupload.php này như sau:

view source

print?

```
01 <?php
02 $site="http://www.qhonline.info";
03 if(isset($_POST['ok_upload']))
04 {
05  $num=$_GET['file'];
06  echo "<h3>Demo Images Script - Copyright by QHOnline.Info</h3>";
07  $conn=mysql_connect("localhost","root","root") or die("can't connect
your database");
08  mysql_select_db("images",$conn);
09  for($i=0; $i< $num; $i++)
10  {
1    move_uploaded_file($_FILES['img']['tmp_name'][$i],"data/".$_FILES['img'
1 ]['name'][$i]);
12    $url="data/".$_FILES['img']['name'][$i];
13   $name=$_FILES['img']['name'][$i];
14   $sql="insert into images(img_url,img_name) values('$url','$name')";
15   mysql_query($sql);
16   echo "Upload Thanh cong file <b>$name</b><br />";
17   echo "<img src='$url' width='120' /><br />";
18   echo "Images URL: <input type='text' name='link' value='$site/$url'
size='35' /><br />";
19
20 }
```

```
06 margin:5px;
07 padding:5px;
08 width:400px;
09 }
10 a{
11 color:#666666;
12 text-decoration:none;
13 font-weight:900;
14 }
15 #cart{
16 border:1px solid #999999;
17 margin:5px;
18 padding:5px;
19 width:400px;
20 text-align:center;
21 }
```

Sau khi đã hoàn tất việc chuẩn bị cơ sở dữ liệu và giao diện, tiếp theo ta sẽ xây dựng trang hiển thị các món hàng trên website, cho phép người sử dụng lựa chọn sách khi truy cập.
Để liệt kê danh sách các quyển sách đang có trong database, ta cần kết nối CSDL với thao tác code như sau.
(xem lại **bài 10: kết hợp php & mysql trong ứng dụng website**).
view source
print?

```
1 <?php
2 $conn=mysql_connect("localhost","root","root") or die("can't connect your
database");
3 mysql_select_db("images",$conn);
4 ?>
```

Lúc này, ta có thể liệt kê các quyển sách bằng cú pháp sau:
view source
print?

```
1 <?php
2 $connect=mysql_connect("localhost","root","root")
3 or die("Can not connect database");
4 mysql_select_db("shop",$connect);
5 ?>
```

Lúc này, ta có thể liệt kê các quyển sách bằng cú pháp sau:
view source
print?

```
01 <?php
02 $sql="select * from books order by id desc";
03 $query=mysql_query($sql);
04 if(mysql_num_rows($query) > 0)
05 {
06  while($row=mysql_fetch_array($query))
07  {
08   echo "<div class=pro>";
09   echo "<h3>$row[title]</h3>";
10   echo "Tac Gia: $row[author] - Gia: ".number_format($row[price],3)."
VND<br />";
```

```
11    echo "<p align=right><a href=addcart.php?item=$row[id]>Mua Sach
   Nay</a></p>";
12    echo "</div>";
13   }
14 }
15 ?>
```

Đoạn code ở trên thực thi việc hiển thị sách nếu trong CSDL ít nhất 1 record. Và chúng sẽ liệt kê tiêu đề sách, tác giả, giá tiền. Ở đây, tôi sử dụng number_format() để lấy ra 3 số 000 cuối, ứng với đơn vị tiền tệ của Việt Nam là VNĐ.

Sau cùng tôi tạo một liên kết cho phép thêm quyển sách đó vào giỏ hàng nếu người mua cảm thấy ưng ý. (addcart.php sẽ là trang thêm vào giỏ hàng với tham số là mã của quyển sách).

Nói đến ứng dụng shopping cart thì hiện nay, trên thị trường tồn tại nhiều phương pháp code. Có thể sử dụng chuỗi để lưu giỏ hàng, cũng có thể lưu vào CSDL nháp giỏ hàng của người dùng và cũng có thể sử dụng mảng và session để lưu thông tin giỏ hàng. Trong khuôn khổ bài viết này, tôi sẽ sử dụng session và mảng để lưu thông tin giỏ hàng. (xem lại **Bài 7: Khái niệm cơ bản về Cookie và session trong PHP**)

Tại trang addcart.php tôi cần khởi tạo một session và lưu mã quyển sách vào một mảng. Cụ thể là: $_SESSION[‘cart'][$id] (xem lại **bài 6: tổng quan về mảng về các hàm hỗ trợ**). Với $id là mã quyển sách mà người dụng đã chọn ở trang xem hàng hóa (index.php). Mục đích chính của trang addcart này là lưu trữ hoặc tính toán lại số lượng sản phẩm khi mà họ lựa chọn. (lưu ý là số lượng các món hàng).

Một vấn đề đặt ra trong trang này, là làm thế nào để nhận biết món hàng người đó đã chọn hay chưa. Chẳng hạn. Lần đầu tôi chọn mua quyển A, sau đó tôi quay lại chọn mua tiếp quyển A. vậy trong giỏ hàng phải ghi nhận số lượng quyển A này là 2. Chứ không thể chỉ lưu là 1 được.

Vậy, lúc này ta sẽ kiểm tra xem. Quyển sách mà ta vừa chọn có tồn tại trong giỏ hàng hay chưa. Nếu có, ta phải tiến hàng lấy số lượng đang có tăng lên 1 đơn vị. Còn nếu không, ta phải gán số lượng của chúng là 1.

Code xử lý hoàn chỉnh trang addcart này sẽ như sau:
view source
print?

```
01 <?php
02 session_start();
03 session_register("cart");
04 $id=$_GET['item'];
05 if(isset($_SESSION['cart'][$id]))
06 {
07  $qty = $_SESSION['cart'][$id] + 1;
08 }
09 else
10 {
11  $qty=1;
12 }
13 $_SESSION['cart'][$id]=$qty;
14 header("location:cart.php");
15 exit();
16 ?>
```

Nôm na, chúng ta có thể hiểu addcart chỉ đơn giản là xử lý số lượng hàng hóa và lưu chúng ở dạng mảng mà thôi.

Như vậy tại trang mua sách, ta cũng cần cho khách hàng biết rằng trong giỏ hàng của họ hiện đang có bao nhiêu món hàng. Hoặc nếu chưa có món nào, ta cũng phải báo cho họ biết về việc đó.

Vậy khi nào thì giỏ hàng rỗng ?. Đó là khi session của giỏ hàng không tồn tại Id của quyển sách nào. Cụ thể, $_SESSION[‘cart'][id]. Khi id không tồn tại trong session này thì cũng là lúc giỏ hàng không tồn tại.

Vậy trước khi cho hiển thị giỏ hàng, ta cần kiểm tra xem có tồn tại id nào trong giỏ hàng hay không. Và vì id lưu ở dạng mảng đa chiều, nên ta cần dùng vòng lặp duyệt mảng foreach.

foreach($_SESSION[‘cart'] as $k=>$v)

Với $k có ý nghĩa tương đương $id quyển sách và $v tương đương là số lượng của quyển sách trong giỏ hàng. Vậy nếu tồn tại biến $k, thì tức có nghĩa là trong giỏ hàng có sách. Khi đó ta sử dụng một biến đã để báo hiệu rằng sách có tồn tại trong giỏ hàng hay không.
view source
print?

```
01 <?php
02  if(isset($_SESSION['cart']))
03  {
04   foreach($_SESSION['cart'] as $k=>$v)
05   {
06    if(isset($k))
07    {
08    $ok=2;
09    }
10   }
11  }
12  if ($ok != 2)
13   {
```

```
14   echo '<p>Ban khong co mon hang nao trong gio hang</p>';
15  } else {
16   $items = $_SESSION['cart'];
17   echo '<p>Ban dang co <a href="cart.php">'.count($items).' mon hang
trong gio hang</a></p>';
18  }
19 ?>
```

Trong đoạn code trên ta sử dụng hàm count nhắm đếm xem trong mảng hiện tại (giỏ hàng) hiện đang có bao nhiêu quyển sách. Và thực thi việc báo cho người dùng biết họ đang có bao nhiều quyển sách trong giỏ hàng.

Toàn bộ code xử lý của trang mua sách (index.php) này như sau:
view source
print?

```
01 <?php
02 session_start();
03 ?>
04 <html>
05 <head>
06  <title>Demo Shopping Cart - Created By My Kenny</title>
07  <link rel="stylesheet" href="style.css" />
08 </head>
09 <body>
10 <h1>Demo Shopping Cart</h1>
11 <div id=cart>
12 <?
13  if(isset($_SESSION['cart']))
14  {
15   foreach($_SESSION['cart'] as $k=>$v)
16   {
17    if(isset($v))
18    {
19    $ok=2;
20    }
21   }
22  }
23  if ($ok != 2)
24   {
25   echo '<p>Ban khong co mon hang nao trong gio hang</p>';
26  } else {
27   $items = $_SESSION['cart'];
28   echo '<p>Ban dang co <a href="cart.php">'.count($items).' mon hang
trong gio hang</a></p>';
29  }
30 ?>
31 </div>
32 <?
33 $connect=mysql_connect("localhost","root","root")
34 or die("Can not connect database");
```

```
5  $item[]=$key;
6 }
7 $str=implode(",",$item);
8 ?>
```

Giả sử lúc này chuỗi của chúng ta sẽ có dạng 7,8,9. Công việc tiếp theo là kết nối CSDL để liệt kê các sản phẩm có mã như ở trên. (xem lại **bài 10: kết hợp php & mysql trong ứng dụng website**). Thay vì sử dụng select * from tên_bảng where id= ??. Thì để tối ưu hơn, tôi sẽ sử dụng phép in trong SQL. Lúc này câu truy vẫn sẽ tương đương:

view source
print?

```
1 <?php
2 $sql="select * from books where id in ('$str') order by id desc";
3 $query=mysql_query($sql);
4 while($row=mysql_fetch_array($query))
5 ?>
```

Tiếp tục, ta lặp toàn bộ thông tin sách bao gồm tên, tác giả, giá tiền và cả số lượng mà ta đã lưu trong session là $_SESSION['cart']['ID_Món_Hàng']. ID_Món_hàng chính là thông tin ta lặp ra từ CSDL ($row[id]).

Bên cạnh đó tại phần số lượng, ta sẽ đưa giá trị lưu ở session ra textbox (vì tại giỏ hàng, người dùng được phép điều chỉnh số lượng, nên lúc này ta cần tạo textbox cho họ điều chỉnh). Vì lặp toàn bộ các quyển sách nên tại tên tên của textbox số lượng ta cũng cần truyền id để nhận biết số lượng đó thuộc ID của quyển sách nào.

view source
print?

```
1 <?php
2 echo "<p align=right>So Luong: <input type=text name=qty[$row[id]] size=5
  value={$_SESSION['cart'][$row[id]]}> - ";
3 ?>
```

Chúng ta cũng cho người dùng được phép xóa 1 món hàng nào đó ra khỏi giỏ hàng của họ. Bằng cách truyền mã quyển sách của từng quyển vào liên kết delcart.php.

view source
print?

```
1 <?php
2 echo "<a href=delcart.php?productid=$row[id]>Xoa Sach Nay</a></p>";
3 ?>
```

Tại đây, ta cũng cần tính luôn giá tiền của từng quyển sách tương ứng với số lượng mà họ đã chọn. Như vậy, số lượng là phần ta lưu ở session, còn giá tiền là phần ta lấy ra từ CSDL ứng với mảng $row ($row['price']).

view source
print?

```
1 <?php
2 echo "<p align=right> Gia tien cho mon hang: ".
  number_format($_SESSION['cart'][$row[id]]*$row[price],3) ." VND</p>";
3 ?>
```

Sau cùng, ta cần tính tổng tiền của toàn bộ sản phẩm có trong giỏ hàng. Bằng cách cộng dồn tổng giá tiền của từng món.

view source
print?

```
1 <?php
2 $total+=$_SESSION['cart'][$row[id]]*$row[price];
```

```
3 ?>
```

Phần còn lại, là chúng ta hiển thị giá tiền với đúng định dạng VND của Việt Nam.

view source

print?

```
1 <?php
2 echo "<b>Tong tien cho cac mon hang: <font color=red>".
  number_format($total,3)." VND</font></b>";
3 ?>
```

Đồng thời, ta cũng tạo nút cho phép người dùng cập nhật. Và cho phép người dúng xóa toàn bộ giỏ hàng.

view source

print?

```
1 <?php
2 echo "<input type="submit" name=submit value="Cap Nhat Gio Hang">";
3 echo "<div class=pro align=center>";
4 echo "<b><a href=index.php>Mua Sach Tiep</a> - <a
  href=delcart.php?productid=0>Xoa Bo Gio Hang</a></b>";
5 ?>
```

Như vậy, code đầy đủ sẽ là:

view source

print?

```
01 <?php
02 echo "<form action=cart.php method=post>";
03 foreach($_SESSION['cart'] as $key=>$value)
04 {
05  $item[]=$key;
06 }
07 $str=implode(",",$item);
08 $connect=mysql_connect("localhost","root","root") or die("Can not connect
   database");
09 mysql_select_db("shop",$connect);
10 $sql="select * from books where id in ($str)";
11 $query=mysql_query($sql);
12 while($row=mysql_fetch_array($query))
13 {
14  echo "<div class=pro>";
15  echo "<h3>$row[title]</h3>";
16  echo "Tac gia: $row[author] - Gia: ".number_format($row[price],3)."
   VND<br />";
17  echo "<p align=right>So Luong: <input type=text name=qty[$row[id]]
   size=5 value={$_SESSION['cart'][$row[id]]}> - ";
18  echo "<a href=delcart.php?productid=$row[id]>Xoa Sach Nay</a></p>";
19  echo "<p align=right> Gia tien cho mon hang: ".
   number_format($_SESSION['cart'][$row[id]]*$row[price],3) ." VND</p>";
20  echo "</div>";
21  $total+=$_SESSION['cart'][$row[id]]*$row[price];
22 }
23 echo "<div class=pro align=right>";
24 echo "<b>Tong tien cho cac mon hang: <font color=red>".
   number_format($total,3)." VND</font></b>";
25 echo "</div>";
```

```
26 echo "<input type="submit" name=submit value="Cap Nhat Gio Hang">";
27 echo "<div class=pro align=center>";
28 echo "<b><a href=index.php>Mua Sach Tiep</a> - <a
   href=delcart.php?productid=0>Xoa Bo Gio Hang</a></b>";
29 echo "</div>";
30 ?>
```

Sau khi thiết lập thành công trang giỏ hàng cơ bản, lúc này ta đã có thể thêm sách một cách dễ dàng. Tuy nhiên, giả sử trong trường hợp không có sách thì sao ?. Chúng ta vẫn chưa xét đến trường hợp giỏ hàng rỗng thì sẽ như thế nào. Vậy khi nào thì giỏ hàng rỗng ?. Đó là khi session của giỏ hàng không tồn tại Id của quyển sách nào. Cụ thể, $_SESSION[‘cart'][id]. Khi id không tồn tại trong session này thì cũng là lúc giỏ hàng không tồn tại.

Vậy trước khi cho hiển thị giỏ hàng, ta cần kiểm tra xem có tồn tại id nào trong giỏ hàng hay không. Và vì id lưu ở dạng mảng đa chiều, nên ta cần dùng vòng lặp duyệt mảng foreach.

foreach($_SESSION[‘cart’] as $k=>$v)

Với $k có ý nghĩa tương đương $id quyển sách và $v tương đương là số lượng của quyển sách trong giỏ hàng. Vậy nếu tồn tại biến $k, thì tức có nghĩa là trong giỏ hàng có sách.

view source
print?

```
01 <?php
02 $ok=1;
03 if(isset($_SESSION['cart']))
04 {
05  foreach($_SESSION['cart'] as $k => $v)
06  {
07   if(isset($k))
08   {
09    $ok=2;
10   }
11  }
```

```
12 }
13 if($ok == 2)
14 {
15 // code xử lý giỏ hàng ở trên.
16 }
17 ?>
```

Đoạn code này, ta sử dụng biến $ok để làm biến kiểm tra, mặc định khi load dữ liệu biến $ok sẽ bằng 1. Và khi trong giỏ hàng tồn tại sách thì chúng ta sẽ thay đổi biến $ok thành 2. Và gọi giỏ hàng như code ở trên.

Ở phần trên, ta cũng có đề cập khi người dùng tiến hành chỉnh sửa số lượng từng món hàng đơn lẻ qua textbox và nhấn cập nhận thì hệ thống sẽ tiến hành chỉnh sửa lại thông tin giỏ hàng. Vậy chúng ta sẽ xử lý như thế nào cho trương hợp đó.

Nếu chú ý, các bạn sẽ thấy dòng code chứa textbox cho phép người dùng nhập số lượng có một tham số đặc biêt là name=qty[$row[id]]. Vậy tham số này được dùng để làm gì ?.

Tham số này, nói cho chúng ta biết số lượng đang hiển thị là thuộc mã sản phẩm nào. Vậy khi tiến hành cập nhật giỏ hàng ta sẽ kiểm tra, nếu $qty[$row[id]] mà có giá trị là 0. Tức là người đó muốn xóa bỏ giỏ hàng. Ngược lại, ta chỉ viêc cập nhật giỏ hàng $id tương ứng với số lượng nhập ở textbox.

view source
print?

```
01 <?php
02 if(isset($_POST['submit']))
03 {
04  foreach($_POST['qty'] as $key=>$value)
05  {
06   if( ($value == 0) and (is_numeric($value)))
07   {
08    unset ($_SESSION['cart'][$key]);
09   }
10   elseif(($value > 0) and (is_numeric($value)))
11   {
12    $_SESSION['cart'][$key]=$value;
13   }
14  }
15  header("location:cart.php");
16 }
17 ?>
```

Việc xóa món hàng ở trên chỉ đơn giản là hủy bỏ session của id đó. Việc cập nhật số lượng chỉ đơn giản là gán đè số lượng người nhập (thẻ value trong textbox) vào số lượng đang lưu trong session.

Vậy code hoàn chỉnh của trang cart.php này sẽ như sau:

view source
print?

```
01 <?php
02 if(isset($_POST['submit']))
03 {
```

```
04  foreach($_POST['qty'] as $key=>$value)
05  {
06   if( ($value == 0) and (is_numeric($value)))
07   {
08    unset ($_SESSION['cart'][$key]);
09   }
10   elseif(($value > 0) and (is_numeric($value)))
11   {
12    $_SESSION['cart'][$key]=$value;
13   }
14  }
15  header("location:cart.php");
16 }
17 ?>
18 <html>
19 <head>
20  <title>Demo Shopping Cart - Created By My Kenny</title>
21  <link rel="stylesheet" href="style.css" />
22 </head>
23 <body>
24 <h1>Demo Shopping Cart</h1>
25 <?
26 $ok=1;
27 if(isset($_SESSION['cart']))
28 {
29  foreach($_SESSION['cart'] as $k => $v)
30  {
31   if(isset($k))
32   {
33    $ok=2;
34   }
35  }
36 }
37 if($ok == 2)
38 {
39
40    echo "<form action=cart.php method=post>";
41    foreach($_SESSION['cart'] as $key=>$value)
42    {
43     $item[]=$key;
44    }
45    $str=implode(",",$item);
46    $connect=mysql_connect("localhost","root","root") or die("Can not
connect database");
47    mysql_select_db("shop",$connect);
48    session_start();
49    $sql="select * from books where id in ($str)";
50    $query=mysql_query($sql);
```

```
51    while($row=mysql_fetch_array($query))
52    {
53    echo "<div class=pro>";
54    echo "<h3>$row[title]</h3>";
55    echo "Tac gia: $row[author] - Gia: ".number_format($row[price],3)." VND<br />";
56    echo "<p align=right>So Luong: <input type=text name=qty[$row[id]] size=5 value={$_SESSION['cart'][$row[id]]}> - ";
57    echo "<a href=delcart.php?productid=$row[id]>Xoa Sach Nay</a></p>";
58    echo "<p align=right> Gia tien cho mon hang: ". number_format($_SESSION['cart'][$row[id]]*$row[price],3) ." VND</p>";
59    echo "</div>";
60    $total+=$_SESSION['cart'][$row[id]]*$row[price];
61    }
62   echo "<div class=pro align=right>";
63   echo "<b>Tong tien cho cac mon hang: <font color=red>". number_format($total,3)." VND</font></b>";
64   echo "</div>";
65   echo "<input type="submit" name=submit value="Cap Nhat Gio Hang">";
66   echo "<div class=pro align=center>";
67   echo "<b><a href=index.php>Mua Sach Tiep</a> - <a href=delcart.php?productid=0>Xoa Bo Gio Hang</a></b>";
68   echo "</div>";
69  }
70 else
71  {
72   echo "<div class=pro>";
73   echo '<p align=center>Ban khong co mon hang nao trong gio hang<br /><a href=index.php>Buy Ebook</a></p>';
74   echo "</div>";
75  }
76 ?>
77 </body>
78 </html>
```

Và cuối cùng, khi người dùng nhấn xóa toàn bộ giỏ hàng hay chỉ xóa một món hàng, ta sẽ gọi tới trang delcart.php. Vậy trang này sẽ xử lý như thế nào ?.

Tại đây, ta có thể nhận tham số là: $_GET[‘productid'];

Tham số này sẽ là $id mà chúng truyền qua liên kết. Nếu xóa toàn bộ giỏ hàng, tức ta sẽ truyền cho nó giá trị bằng 0. Lúc này, ta sẽ hủy toàn bộ $_SESSION[‘cart']. Ngược lại, nếu là một $id cụ thể, thì ta chỉ xóa món hàng đó mà thôi. $_SESSION[‘cart'][$id].

Code hoàn chỉnh của file delcart.php như sau:
view source
print?

```
01 <?php
02 session_start();
03 $cart=$_SESSION['cart'];
04 $id=$_GET['productid'];
05 if($id == 0)
```

```
06 {
07  unset($_SESSION['cart']);
08 }
09 else
10 {
11 unset($_SESSION['cart'][$id]);
12 }
13 header("location:cart.php");
14 exit();
15 ?>
```

Download toàn bộ mã nguồn của bài này **tại đây**.

Như vậy, chúng ta đã hoàn tất việc xây dựng hoàn chỉnh một hệ thống **shopping cart** đơn giản. Tuy rằng, đây không phải là một bài viết hoàn chỉnh trong việc xây dựng mô hình **thương mại điện tử**. Nhưng quá đó, phần nào giúp các bạn hiểu và dễ dàng phát triển hệ thống của mình có tổ chức hơn.

Hy vọng trong thời gian tới, tôi sẽ viết tiếp bài viết sử dụng mô hình **thanh toán trực tuyến (paypal)** kết hợp với giỏ hàng để hoàn chỉnh bài hướng dẫn về **shopping cart** này. Mọi sự góp ý, cũng như thắc mắc của các bạn dành cho bài viết vui lòng sử dụng tính năng **thảo luận tại diễn đàn**. Hoặc thông qua việc trả lời với khung **phản hồi bên dưới**. Mong sẽ nhận được nhiều ý kiến đóng góp từ các bạn.