



# LẬP TRÌNH WEB PHP

**Giới thiệu về môn học**

# Thông tin giảng viên

- **LƯƠNG TRẦN HY HIỂN**
- **Bộ Môn Công nghệ Phần mềm**
- **Khoa Công nghệ Thông tin**
- **Phone: 0989 366 990**
- **Email: [hienlth@hcmup.edu.vn](mailto:hienlth@hcmup.edu.vn)**

# Thông tin môn học

- **LẬP TRÌNH WEB PHP**
- **Số đvht: 5 (60t = 45LT + 30TH)**
- **Môi trường làm việc:**
  - **XAMPP** [www.apachefriends.org](http://www.apachefriends.org)
    - Có đầy đủ Apache, PHP, MySQL, phpMyAdmin
    - Dễ dàng chạy Webserver trên USB
  - **AMP Start** [www.ampstart.com](http://www.ampstart.com)
    - Chương trình khởi động XAMPP



# Mục tiêu

- Nhằm cung cấp cho sinh viên các kiến thức liên quan đến các kiến thức cơ bản về web (HTML/DHTML + JavaScript + CSS).
- Cung cấp các kiến thức về ngôn ngữ lập trình ứng dụng Web bằng PHP/MySQL.
- Nắm được các phương pháp tổ chức, xây dựng một website cụ thể.

# Đề cương môn học: LẬP TRÌNH WEB

- **Chương 1:** Giới thiệu tổng quan HDJ
- **Chương 2:** PHP căn bản
- **Chương 3:** PHP nâng cao (file/ảnh)
- **Chương 4:** Kết nối PHP với MySQL
- **Chương 5:** Xây dựng ứng dụng thực tế

# Đánh giá kết quả học tập

- Thi lý thuyết (trắc nghiệm): **30%**
- Đồ án (bắt buộc): **70%**
- Đồ án không nộp đúng hạn, bài làm giống nhau **0** điểm.

# Tài liệu tham khảo

- Slide bài giảng của giảng viên
- PHP Manual
- Các tài liệu trên Internet



# Yêu cầu đồ án môn học

- File WORD:
  - Phân tích quy trình nghiệp vụ
    - Chức năng gì?
    - Ai làm?
    - Cần tổ chức như thế nào?
    - ...
  - Sơ đồ cơ sở dữ liệu
  - Phải đưa cho giáo viên duyệt trước
- Source Code:
  - Code + Database
- Nộp qua email: [hyhien@gmail.com](mailto:hyhien@gmail.com). Hạn chót nhận email: **23g59'** ngày **29/12/2010**

# Phân nhóm

- **Tổ chức nhóm**

- Mỗi nhóm có 1 – 3 học viên
- Mỗi nhóm có 1 nhóm trưởng

- **Nhiệm vụ của nhóm trưởng**

- Đăng ký đề tài môn học theo gợi ý của GV
- Chịu trách nhiệm điều khiển mọi hoạt động của nhóm trong quá trình thực hiện đồ án môn học.
- Báo cáo danh sách thành viên cho lớp trưởng
- Nộp đồ án môn học cho lớp trưởng

- **Nhiệm vụ của lớp trưởng**

- Lập danh sách nhóm đăng ký (tập tin excel)
- Nhận các bài nộp của các nhóm trong lớp
- Tổng hợp và nộp các đồ án của các nhóm cho GV

# Danh mục đề án môn học gợi ý

## 1. Quản lý phòng khách sạn

Giới thiệu các phòng khách sạn, cho khách đặt phòng, quản trị theo dõi và xác nhận đặt phòng.

## 2. Quản lý thư viện

Cập nhật sách, mượn/trả sách

## 3. File manager

Quản lý files, folders trên server. Mỗi người sử dụng được quản lý một vùng trên server và có quyền hạn nhất định trên đó.

## 4. Album ảnh

Tạo các album ảnh theo các chủ đề khác nhau. Người sử dụng có thể xem ảnh trong album.

# Danh mục đề án môn học gợi ý

## 5. Lịch làm việc

Tạo trang web lưu trữ và thể hiện lịch làm việc.

## 6. Lịch phòng học/thiết bị

Có một số thiết bị/phòng học dùng chung. Tạo web site cho phép người sử dụng đăng ký các thiết bị đó.

## 7. Web site cho phép gửi e-card.

Cho phép gửi và xem e-card.

## 8. Danh bạ điện thoại

Tạo danh bạ điện thoại, cho phép mọi người xem, tìm kiếm và cập nhật. Mở rộng là danh bạ công ty, danh bạ web site,...



# Danh mục đề án môn học gợi ý

## 9. Đăng ký học tín chỉ

Có danh sách sinh viên, các lớp học, danh sách các môn học và quan hệ giữa các môn học. Cho phép sinh viên đăng ký các môn để học tín chỉ.

## 10. RSS

Web site tin tức cung cấp RSS và đọc RSS feeds của các web site khác.

## 11. Site bán file nhạc, video

Web site rao bán các bản nhạc, phim,... cho phép nghe thử. Và chỉ một số người được phép download.

## 12. Trắc nghiệm

Web site cho phép một số người cập nhật câu hỏi thi trắc nghiệm, những người khác có thể thi. Điểm có thể lưu trữ hoặc không.

# Danh mục đề án môn học gợi ý

## 13. Module tin tức

Hiển thị tin tức theo chuyên mục (giống như các báo điện tử hiện có).

## 14. Kho tài liệu upload/download

Web site chia sẻ ebook cho phép thành viên có thể download hoặc upload tài liệu.

## 15. Forum đơn giản

Web site có chức năng forum.

## 16. Website giới thiệu sản phẩm

Web site bán hoa/thiết bị điện tử/điện máy.

# Câu hỏi và thảo luận



Đại Học Sư Phạm Tp. Hồ Chí Minh



# LẬP TRÌNH WEB PHP

## Chương 02: PHP CƠ BẢN

- **Phần 1: Tổng quan về PHP**
- **Phần 2: PHP cơ bản**

## 1. Giới thiệu PHP

- a. Lịch sử phát triển
- b. PHP là gì
- c. Ưu điểm
- d. Một số website dùng PHP

## 2. Hoạt động của Web Server

- a. Cơ chế
- b. Ví dụ

## 3. Yêu cầu cài đặt

- a. Web server (Apache, IIS)
- b. Phần mềm PHP (trình thông dịch PHP)
- c. Hệ quản trị cơ sở dữ liệu (mysql, SQL Server, Access)

## 4. Giải pháp cài đặt trọn gói (WAMP, LAMP, XAMP)

# 1. Giới thiệu PHP – Lịch sử



- **PHP** : **Rasmus Lerdorf** in 1994 (được phát triển để phát sinh các form đăng nhập sử dụng giao thức HTTP của Unix)
- **PHP 2 (1995)** : Chuyển sang **ngôn ngữ script xử lý trên server**. Hỗ trợ CSDL, Upload File, khai báo biến, mảng, hàm đệ quy, câu điều kiện, biểu thức, ...
- **PHP 3 (1998)** : Hỗ trợ ODBC, **đa hệ điều hành**, giao thức email (SNMP, IMAP), bộ phân tích mã PHP (parser) của **Zeev Suraski** và **Andi Gutmans**
- **PHP 4 (2000)** : Trở thành một thành phần độc lập cho các webserver. Parse đổi tên thành **Zend Engine**. Bổ sung các tính năng bảo mật cho PHP
- **PHP 5 (2005)** : Bổ sung Zend Engine II hỗ trợ **lập trình HĐT, XML, SOAP** cho Web Services, SQLite
- Phiên bản mới nhất của PHP là version PHP 5.2.4 (*www.php.net*)

# 1. Giới thiệu PHP – PHP là gì?



- **PHP** được phát triển từ ngôn ngữ kịch bản (script) với mục đích xây dựng trang Web cá nhân (**Personal Home Page**). Sau đó đã được phát triển thành một ngôn ngữ hoàn chỉnh và được ưa chuộng trên toàn thế giới trong việc phát triển các ứng dụng Web.
- **PHP** là một ngôn ngữ thông dịch.
- Là ngôn ngữ server-side script, tương tự như ASP, JSP, ... thực thi ở phía Server.
- Tập tin **PHP** có phần mở rộng là **.php**
- Cú pháp tương tự ngôn ngữ **C & Perl**



# 1. Giới thiệu PHP – Ưu điểm



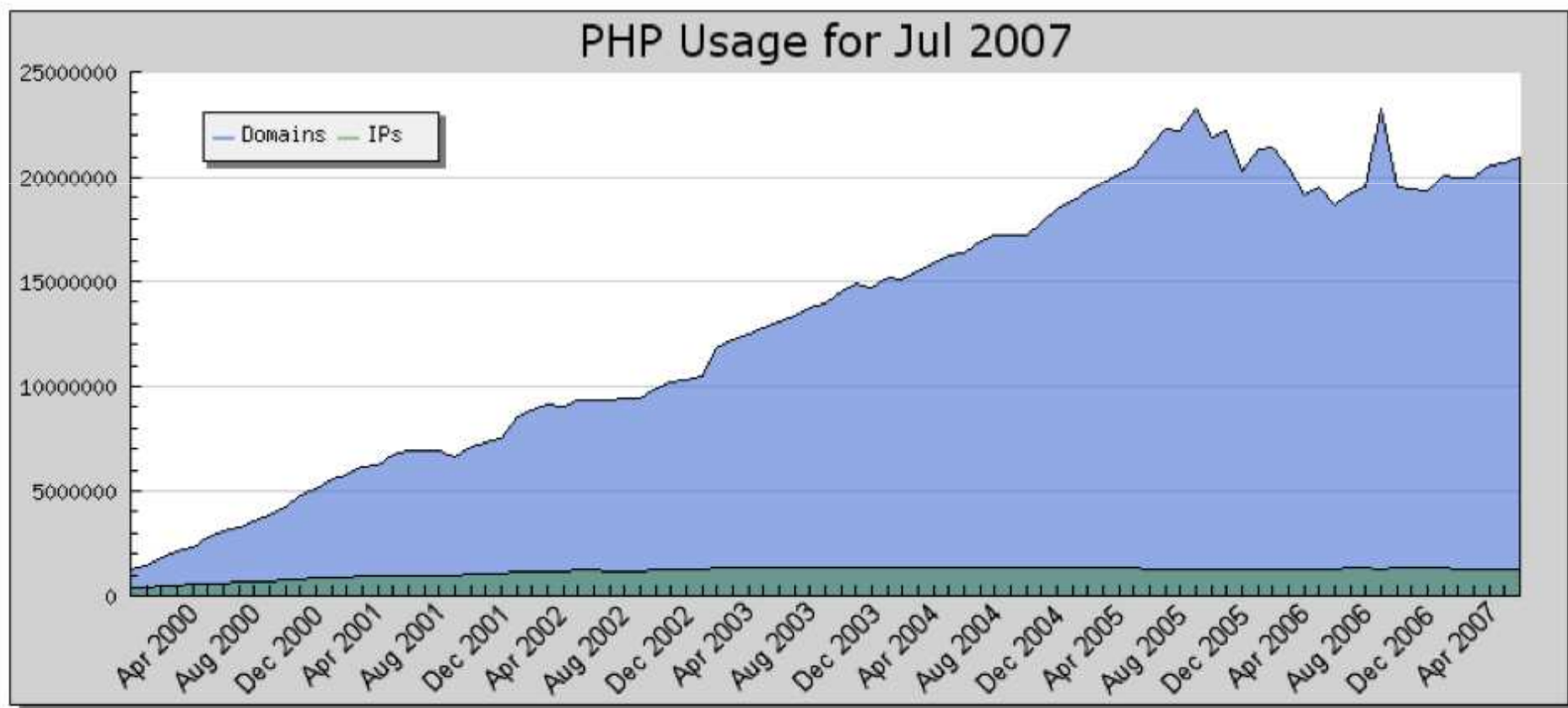
## (Multi - Platform)

- **Web Servers:** Apache, Microsoft IIS, Caudium, Netscape Enterprise Server
- **Hệ điều hành:** UNIX (HP-UX, OpenBSD, Solaris, Linux), Mac OSX, Windows NT/98/2000/XP/2003/vista
- **Hệ QTCSDL:** Adabas D, dBase, Empress, FilePro (read-only), Hyperwave, IBM DB2, Informix, Ingres, InterBase, FrontBase, mSQL, Direct MS-SQL, MySQL, ODBC, Oracle (OCI7 and OCI8), Ovrimos, PostgreSQL, SQLite, Solid, Sybase, Velocis, Unix dbm

# 1. Giới thiệu PHP – Ưu điểm



**PHP được sử dụng rộng rãi trên môi trường phát triển web**



# 1. Giới thiệu PHP – Một số website dùng PHP



PHP at Yahoo!

<http://www.yahoo.com>

The Internet's most trafficked site



Portal



Portal



Course Management System



Wiki



Customer Relationship Management



e-Commerce



Portal



Bulletin Board

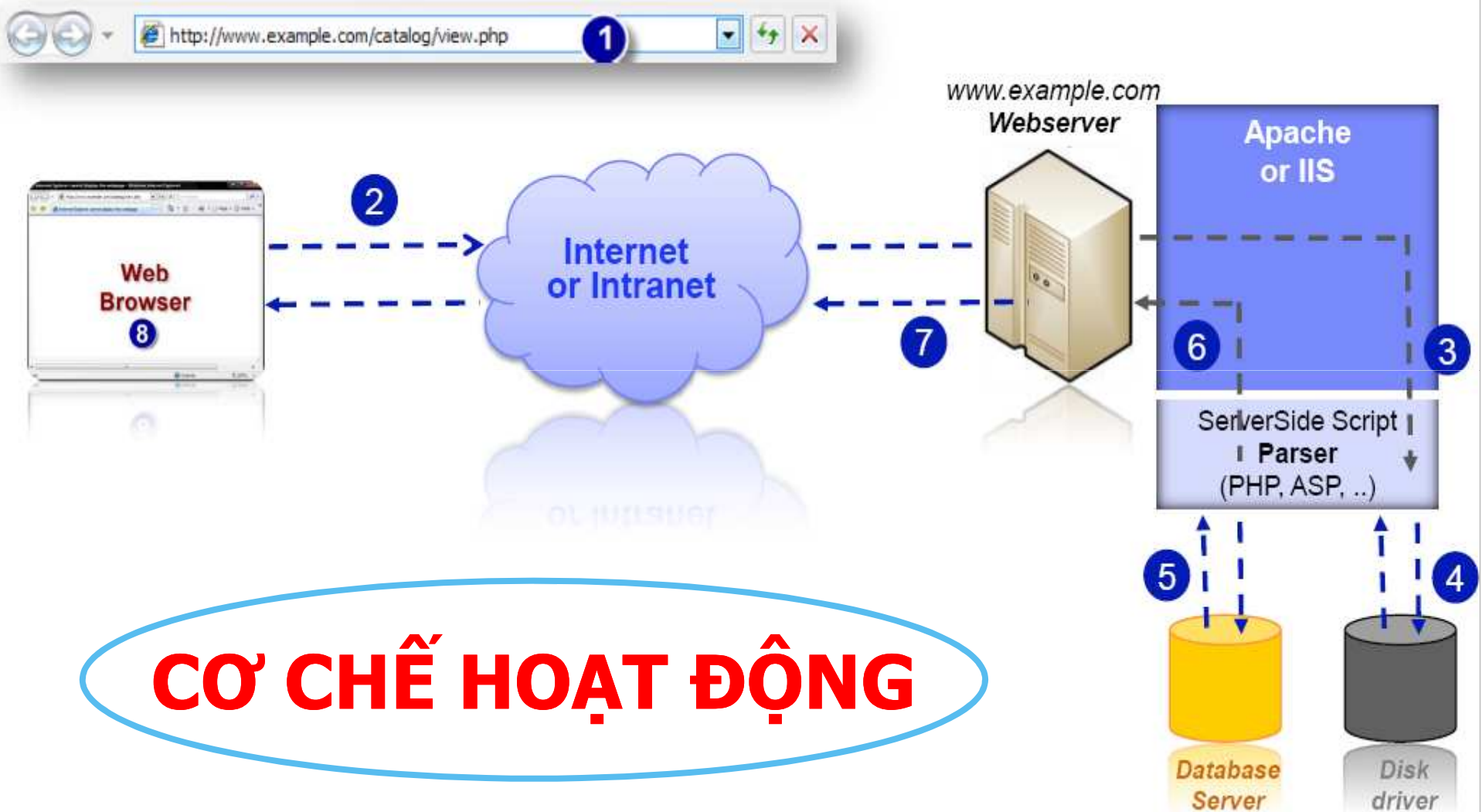


Content Management System



Help Desk

# 2. Hoạt động của Web Server



**CƠ CHẾ HOẠT ĐỘNG**

## 2. Hoạt động của Web Server



- **Apache** và **IIS** được xây dựng để đáp ứng các yêu cầu (request) các tập tin HTML
- Không hiểu cách thực thi mã **PHP** ở phía server
- **Apache** sử dụng các **modules** để “hiểu” cách thực thi các đoạn mã viết bằng PHP
- **IIS** sử dụng tập lệnh **ISAPI** - Internet Server Application Programming Interface - để “hiểu” các lệnh của PHP và ASP
- Ngoài ra, cả **Apache** và **IIS** đều hỗ trợ nhiều module khác phục vụ cho việc triển khai ứng dụng web một cách hiệu quả

## 2. Hoạt động của Web Server



```
<html>
<head>
  <title>PHP Test</title>
</head>
<body>
<?php echo '<p>Hello World</p>'; ?>
</body>
</html>
```

hello.php

```
<html>
<head>
  <title>PHP Test</title>
</head>
<body>
  <p>Hello World</p>
</body>
</html>
```

output

# 2. Hoạt động của Web Server



```
1 <html>
2 <head>
3     <title>Test Server Script Parser</title>
4 </head>
5 <body>
6
7     <h1>Server Script Parser</h1>
8     Hello world HTML
9     <br />
10    <br />
11
12    <% response.write("Hello ASP Parser !!!") %>
13    <br />
14    <br />
15
16    <?php echo "Hello PHP Parser !!!" ?>
17    <br />
18    <br />
19
20 </body>
21 </html>
22
```

HTML Code

ASP Code

PHP Code

# 3. Yêu cầu cài đặt



- **Download Apache Server**

Download Apache for free here:

<http://httpd.apache.org/download.cgi>

- **Download PHP**

Download PHP for free here:

<http://www.php.net/downloads.php>

- **Download MySQL Database**

Download MySQL for free here:

<http://www.mysql.com/downloads/index.html>



## 4. Giải pháp trọn gói



- **LAMP – Linux**
- **WAMP – Windows**
- **XAMPP – Windows/Linux**



# Phần 2: PHP căn bản

# Nội dung phần 2



- 1. PHP và HTML**
- 2. Câu lệnh PHP**
- 3. Ghi chú trong PHP**
- 4. Khai báo biến**
- 5. Kiểu dữ liệu**
- 6. Toán tử**
- 7. Cấu trúc điều khiển**
- 8. Hàm**
- 9. Phạm vi của biến**
- 10. Biến tĩnh**
- 11. Làm việc với file và thư mục**
- 12. Lớp – Đối tượng trong PHP**

# 1. PHP và HTML



```
1 <?
2     echo "<P>This is a test using the <b>first</b> tag type.</P>";
3 >?
4
5 <?php
6     echo "<P>This is a test using the <b>second</b> tag type.</P>";
7 >?
8
9 <script language="php">
10     echo "<P>This is a test using the <b>third</b> tag type.</P>";
11 </script>
```

Thẻ mở

Thẻ đóng

<?

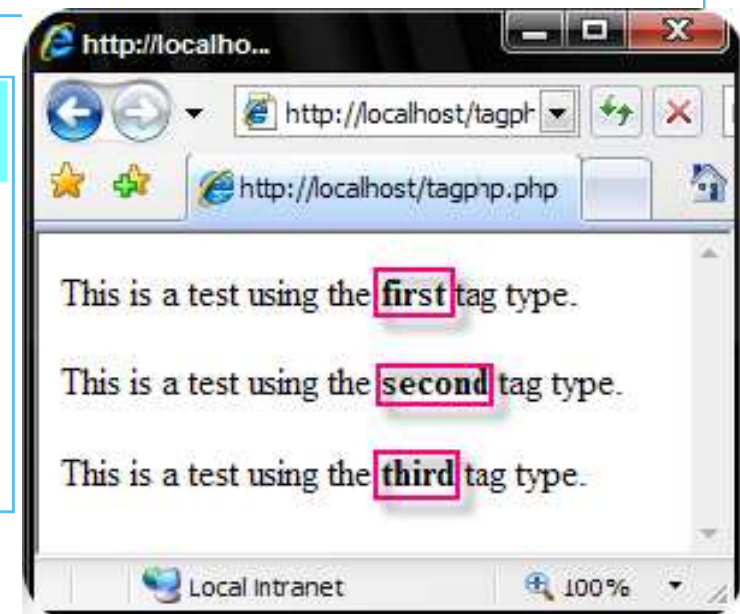
?>

<?php

?>

<script language="php">

<script>



## 2. Câu lệnh PHP



- **Phát biểu:**
  - Các câu lệnh php cách nhau bởi dấu ‘;’
  - Không phân biệt khoảng trắng, tab hay ký tự xuống dòng.
- **Ví dụ:**

```
<?php print "Hello"; print " World!"; ?>
<?php
    Print "Hello"           ;
    print " World!";
?>
```

# 3. Ghi chú trong PHP



- Ghi chú của PHP tương tự như của **C++** và **Perl**
- Có **3 loại** ghi chú:
  - Sử dụng **'//'** để ghi chú
  - Sử dụng **'#'** để ghi chú
  - Sử dụng **/\* ... \*/** để ghi chú
- Ví dụ:

```
<?
print "chào mọi người."; //print dùng để xuất chuỗi
?>
```

```
<?=$bien; ?> #tương tự như <? print $bien; ?>
```

# 4. Khai báo biến



- Cú pháp: `$ten_bien = value;`
- **Không khai báo kiểu dữ liệu**
- Biến tự động được khởi tạo khi gán giá trị lần đầu
- Tên biến
  - Bao gồm các ký tự (A..Z, a..z), ký số(0..9),\_
  - **Không được bắt đầu** bằng ký số (0..9)
  - **Phân biệt chữ hoa** – chữ thường
- Ví dụ:
  - Đúng cú pháp: `$hoten, $_pass`
  - Sai cú pháp: `$2host`

# 4. Khai báo biến



- **Variable variables**

- Cho phép thay đổi tên biến

- Ví dụ

```
<?
```

```
$varname="Bien_moi";
```

```
$$varname= "xyz"; //$Bien_moi= "xyz"
```

```
?>
```

- **Hằng số - Constants**

```
<?
```

```
define("MY_CONST", 10);
```

```
echo MY_CONST;
```

```
?>
```



# 5. Kiểu dữ liệu



- **boolean**
- **integer**
- **double**
- **string**
- **array**
- **object**

**Một biến trong PHP có thể lưu bất kỳ kiểu dữ liệu nào**

# 5. Kiểu dữ liệu



- **Chuyển đổi kiểu dữ liệu**

- Cách 1 (**automatic**)

- \$var = "100" + 15;

- \$var = "100" + 15.0;

- \$var = 39. "Steps"; //Chú ý

- Cách 2 (**datatype**) \$var

- Cách 3 **settype**(\$var, "**datatype**")

- **VD:**

| \$var    | (int)\$var | (bool)\$var | (string)\$var |
|----------|------------|-------------|---------------|
| null     | 0          | false       | ""            |
| true     | 1          |             | "1"           |
| false    | 0          |             | ""            |
| "6 feet" | 6          | true        |               |
| "foo"    | 0          | true        |               |

# 5. Kiểu dữ liệu



- Kiểm tra kiểu dữ liệu

|                         |                        |              |
|-------------------------|------------------------|--------------|
| <code>gettype</code>    | <code>is_string</code> | <b>isset</b> |
| <code>is_integer</code> | <code>is_array</code>  | <b>unset</b> |
| <code>is_double</code>  | <code>is_object</code> | <b>empty</b> |

- Ví dụ

```
$var = "test";  
if (isset($var))  
    echo "Variable is Set";  
if (empty($var))  
    echo "Variable is Empty";
```

# 5. Kiểu dữ liệu



- Một số hàm xử lý số

|       |       |        |                |
|-------|-------|--------|----------------|
| abs   | pow   | decbin | srand(seed)    |
| ceil  | sqrt  | bindec | rand           |
| floor | log   | dechex | rand(min, max) |
| round | log10 | hexdec | ...            |

- Ví dụ:

```
// Phát sinh một “mầm” ngẫu nhiên
```

```
$seed = (float) microtime()*1000000000;
```

```
// Khởi tạo bộ phát sinh số ngẫu nhiên
```

```
srand($seed);
```

```
// In số ngẫu nhiên
```

```
print rand(); // Giá trị ngẫu nhiên từ 0 đến getmaxrand( )
```

```
print rand(1, 6); // Giá trị ngẫu nhiên từ 1 đến 6
```

**Tra Hướng Dẫn  
trong PHP Manual**

# 5. Kiểu dữ liệu



- **Kiểu string**

- **Toán tử nối chuỗi “.”**

```
$str="Hello " . "World!"; // $str = "Hello World!";
```

- **Phân biệt dấu nháy đơn, dấu nháy kép**

```
$user = "Mr Bean";
```

```
print 'Hi $user'; // Hi $user
```

```
print "Hi $user"; // Hi Mr Bean
```

```
print "Hi". $user; // ????
```

```
print "Hi". "$user"; // ????
```

## 5. Kiểu dữ liệu: Kiểu chuỗi (tt)



- Giới hạn bởi nháy đơn (') hoặc kép (")
- Chuỗi đặt trong nháy kép bị thay thế và xử lý ký tự thoát. Trong nháy đơn thì không.
- Ví dụ:

```
$a = "Hello";
```

```
$b = "$a world"; //tương đương $b="Hello  
world"
```

```
$c = '$a world'; //$c='$a world' (không  
thay đổi)
```

## 5. Kiểu dữ liệu: Kiểu chuỗi (tt)



- Để làm rõ các biến trong chuỗi, cần bao biến vào giữa cặp ngoặc nhọn **{ }**

```
$a = "He";
```

```
$b = "$allo"; //lỗi vì PHP hiểu là  
    $allo
```

```
$c = "{$a}llo"; //đúng ($c = "Hello")
```

# 5. Kiểu dữ liệu: Kiểu chuỗi (tt)



- **Ký tự thoát: \**

- Sử dụng để viết các ký tự đặc biệt trong chuỗi

- VD:

**Cần có chuỗi: Người ta nói “PHP rất tốt”**

```
$a = "Người ta nói "PHP rất tốt""; //Sai
```

```
$a = "Người ta nói \"PHP rất tốt\""; //Đúng
```

- Một số ký tự phải sử dụng ký tự thoát: \$, \, “

- Ngoài ra:

- \n: Xuống dòng
- \r: trở về đầu dòng
- \t: dấu tab
- ...



# 5. Kiểu dữ liệu: Kiểu chuỗi (tt)



- Kiểu tài liệu (heredoc):

- Cho phép viết 1 chuỗi trên nhiều dòng.

- Không cần sử dụng ký tự thoát:

- Cách viết:

```
$biên = <<<Ký_hiệu
```

```
nội dung trên nhiều dòng
```

```
Ký_hiệu;
```

- Chú ý: Ký\_hiệu phải được viết ở ký tự đầu tiên của dòng

- Ví dụ:

```
$a = <<<EOD
```

```
Đây là chuỗi nằm trên nhiều dòng sử dụng cú  
pháp kiểu tài liệu 'heredoc'
```

```
EOD;
```

# 5. Kiểu dữ liệu: Kiểu chuỗi (tt)



## Kiểu Heredoc

```
<?php
$str = <<<EOA
Example of string <br>
spanning multiple lines<br>
using heredoc syntax.<br>
EOA;//không được có khoảng trắng đầu dòng

echo($str);
$name = "mr bean";
$d = date("d/m/y");
$str = <<<EOQ
This is a lecture of $name.<br>
Ngày $d<br>
EOQ;

echo($str);
?>
```

- **Các xử lý cơ bản**
  - **strlen**(\$chuỗi)
  - **substr**(\$chuỗi, \$vị trí, \$chiều\_dài)
  - **strtoupper** (\$chuỗi)
  - **strtolower** (\$chuỗi)
  - **iconv**(mã nguồn, mã đích, \$chuỗi)
  - **trim**(\$chuỗi, 'ký tự muốn cắt')
  - **ltrim**(\$chuỗi, 'ký tự muốn cắt')
  - **rtrim**(\$chuỗi, 'ký tự muốn cắt')

- **Tìm kiếm**
  - **strpos**(\$chuỗi, \$chuỗi\_con, \$vị\_trí\_bắt\_đầu)
  - **strrpos** (\$chuỗi, \$chuỗi\_con, \$vị\_trí\_bắt\_đầu)
- **So sánh**
  - **strcmp**(\$chuỗi\_1, \$chuỗi\_2)
  - **strncmp**(\$chuỗi\_1, \$chuỗi\_2, \$chiều\_dài)
  - **strcasecmp**(\$chuỗi\_1, \$chuỗi\_2)
  - **strncasecmp**(\$chuỗi\_1, \$chuỗi\_2, \$chiều\_dài)
  - **strnatcmp**(\$chuỗi\_1, \$chuỗi\_2)
  - **strnatcasecmp**(\$chuỗi\_1, \$chuỗi\_2)

# 5. Kiểu dữ liệu



```
<?php
$n = 43951789;
$u = -43951789;
$c = 65; // ASCII 65 is 'A'
```

```
printf("%b = '%b'\n", $n);
printf("%c = '%c'\n", $c);
printf("%d = '%d'\n", $n);
printf("%e = '%e'\n", $n);
printf("%u = '%u'\n", $n);
printf("%u = '%u'\n", $u);
printf("%f = '%f'\n", $n);
printf("%o = '%o'\n", $n);
printf("%s = '%s'\n", $n);
printf("%x = '%x'\n", $n);
printf("%X = '%X'\n", $n);
printf("%+d = '%+d'\n", $n);
printf("%+d = '%+d'\n", $u);
?>
```

## Hàm printf

```
%b = '10100111101010011010101101'
%c = 'A'
%d = '43951789'
%e = '4.39518e+7'
%u = '43951789'
%u = '4251015507'
%f = '43951789.000000'
%o = '247523255'
%s = '43951789'
%x = '29ea6ad'
%X = '29EA6AD'
%+d = '+43951789'
%+d = '-43951789'
```

# 5. Kiểu dữ liệu



```
<?php
$input = "Alien";
echo str_pad($input, 10); // produces "Alien      "
echo str_pad($input, 10, "--", STR_PAD_LEFT); // produces "-----Alien"
echo str_pad($input, 10, "_", STR_PAD_BOTH); // produces "__Alien__"
echo str_pad($input, 6, "___"); // produces "Alien_"
?>
```

## Hàm str\_pad

STR\_PAD\_RIGHT : Thêm vào bên phải (mặc định)  
STR\_PAD\_LEFT : Thêm vào bên trái  
STR\_PAD\_BOTH : Thêm cả hai phía

# 5. Kiểu dữ liệu



```
<?php

$array = array('lastname', 'email', 'phone');
$comma_separated = implode(",", $array);

echo $comma_separated; // lastname,email,phone

?>
```

**Xử lý chuỗi**

```
<?php
// Example 1
$pizza = "piece1 piece2 piece3 piece4 piece5 piece6";
$pieces = explode(" ", $pizza);
echo $pieces[0]; // piece1
echo $pieces[1]; // piece2
// Example 2
$data = "foo:*:1023:1000::/home/foo:/bin/sh";
list($user,$pass,$uid,$gid,$gecos,$home,$shell) = explode(":", $data);
echo $user; // foo
echo $pass; // *
?>
```

# 5. Kiểu dữ liệu



- Mảng (**Array**)

```
<?php
$colors = array('red', 'blue', 'green', 'yellow');
sort($colors);
foreach ($colors as $color)
{
    echo "Do you like $color?<br>";
}
$num = count($colors);
echo "We have $num items<br>";
?>
```

```
Do you like blue?
Do you like green?
Do you like red?
Do you like yellow?
We have 4 items
```



# 5. Kiểu dữ liệu



- Mảng (**Array**)

```
<?php
// PHP 5
foreach ($colors as &$color) {
    $color = strtoupper($color);
}
unset($color); /* ensure that following writes to
$color will not modify the last array element */

// Workaround for older versions (phiên bản trước PHP 5)
foreach ($colors as $key => $color) {
    $colors[$key] = strtoupper($color);
}

print_r($colors);
?>
```

```
Array
(
    [0] => RED
    [1] => BLUE
    [2] => GREEN
    [3] => YELLOW
)
```

# 5. Kiểu dữ liệu



```
<?php
$a = array("a" => "apple", "b" => "banana");
$b = array("a" => "pear", "b" => "strawberry", "c" => "cherry");

$c = $a + $b; // Union of $a and $b
echo "Union of \$a and \$b: \n";
var_dump($c);

$c = $b + $a; // Union of $b and $a
echo "Union of \$b and \$a: \n";
var_dump($c);
?>
```

Union of \$a and \$b:

```
array(3)
{
    ["a"]=> string(5) "apple"
    ["b"]=> string(6) "banana"
    ["c"]=> string(6) "cherry"
}
```

Union of \$b and \$a:

```
array(3)
{
    ["a"]=> string(4) "pear"
    ["b"]=> string(10) "strawberry"
    ["c"]=> string(6) "cherry"
}
```

# 5. Kiểu dữ liệu



```
<?php
/* Suppose that $var_array is an array returned from
   wddx_deserialize */
$size = "large";
$var_array = array("color" => "blue",
                  "size"  => "medium",
                  "shape" => "sphere");
extract($var_array, EXTR_PREFIX_SAME, "wddx");
echo "$color, $size, $shape, $wddx_size\n";
?>
```

**blue, large, sphere, medium**

# 5. Kiểu dữ liệu



```
<?php
$info = array('coffee', 'brown', 'caffeine');

// Listing all the variables
list($drink, $color, $power) = $info;
echo "$drink is $color and $power makes it special.\n";

// Listing some of them
list($drink, , $power) = $info;
echo "$drink has $power.\n";

// Or let's skip to only the third one
list( , , $power) = $info;
echo "I need $power!\n";

// list() doesn't work with strings
list($bar) = "abcde";
var_dump($bar); // NULL
?>
```

- **Sắp xếp**

## Theo giá trị

- **sort**(\$mảng) / **asort**(\$mảng) // tăng dần
- **rsort**(\$mảng) / **arsort**(\$mảng) // giảm dần
- **natsort**(\$mảng) / **natcasesort**(\$mảng) // tăng dần, dùng cho chuỗi
- **usort**(\$mảng, "hàm\_so\_sánh") // tự định nghĩa thứ tự
- **uasort**(\$mảng, "hàm\_so\_sánh") // tự định nghĩa thứ tự

## Theo khóa

- **ksort**(\$mảng) // tăng dần
- **krsort**(\$mảng) // giảm dần
- **uksort**(\$mảng, "hàm\_so\_sánh") // tự định nghĩa thứ tự

- **Nối ghép hai mảng**  
**array\_merge**(\$mảng1, \$mảng2)  
**array\_combine**(\$mảng1, \$mảng2)  
**array\_intersect**(\$mảng1, \$mảng2)
- **Tìm kiếm**  
**array\_search**(\$giá\_trị, \$mảng)

# 6. Toán tử



| Loại      | Toán tử                                 | Ghi chú                    |
|-----------|---|----------------------------|
|           | new                                     |                            |
|           | . [] ()                                 |                            |
| Toán học  | + - * / % ++ --                         |                            |
| So sánh   | < > <= >= != == === !==                 |                            |
| Luận lý   | &&    ?: ,                              |                            |
| Xử lý bit | ! ~ << >> >>><br>AND OR XOR             |                            |
| Gán       | = += -= *= /= %=<br>>>= <<= &=  = ^= .= |                            |
| Ép kiểu   | (kiểu dữ liệu)                          | (int) (double) (string)... |

# 6. Toán tử



| <b>Ex</b>     | <b>Name</b>              | <b>Result</b>  |
|---------------|--------------------------|--|
| $\$a == \$b$  | Equal                    | TRUE if \$a is equal to \$b.   |
| $\$a === \$b$ | Identical                | TRUE if \$a is equal to \$b, and they are of the same type. (PHP 4)        |
| $\$a != \$b$  | Not equal                | TRUE if \$a is not equal to \$b.   |
| $\$a <> \$b$  | Not equal                | TRUE if \$a is not equal to \$b.   |
| $\$a !== \$b$ | Not identical            | TRUE if \$a is not equal to \$b, or they are not of the same type. (PHP 4) |
| $\$a < \$b$   | Less than                | TRUE if \$a is strictly less than \$b.                                     |
| $\$a > \$b$   | Greater than             | TRUE if \$a is strictly greater than \$b.                                  |
| $\$a <= \$b$  | Less than or equal to    | TRUE if \$a is less than or equal to \$b.                                  |
| $\$a >= \$b$  | Greater than or equal to | TRUE if \$a is greater than or equal to \$b.                               |



# 7. Cấu trúc điều khiển



- Điều kiện **if**
- Điều khiển **switch**
- Vòng lặp **for**
- Vòng lặp **while**
- Vòng lặp **do.. While**
- Vòng lặp **foreach**
- Từ khóa **break, continue**

Tương tự như C++

**if** (biểu thức điều kiện)

khối lệnh 1

**else**

khối lệnh 2

# 7. Cấu trúc điều khiển



- Rẽ nhánh với **if ... else**

```
if ($a > $b) {  
    echo "a > b";  
    $a = $b;  
} else {  
    echo "a <= b";  
    $b = $a;  
}
```

# Câu lệnh switch



**switch** (biểu thức)

{

**case** biểu thức 1:  
khối lệnh 1

**case** biểu thức 2:  
khối lệnh 2

...

**case** biểu thức n:  
khối lệnh n

**default:**  
khối lệnh cuối

}

# 7. Cấu trúc điều khiển



- Rẽ nhánh với **switch ... case**

```
switch ($a) {  
    case 0:  
        echo "a = 0";  
        break;  
    case 1:  
        echo "a = 1";  
        break;  
}
```

# Vòng lặp while/do...while



**while** (biểu thức điều kiện)  
khối lệnh

Hay:

**do**

khối lệnh

**while** (biểu thức điều kiện);

# 7. Cấu trúc điều khiển



- Lặp với **while & do ... while**

```
while ($i++ < 5) {  
    switch ($i) {  
        case 2:  
            echo "At 2"; break;  
        case 5:  
            echo "At 5"; break 2;  
        default:  
            break;  
    }  
}
```

# Vòng lặp for



**for** (biểu thức 1; biểu thức 2; biểu thức 3)  
khởi lệnh

- **biểu thức 1: thực hiện 1 lần khi bắt đầu vòng lặp**
- **biểu thức 2: điều kiện lặp, được xem xét trước mỗi lần lặp**
- **biểu thức 3: thực hiện sau mỗi lần lặp**



# 7. Cấu trúc điều khiển



- **Lặp với for & foreach**

```
for ($i=1, $j=0; $i<=10; $j+=$i, print $i,
    $i++);
```

```
$arr = array(1, 2, 3, 4);
foreach ($arr as &$value) {
    $value = $value * 2;
}
```

```
$arr = array("one", "two", "three");
foreach ($arr as $key => $value) {
    echo "Key: $key; Value: $value";
}
```

- **break**

Ngưng và thoát ra khỏi vòng lặp hiện tại

- **continue**

Ngưng thực hiện lần lặp hiện hành để chuyển sang lần lặp tiếp theo

# 7. Cấu trúc điều khiển



```
<?php
$arr = array(1, 2, 3, 4);
foreach ($arr as &$value) {
    $value = $value * 2;
}
// $arr is now array(2, 4, 6, 8)
unset($value); // break the reference with the last element
?>
```

```
<?php
$arr = array("one", "two", "three");
reset($arr); // reset pointer, start again on first element
while (list(, $value) = each($arr)) {
    echo "Value: $value<br />\n";
}

foreach ($arr as $value) {
    echo "Value: $value<br />\n";
}
?>
```

# 7. Cấu trúc điều khiển



```
<?php
$arr = array("one", "two", "three");
reset($arr);
while (list($key, $value) = each($arr)) {
    echo "Key: $key; Value: $value<br />\n";
}

foreach ($arr as $key => $value) {
    echo "Key: $key; Value: $value<br />\n";
}
?>
```

```
Key: 0; Value: one
Key: 1; Value: two
Key: 2; Value: three
Key: 0; Value: one
Key: 1; Value: two
Key: 2; Value: three
```

# 8. Hàm



```
<?php
function takes_array($input)
{
    echo "$input[0] + $input[1] = ", $input[0]+$input[1];
}

$input = array(4,7);
takes_array($input);
?>
```

**lưu ý: dấu &**

```
<?php
function add_some_extra(&$string)
{
    $string .= 'and something extra.';
}

$str = 'This is a string, ';
add_some_extra($str);
echo $str; // outputs 'This is a string, and something extra.'
?>
```

# 8. Hàm



- Định nghĩa

```
<?php
```

```
function tên_hàm([danh sách tham số ...])
```

```
{
```

```
    [thân hàm ...]
```

```
}
```

```
?>
```

- Gọi hàm

Nhập tên hàm (không phân biệt chữ in hoa-thường) và cung cấp đầy đủ các tham số cần thiết trong cặp dấu ()

# 8. Hàm



- **Kết thúc và trả kết quả**

Lệnh **return** dùng để kết thúc và trả kết quả cũng như quyền điều khiển lại cho nơi đã gọi hàm. Nếu không có lệnh return thì mặc định hàm trả về giá trị **NULL**.

***Muốn trả về hơn một giá trị thì phải dùng mảng***

- **Truyền tham số**

Mặc định các tham số được truyền vào bên trong hàm theo phương pháp tham trị. Trường hợp muốn thay đổi trực tiếp trên các tham số truyền thì người ta dùng phương pháp tham chiếu, thêm dấu **&** trước tên tham số (khi định nghĩa) cũng như tên biến được truyền làm tham (khi gọi hàm)

# 8. Hàm



- **Tham số có giá trị mặc định**  
Tương tự cách khai báo và gán giá trị đầu tiên cho biến, thông thường loại tham số này nên đặt cuối trong danh sách tham số  
Khi gọi hàm nếu bỏ trống tại vị trí tham số có giá trị mặc định thì mặc nhiên giá trị mặc định được dùng cho tham số đó
- **Hàm có số lượng tham số không xác định**  
Khai báo danh sách tham số rỗng ()  
Sử dụng các hàm sau để lấy danh sách các tham số:  
**func\_num\_args():** số lượng tham số khi hàm được gọi  
**func\_get\_arg(i):** giá trị các tham số thứ **i** được truyền (bắt đầu từ 0)  
**func\_get\_args():** danh sách tất cả các tham số



# 8. Hàm



- **Biến tĩnh**

Thêm từ khóa **static** khi khai báo biến

Được khởi tạo (và gán giá trị) một lần đầu tiên duy nhất trong suốt quá trình thực thi của script

- **Sử dụng biến toàn cục**

Khai báo lại biến toàn cục với từ khóa **global** (bên trong hàm) để có thể sử dụng được biến toàn cục này bên trong hàm

Sử dụng các hàm sau để lấy danh sách các tham số:

**func\_num\_args()**: số lượng tham số khi hàm được gọi

**func\_get\_arg(i)**: giá trị các tham số thứ i được truyền (bắt đầu từ 0)

**func\_get\_args()**: danh sách tất cả các tham số

# 8. Hàm



- **Phạm vi**

Có giá trị sử dụng trong toàn script, ngay cả trước và sau khi định nghĩa

- **Lồng hàm**

Cho phép định nghĩa lồng hàm, thậm chí lồng bên trong một cấu trúc điều khiển (if, switch, while/do, while...)

Loại hàm này có phạm vi trong toàn script và không thể định nghĩa lại

# 8. Hàm



```
<?php
function makecoffee($type = "cappuccino")
{
    return "Making a cup of $type.\n";
}
echo makecoffee();
echo makecoffee(null);
echo makecoffee("espresso");
?>
```

```
<?php
function makecoffee($types = array("cappuccino"), $coffeeMaker = NULL)
{
    $device = is_null($coffeeMaker) ? "hands" : $coffeeMaker;
    return "Making a cup of ".join(", ", $types)." with $device.\n";
}
echo makecoffee();
echo makecoffee(array("cappuccino", "lavazza"), "teapot");
?>
```

# 8. Hàm



```
<?php
function makeyogurt($type = "acidophilus", $flavour)
{
    return "Making a bowl of $type $flavour.\n";
}

echo makeyogurt("raspberry");// won't work as expected
?>
```

```
<?php
function makeyogurt($flavour, $type = "acidophilus")
{
    return "Making a bowl of $type $flavour.\n";
}

echo makeyogurt("raspberry");// works as expected
?>
```

# 8. Hàm



```
<?php
function square ($num)
{
    return $num * $num;
}
echo square (4); //outputs '16'
?>
```

```
<?php
function &returns_reference()
{
    return $someref;
}
$newref=&returns_reference();
?>
```

```
<?php
function small_numbers()
{
    return array (0, 1, 2);
}
list($zero,$one,$two)=small_numbers();
?>
```

# 8. Hàm



```
<?php
function foo() {
    echo "In foo()<br />\n";
}
function bar($arg = '')
{
    echo "In bar(); argument was '$arg'.<br />\n";
}
// This is a wrapper function around echo
function echoit($string)
{
    echo $string;
}
$func = 'foo';
$func();          // This calls foo()

$func = 'bar';
$func('test');   // This calls bar()

$func = 'echoit';
$func('test');   // This calls echoit()
?>
```

# 8. Hàm



```
<?php
class Foo
{
    function Variable()
    {
        $name = 'Bar';
        $this->$name(); // This calls the Bar() method
    }

    function Bar()
    {
        echo "This is Bar";
    }
}
$foo = new Foo();
$funcname = "Variable";
$foo->$funcname(); // This calls $foo->Variable()
?>
```

# 9. Phạm vi của biến



```
<?php
$a = 1; $b = 2;
function Sum()
{
    global $a, $b;
    $b = $a + $b;
}
Sum();
echo $b;
?>
```

3

```
<?php
$a = 1;
$b = 2;
function Sum()
{
    $GLOBALS["b"] = $GLOBALS["a"] + $GLOBALS["b"];
}
Sum();
echo $b;
?>
```



# 9. Phạm vi của biến



```
<?php
    $a = 1;
    include "b.inc";
?>
```

**Phạm vi của biến [a] có tác dụng trong cả b.inc**

---

```
<?php
$a = 1;      // global scope

function Test()
{
    echo $a; // reference to local scope variable
}

Test();
?>
```

**Biến [a] ở ngoài và trong hàm Test khác nhau**

# 9. Phạm vi của biến



| Tên biến          | Mô tả   |
|-------------------|---|
| <b>\$GLOBALS</b>  | chứa tất cả các biến toàn cục mà đoạn script cục bộ có thể truy cập |
| <b>\$_SERVER</b>  | chứa thông tin về môi trường của web server                         |
| <b>\$_GET</b>     | chứa thông tin của yêu cầu dạng GET                                 |
| <b>\$_POST</b>    | chứa thông tin của yêu cầu dạng POST                                |
| <b>\$_COOKIE</b>  | chứa thông tin từ HTTP cookie                                       |
| <b>\$_FILE</b>    | chứa thông tin từ việc tải tập tin lên kiểu POST                    |
| <b>\$_ENV</b>     | chứa thông tin về môi trường thực thi của đoạn script               |
| <b>\$_REQUEST</b> | chứa thông tin nhập vào của người dùng                              |
| <b>\$_SESSION</b> | chứa thông tin của các session được đăng ký                         |

# 10. Biến tĩnh



```
<?php
function Test()
{
    $a = 0;
    echo $a;
    $a++;
}
Test();
Test();
?>
```

```
<?php
function Test()
{
    static $a = 0;
    echo $a;
    $a++;
}
Test();
Test();
?>
```



# 11. Làm việc với tập tin và thư mục



1. Sử dụng `include()`
2. Sử dụng `include_once()`
3. Sử dụng `include_path` cho các dự án lớn
4. Kiểm tra file
5. Đọc, ghi file
6. Làm việc với thư mục

# 11. Làm việc với tập tin và thư mục



- Lệnh **include()** cho phép bạn phối hợp giữa các file trong một PHP project, ko giống như cú pháp `#include` của ngôn ngữ C, lệnh này không chèn mã lệnh vào file mà thực thi file php giống như cú pháp gọi hàm
- **include()** sử dụng để chia sẻ các hàm dùng chung, các đoạn mã chung trong một project có nhiều file
- Nếu không tìm thấy file, **include()** thông báo warning nhưng không dừng chương trình
- PHP cung cấp 1 lệnh tương tự **include()** là **require()**, lệnh này có sự khác biệt là sẽ dừng ngay chương trình khi không tìm thấy file

# 11. Làm việc với tập tin và thư mục



```
//vars.php
<?php

$color = 'green';
$fruit = 'apple';

?>
```

```
//test.php
<?php

echo "A $color $fruit";
// A
include 'vars.php';
echo "A $color $fruit";
// A green apple

?>
```

Vì `include()` thực hiện lời gọi đến file php, do đó bạn có thể trả về giá trị từ file PHP được include

```
<!--File1.php-->
<?php

    return 4 + 4;

?>

-----

<!--File2.php-->
<?php
echo "This is from file 2<br>";
$retVal = include("file1.php");
echo "Value file 1: $retVal<br>";
echo "This is from file 2\n";

?>
```

# 11. Làm việc với tập tin và thư mục



- Có thể đặt lệnh include bên trong 1 cấu trúc điều kiện hoặc cấu trúc lặp,
- Khi đó tùy theo điều kiện của cấu trúc mà include() có được thực hiện hay không, 1 hay nhiều lần
- Việc này giúp hỗ trợ cho việc thiết kế kiến trúc trang web tốt hơn.

# 11. Làm việc với tập tin và thư mục



- `include_once()` giống như `include()`, tuy nhiên có điểm khác biệt là chỉ include 1 lần, lần sau nếu gặp lại file này thì ko include nữa
- `include_once()` phân biệt chữ hoa, chữ thường

```
<?php
include_once("a.php");
// this will include a.php
include_once("A.php");
// this will include a.php again on Windows!
?>
```

Vì phân biệt chữ hoa/thường nên  
`include_one` chèn thêm lần thứ hai



# 11. Làm việc với tập tin và thư mục



- **include** file theo đường dẫn tuyệt đối: Cách này dở vì khi cài đặt trên máy khác sẽ không tìm thấy file được include
- **include** file theo đường dẫn tương đối: Cách này tốt hơn, nhưng mỗi khi đổi vị trí của file được include thì phải sửa lại tại tất cả các file thực hiện lời gọi include
- Cách tốt nhất là sử dụng **include\_path** (thiết lập trong file PHP.INI) đối với những file thư viện dùng chung được sử dụng nhiều (giống như đối với ngôn ngữ C)

# 11. Làm việc với tập tin và thư mục



- thay đổi `include_path` trong `PHP.INI`
- dùng lệnh `set_include_path()`

```
<?php
var_dump(get_include_path());
set_include_path('/inc'); // Works as of PHP 4.3.0
var_dump(get_include_path());
restore_include_path();
var_dump(get_include_path());
?>
```

- dùng lệnh `ini_set()`

```
<?php
var_dump(ini_get("include_path"));
ini_set("include_path", "/inc"); // Works in all PHP versions
var_dump(ini_get("include_path"));
ini_restore("include_path");
var_dump(ini_get("include_path"));
?>
```

# 11. Làm việc với tập tin và thư mục



file\_exist(), is\_file(), is\_dir(), is\_readable(),  
is\_writable(), is\_executable(), filesize(), fileatime()

●<?php

```
function outputFileTestInfo( $file ) {  
    if ( ! file_exists( $file ) ) {  
        print "$file does not exist<br/>";  
        return;  
    }  
    print "$file is ".(is_file( $file )?"":"not ")."a file<br/>\n";  
    print "$file is ".(is_dir( $file )?"":"not ")."a directory<br/>\n";  
    print "$file is ".(is_readable( $file )?"":"not ")."readable<br/>\n";  
    print "$file is ".(is_writable( $file )?"":"not ")."writable<br/>\n";  
    print "$file is ".( filesize($file) )." bytes<br/>\n";  
    print "$file was accessed on ".date( "D d M Y g:i A",  
        fileatime($file ))."<br/>";  
    print "$file was modified on ".date( "D d M Y g:i A",  
        filemtime( $file))."<br/>";  
    print "$file was changed on".date( "D d M Y g:i A",  
        filectime($file))."<br/>";  
}  
outputFileTestInfo( "c:\\windows\\system32\\cmd.exe" );  
?>
```

# 11. Làm việc với tập tin và thư mục



```
fopen($filename, $mode);
```

```
fwrite($handle, $string);
```

```
fread($handle, $length);
```

```
fgets($handle);
```

```
sprintf($format);
```

```
fscanf($handle, $format);
```

```
fseek($handle, $offset);
```

```
fclose($handle);
```

```
file_get_contents($filename);
```

# 11. Làm việc với tập tin và thư mục



```
<?php
$var1 = 10;
$var2 = "This is a String";
$var3 = true;
$f = fopen("test.txt", "wt");
fwrite($f, "$var1 $var2
$var3\n");
fwrite($f,
"$var1\n$var2\n$var3\n");
fclose($f);

echo "Read line by
line.....\n";
$f = fopen("test.txt", "rt");
while (!feof($f)) {
    $line = fgets($f);
    echo "$line";
}
fclose($f);
```

```
echo "Read all file by
fread.....\n";
$f = fopen("test.txt", "rb");
$myfile = fread($f,
filesize("test.txt"));
echo $myfile;
fclose($f);

echo "Read all file.....\n";
$myfile =
file_get_contents("test.txt");
echo($myfile);
?>
```

# 11. Làm việc với tập tin và thư mục



```
<?php
$var1=10;
$var2=100;
$var3=100.3434;
$var4="Test string";
$f=fopen("test.txt", "wt");
fwrite($f, sprintf("%d %10.3f %10.3lf\n\r", $var1, $var2,
$var3));
fwrite($f, sprintf("%s", $var4));
fclose($f);

$f=fopen("test.txt", "rt");
if (list($v1, $v2, $v3, $v4) = fscanf($f, "%d %f %lf\n\r%s"))
{
    var_dump($v1);
    var_dump($v2);
    var_dump($v3);
    var_dump($v4);
}
$v4 = fgets($f);
var_dump($v4);
fclose($f);
?>
```

# 11. Làm việc với tập tin và thư mục



```
<?php
class AClass {
};

$obj1 =& new AClass();
$obj1->a = 10;
$obj1->b = 100.023;
$obj1->c = "Test String";
var_dump($obj1);

$f = fopen("test.txt", "wb");
fwrite($f, serialize($obj1));
fclose($f);

$f = fopen("test.txt", "rb");
$obj2 = unserialize(fgets($f));
fclose($f);
var_dump($obj2);

?>
```

# 11. Làm việc với tập tin và thư mục



- mkdir(), rmdir()
- opendir(), readdir(), closedir()

```
<?php
```

```
    $dir=opendir("c:\\windows");  
    while ($file=readdir($dir)) {  
        echo "$file\n";  
    }  
    closedir($dir);
```

```
?>
```



## Viết một script thực hiện công việc sau:

- Yêu cầu người dùng nhập vào một số nguyên
- Đem so sánh số vừa nhập với một số nguyên cho trước
- Nếu bằng thì xuất ra câu chúc mừng người dùng đã đoán đúng con số bí mật
- Ngược lại thì thông báo kết quả là con số vừa nhập là lớn hay bé hơn con số bí mật và yêu cầu nhập lại cho đến khi nhập đúng

# 12. Lớp và Đối tượng trong PHP



- Khai báo lớp
- Hàm tạo
- Phạm vi
- Kế thừa
- Hàm serialize, unserialize

# 12. Lớp và Đối tượng trong PHP



- **Khai báo**

```
<?php
class Cat {
}

$fluffy = new Cat();
echo "Fluffy is a new ".gettype($fluffy)."!";
?>
```

Fluffy is a new object!

- **Hàm tạo**

```
<?php
class Cat {
    // Constructor
    function Cat() {
    }
}
?>
```

```
<?php
class Cat {
    // Constructor
    Function __constructor() {
    }
}
?>
```

# 12. Lớp và Đối tượng trong PHP



- Khai báo lớp

```
class tên_lớp
```

```
{
```

```
    các thuộc tính và phương thức
```

```
}
```

- Tạo và hủy một đối tượng

```
$tên_biến = new tên_lớp();
```

Đối tượng sẽ tự động bị hủy khi không còn tham chiếu nào đến nó

```
$tên_biến = NULL;
```

# 12. Lớp và Đối tượng trong PHP



- **Các từ khóa khai báo:**
  - **public**: có thể sử dụng bên ngoài lớp
  - **private**: chỉ sử dụng cục bộ bên trong lớp
  - **protected**: sử dụng được bởi các lớp kế thừa
  
- **Một số quy tắc chung:**
  - **Không thể khai báo hai method trùng tên**
  - Method phải được khai báo ngay bên trong khai báo lớp
  - Dùng biến giả **\$this** để truy xuất các member và method trong lớp
  - Dùng toán tử **->** để truy xuất đến member và method

# 12. Lớp và Đối tượng trong PHP



- Khai báo constructor và destructor

```
public function __construct(danh sách tham số)
{
    khởi tạo giá trị các member
}
```

*constructor được tự động thực hiện khi đối tượng được tạo*

```
public function __destruct()
{
    dọn dẹp
}
```

*destructor được tự động thực hiện khi đối tượng bị hủy*

# 12. Lớp và Đối tượng trong PHP



```
<?php
class Cat {
    // How old the cat is
    var $age;

    // Constructor
    function Cat($new_age){

        // Set the age of this cat to the new age
        $this->age = $new_age;
    }
    //The birthday method increments the age variable
    function Birthday(){

        $this->age++;
    }
}
// Create a new instance of the cat object that's one year old
$fluffy = new Cat(1);
echo "Age is $fluffy->age <br />";
echo "Birthday<br/>";
// Increase fluffy's age
$fluffy->Birthday();
echo "Age is $fluffy->age <br />";
?>
```

Age is 1  
Birthday  
Age is 2

- Khai báo

```
const TÊN_HẰNG = giá trị;
```

- Truy xuất

```
tên_lớp::TÊN_HẰNG           // ngoài lớp
```

```
seft::TÊN_HẰNG           // trong lớp
```



- Khai báo

... **static** \$thuộc\_tính

- Truy xuất

tên\_lớp::\$thuộc\_tính // ngoài lớp

seft::\$thuộc\_tính // trong lớp

- Khai báo

... **static** function phương\_thức(...)

- Truy xuất

tên\_lớp::phương\_thức(...) // ngoài lớp

seft:: phương\_thức(...) // trong lớp

- Khai báo lớp con

```
class lớp_con extends lớp_cha
```

```
{
```

```
    các thuộc tính và phương thức
```

```
}
```

***Tất cả các member và method được khai báo public hay protected trong lớp cha được thừa kế và có thể sử dụng trong lớp con***

# Lập trình hướng đối tượng: phương thức nạp chồng



- Gọi một method lớp cha

**parent::phương\_thức(...)**

*Bằng cách định nghĩa lại một phương thức đã có ở lớp cha, tất cả các lời gọi đến phương thức này mà không chỉ định rõ như trên sẽ được hiểu là gọi phương thức có cùng tên của lớp con*

- Khai báo lớp trừu tượng

```
abstract class lớp_trừu_tượng  
{  
    // các thuộc tính  
    abstract public function  
    phương_thức_trừu_tượng(...);  
    ...  
    // các phương thức khác  
}
```

*Không thể tạo đối tượng trực tiếp từ lớp trừu tượng  
Lớp con bắt buộc phải định nghĩa các phương thức  
trừu tượng của lớp cha*

# Lập trình hướng đối tượng:

## ngăn kế thừa và nạp chồng



- Lớp không thể kế thừa

**final class** không\_thể\_kế\_thừa { ... }

- Phương thức không thể nạp chồng

**final public function** không\_thể\_nạp\_chồng(...) { ... }

# Lập trình hướng đối tượng: **interface**



- **Khai báo Interface**

```
interface giao_diện  
{  
    public function phương_thức();  
    ...  
}
```

- **Khai báo lớp theo mẫu Interface**

```
abstract class tên_lớp implements giao_diện  
{  
    ...  
}
```

*Các lớp sử dụng Interface hay kế thừa từ một lớp sử dụng Interface bắt buộc phải định nghĩa tất cả các phương thức trong Interface đó*

# 12. Lớp và Đối Tượng trong PHP



```
<?php
class Cat {
    // How old the cat is
    var $age;

    function Cat($new_age){

        // Set the age of this cat to the new age
        $this->age = $new_age;
    }
    function Birthday(){

        $this->age++;
    }
}
class Domestic_Cat extends Cat {
    // Constructor
    function Domestic_Cat() {
    }

    // Sleep like a domestic cat
    function sleep() {
        echo("Zzzzzz.<br />");
    }
}
$fluffy=new Domestic_Cat();
$fluffy->Birthday();
$fluffy->sleep();
echo "Age is $fluffy->age <br />";
?>
```



Zzzzzzzz.  
Age is 1



# 12. Lớp và Đối Tượng trong PHP



1. Hàm `serialize()` dùng để lưu trữ đối tượng, hàm trả về một chuỗi các byte để lưu thông tin của đối tượng
2. Hàm `unserialize()` dùng để khôi phục đối tượng được lưu giữ bởi hàm `serialize()`

```
<?php
```

```
class AClass {
    var $a;
    function AClass() {
    }
};
$obj1 = new AClass();
$obj1->a = 10;
$obj1->b = 100;
$obj1->c = "Nguyen Ngoc Thuy Hang";

$lưu = serialize($obj1);
echo "$lưu <br>";
$obj2 = unserialize($lưu);
var_dump($obj2);
```

```
?>
```

# Câu hỏi và thảo luận





# LẬP TRÌNH WEB PHP

## Chương 03: PHP nâng cao

- 1. Truyền, nhận dữ liệu**
- 2. Cookies**
- 3. Session**
- 4. Upload File**
- 5. Image**

# 1. Truyền, Nhận dữ liệu



- **Trang web nhập dữ liệu**
  - **Cách 1:** truyền dữ liệu trực tiếp qua url
    - Sử dụng liên kết để sửa lại url
    - Thông qua đối tượng location
  - **Cách 2:** truyền dữ liệu thông qua form
    - Sử dụng đối tượng `<form>`
    - Nhập liệu thông qua các `formfield`
    - Thực hiện việc truyền dữ liệu thông qua `submit`
    - Hai phương thức gửi dữ liệu `GET / POST`
- **Trang web nhận dữ liệu** (dùng các biến toàn cục)
  - **Cách 1:** `$_GET["biến"]`
  - **Cách 2:** `$_POST["biến"]`
  - **Cách 3:** `$_REQUEST["biến"]`

## 2. Khái niệm cookie



- Là những mẫu thông tin nhỏ được lưu trên máy của người sử dụng thường dùng để lưu trữ thông tin về người sử dụng.
- Khả năng tạo cookie phụ thuộc vào trình duyệt và sự cho phép của người sử dụng.
- Trong IE, để xem các cookies đã được tạo thì vào: *Tools/Internet Options/Setting.../View Files*

## 2. Thiết lập cookie



- **setcookie**(tên, giá\_trị, t)
  - Tên: Tên cookie được tạo ra
  - giá\_trị: giá trị được đặt cho cookie
  - t: Số: thời gian hết hạn của cookie
- **Chú ý:**
  - Lệnh **setcookie** phải được gọi trước khi gửi bất cứ nội dung gì về client (Trước các thẻ HTML, trước echo, print)
  - Để thiết lập thời gian hết hạn của cookie thường sử dụng hàm **time()**+khoảng thời gian (tính bằng giây)

# Đọc cookie



- Sử dụng mảng `$_COOKIE` với chỉ các số là tên các cookie.
- Chú ý:
  - Ta không thể đọc cookie vừa được thiết lập ngay trong cùng 1 trang vừa thiết lập gọi `setcookie`.



# 3. Sử dụng SESSION



- Để sử dụng session, cần khai báo:  
*session\_start();* ngay đầu trang PHP.
- Khai báo biến session:  
*\$\_SESSION[ 'tên\_biến' ] = trị;*
- Hủy bỏ một biến session:  
*unset( \$\_SESSION[ 'tên\_biến' ] );*
- Hủy toàn bộ các biến session:  
*session\_destroy();*

# Ví dụ: Kiểm tra đăng nhập dùng Session



login.php

```
<?php
if ($_POST['submit'] == "Login")
{
    //script to check user name and password would be coded here

    //if authentication is successfull
    session_start();
    $_SESSION['access'] = "yes";
    header(Location:access.php);
}

if ($_POST['submit'] == "Log Out")
{
    //if the user decides to exit
    session_destroy();
}

?>
```

# Ví dụ: Kiểm tra đăng nhập dùng Session



**access.php**

```
<?php
// if the user accesses this page, make sure they have been authenticated through login.php

if ($_SESSION['access'] != "yes")
{
    header (Location:login.php) ;
}

?>
```

# 3. SESSION



```
<?php
// page1.php
session_start();
echo 'Welcome to page #1';
$_SESSION['favcolor'] = 'green';
$_SESSION['animal']    = 'cat';
$_SESSION['time']      = time();
// Works if session cookie was accepted
echo '<br /><a href="page2.php">page 2</a>';
// Or maybe pass along the session id, if needed
echo '<br /><a href="page2.php?' . SID . '">page 2</a>';
?>
```

```
<?php
// page2.php
session_start();
echo 'Welcome to page #2<br />';
echo $_SESSION['favcolor']; // green
echo $_SESSION['animal'];   // cat
echo date('Y m d H:i:s', $_SESSION['time']);
// You may want to use SID here, like we did in page1.php
echo '<br /><a href="page1.php">page 1</a>';
?>
```

## 4. Upload tập tin lên server



- Gửi các file lên server (upload) là một chức năng không thể thiếu đối với các ứng dụng Web-based
- Để form có thể upload file, ta phải xác định thuộc tính **enctype** của form có giá trị "**multipart/form-data**"
- Với PHP, bạn có thể xác định kích thước lớn nhất của file upload bằng cách đặt thêm 1 hidden field có tên `MAX_FILE_SIZE` trước control file

# 4. Upload tập tin lên server



## Mảng `$_FILE`

| Element                                     | Contains   | Example                     |
|---|--|-----------------------------|
| <code>\$FILES['taptin']['name']</code>      | Name of uploaded file                              | <b>test.gif</b>             |
| <code>\$_FILES['taptin']['tmp_name']</code> | Path to temporary file                             | <b>/tmp/phprDfZvN</b>       |
| <code>\$_FILES['taptin']['size']</code>     | Size (in bytes) of uploaded file                   | <b>6835</b>                 |
| <code>\$_FILES['taptin']['error']</code>    | An error code corresponding to a PHP constant      | <b>UPLOAD_ERR_FORM_SIZE</b> |
| <code>\$_FILES['taptin']['type']</code>     | MIME type of uploaded file (where given by client) | <b>image/gif</b>            |

# 4. Upload tập tin lên server



## Mã lỗi

| Constant Name        | Value | Explanation   |
|----------------------|-------|---|
| UPLOAD_ERR_OK        | 0     | No problem  |
| UPLOAD_ERR_INI_SIZE  | 1     | File size exceeds php.ini limit set in upload_max_filesize        |
| UPLOAD_ERR_FORM_SIZE | 2     | File size exceeds limit set in hidden element named MAX_FILE_SIZE |
| UPLOAD_ERR_PARTIAL   | 3     | File only partially uploaded                                      |
| UPLOAD_ERR_NO_FILE   | 4     | File was not uploaded   |

# 4. Upload tập tin lên server



```
1 <html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4 <title>ví dụ upload tập tin</title>
5 </head>
6 <body>
7 <form action="xuly_upload.php" method="post" enctype="multipart/form-data">
8     <input type="file" name="taptin" />
9     <input type="submit" name="Submit" value="Submit" />
10 </form>
11 </body>
12 </html>
```





```

1 <html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4 <title>Upload file</title>
5 <?php
6 function SaveUploadFile($strfile,$strpath)
7 {
8     $ketqua="";
9     if(isset($_FILES[$strfile]))
10    {
11        if($_FILES[$strfile]['error'] == UPLOAD_ERR_OK)
12        {
13            $name=$_FILES[$strfile]['name'];
14            if(move_uploaded_file($_FILES[$strfile]['tmp_name'],$strpath.$name))
15                $ketqua="Ok!";
16            else
17                $ketqua="Không di chuyển đợc file!";
18        }else
19            $ketqua="File bị lỗi!";
20    }else
21        $ketqua="Không tồn tại file upload!";
22    return $ketqua;
23 }
24 ?>
25 </head>
26 <body>
27 <?php
28     echo SaveUploadFile('taptin','./');
29 ?>
30 </body>
31 </html>

```

## 4. Upload tập tin lên server

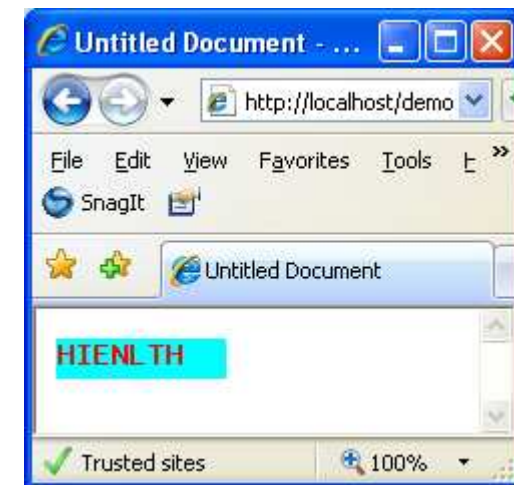
# 5. Vẽ image



```
1 <?php
2     $text = "HIENLTH";
3     header("Content-type: image/png");
4     $im = @imagecreatetruecolor(85,20) or die("Không thể tạo file ảnh");
5     $bg = imagecolorallocate($im, 0, 255, 255);
6     imagefill($im, 1, 1, $bg);
7     $text_color = imagecolorallocate($im, 222, 18, 9);
8     imagestring($im, 18, 1, 0, $text, $text_color);
9     imagepng($im);
10    imagedestroy($im);
11 ?>
```

## Vẽ ảnh từ văn bản

```
1 <html>
2 <body bgcolor=green>
3 
4 </body>
5 </html>
```



# 6. Date



```
1 <html>
2 <head>
3 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4 <title>Untitled Document</title>
5 </head>
6 <?php
7 $tomorrow = mktime(0, 0, 0, date("m") , date("d")+1, date("Y"));
8 $lastmonth = mktime(0, 0, 0, date("m")-1, date("d"), date("Y"));
9 $nextyear = mktime(0, 0, 0, date("m"), date("d"), date("Y")+1);
10 print "Ngày hôm nay: <b>".date("d-m-Y")."</b><br>";
11 print "Ngày mai: <b>".date("D-M-Y", $tomorrow)."</b><br>";
12 print "Tháng trước: <b>".date("D-M-Y", $lastmonth)."</b><br>";
13 print "Năm sau: <b>".date("D-M-Y", $nextyear)."</b><br>";
14 print "Ngày mai: <b>".date("d-m-Y", $tomorrow)."</b><br>";
15 print "Tháng trước: <b>".date("d-m-Y", $lastmonth)."</b><br>";
16 print "Năm sau: <b>".date("d-m-Y", $nextyear)."</b><br>";
17 ?>
18 <body>
19 </body>
20 </html>
```

# 6. Date



Ngày hôm nay: 02-07-2008  
Ngày mai: Thu-Jul-2008  
Tháng trước: Mon-Jun-2008  
Năm sau: Thu-Jul-2009  
Ngày mai: 03-07-2008  
Tháng trước: 02-06-2008  
Năm sau: 02-07-2009

```
<?php
// Assuming today is: March 10th, 2001, 5:16:18 pm
$today = date("F j, Y, g:i a");           // March 10, 2001, 5:16 pm
$today = date("m.d.y");                   // 03.10.01
$today = date("j, n, Y");                  // 10, 3, 2001
$today = date("Ymd");                      // 20010310
$today = date('h-i-s, j-m-y, it is w Day z ');
                                           // 05-16-17, 10-03-01, 1631 1618 6 Fripm01
$today = date('\i\t \i\s \t\h\e jS \d\a\y. '); // It is the 10th day.
$today = date("D M j G:i:s T Y");         // Sat Mar 10 15:16:08 MST 2001
$today = date('H:m:s \m \i\s\ \m\o\n\t\h'); // 17:03:17 m is month
$today = date("H:i:s");                   // 17:16:17
?>
```

# Câu hỏi và thảo luận





# LẬP TRÌNH WEB PHP

## Chương 04: Kết nối PHP với MySQL

- 1. Các kiểu dữ liệu cơ bản trong MySQL**
- 2. Các lệnh thông dụng trong MySQL**
- 3. Kết nối MySQL từ PHP**
- 4. Quy trình kết nối vào MySQL**
- 5. Các bước truy cập CSDL MySQL**

# 1. Các kiểu dữ liệu cơ bản trong MySQL

| Kiểu                        | Mô tả  |
|-----------------------------|--|
| <b>char</b> (length)        | tối đa 255 ký tự, chiều dài cố định = <i>length</i>              |
| <b>varchar</b> (length)     | tối đa 255 ký tự, chiều dài động $\leq$ <i>length</i>            |
| <b>text</b>                 | tối đa 65536 ký tự   |
| <b>int</b> (length)         | -2.147.483.648 đến +2.147.483.647                                |
| <b>decimal</b> (length,dec) | tối đa <i>length</i> chữ số trong đó <i>dec</i> chữ số thập phân |



# 1. Các kiểu dữ liệu thông dụng trong MySQL



| Kiểu                                   | Mô tả  |
|--|--|
| <b>enum</b> ("option1", "option2",...) | tập hợp tự định, nghĩa tối đa 65.535 giá trị |
| <b>date</b>                            | yyyy-mm-dd                                   |
| <b>time</b>                            | hh:mm:ss                                     |
| <b>datetime</b>                        | yyyy-mm-dd hh:mm:ss                          |

## 2. Các lệnh thông dụng trong MySQL



| Kiểu               | Mô tả                                |
|--------------------|--------------------------------------|
| <b>CREATE</b>      | tạo CSDL hoặc bảng                   |
| <b>ALTER</b>       | thay đổi bảng có sẵn                 |
| <b>SELECT</b>      | chọn dữ liệu từ bảng                 |
| <b>DELETE</b>      | xóa dữ liệu khỏi bảng                |
| <b>DESCRIBE</b>    | xem thông tin mô tả về cấu trúc bảng |
| <b>INSERT INTO</b> | ghi giá trị vào bảng                 |
| <b>UPDATE</b>      | cập nhật dữ liệu đã có trong bảng    |
| <b>DROP</b>        | xóa bảng hay toàn bộ CSDL            |

## 2. Các lệnh thông dụng trong MySQL



```
CREATE INDEX indexname ON tablename (column [ASC|DESC],  
...);
```

```
CREATE PROCEDURE procedurename( [parameters] ) BEGIN ...  
END;
```

```
CREATE TABLE tablename
```

```
(  
    column    datatype [NULL|NOT NULL]    [CONSTRAINTS],  
    column    datatype [NULL|NOT NULL]    [CONSTRAINTS],  
    ...  
);
```

```
CREATE USER username[@hostname] [IDENTIFIED BY  
[PASSWORD] 'password'];
```

```
CREATE [OR REPLACE] VIEW viewname AS SELECT ...;
```

## 2. Các lệnh thông dụng trong MySQL



**ALTER TABLE** *tablename*

(

**ADD** column datatype [ **NULL|NOT**  
**NULL**] [ **CONSTRAINTS** ],

**CHANGE** column columns datatype [ **NULL|NOT**  
**NULL**] [ **CONSTRAINTS** ],

**DROP** column,

...

);

## 2. Các lệnh thông dụng trong MySQL



**SELECT** *columnname*, ...

**FROM** *tablename*, ...

[**WHERE** ...]

[**UNION** ...]

[**GROUP BY** ...]

[**HAVING** ...]

[**ORDER BY** ...];

**VD: SELECT \* FROM ketqua;**

## 2. Các lệnh thông dụng trong MySQL



**DELETE FROM** *tablename*

**[WHERE ...];**

**VD:**

**DELETE FROM ketqua WHERE  
MaSV = 'K29.103.010'**

## 2. Các lệnh thông dụng trong MySQL



**INSERT INTO** *tablename* [(columns, ...)]  
**VALUES**(values, ...);

**INSERT INTO** *tablename* **VALUES**(*value1*,  
*value2*, ..., *valuen*);

**VD:**

**INSERT INTO** *ketqua* (*mamon*, *diem*) **VALUES**  
(*'LTWEB'*, 10);

**INSERT INTO** *ketqua* **VALUES**(*'*, *'K29.103.010'*,  
*'LTWEB'*, 10);

## 2. Các lệnh thông dụng trong MySQL



**UPDATE** *tablename*

**SET** *columnname* = value, ...

**[WHERE ...];**

**VD:**

**UPDATE** ketqua **SET** diem = 10 **WHERE** MaSV  
= 'K29.103.010'



## 2. Các lệnh thông dụng trong MySQL



**DROP DATABASE | INDEX | PROCEDURE |  
TABLE | TRIGGER | USER | VIEW itemname;**

**VD:**

- Xóa bảng SINHVIEN: **DROP TABLE SinhVien**
- Xóa CSDL QLSV: **DROP DATABASE QLSV;**

- **Kết nối mysql server**

`mysql [-h hostname] [-P portnumber] -u username -p`

`mysql [-h hostname] [-P portnumber] --user=user --  
password=pass`

Nhập lệnh sau dấu nhắc lệnh ***mysql***>

Mỗi lệnh SQL kết thúc bằng dấu ;

# Giao tiếp đồ họa



- **Một số công cụ thông dụng**

SQLyog Enterprise

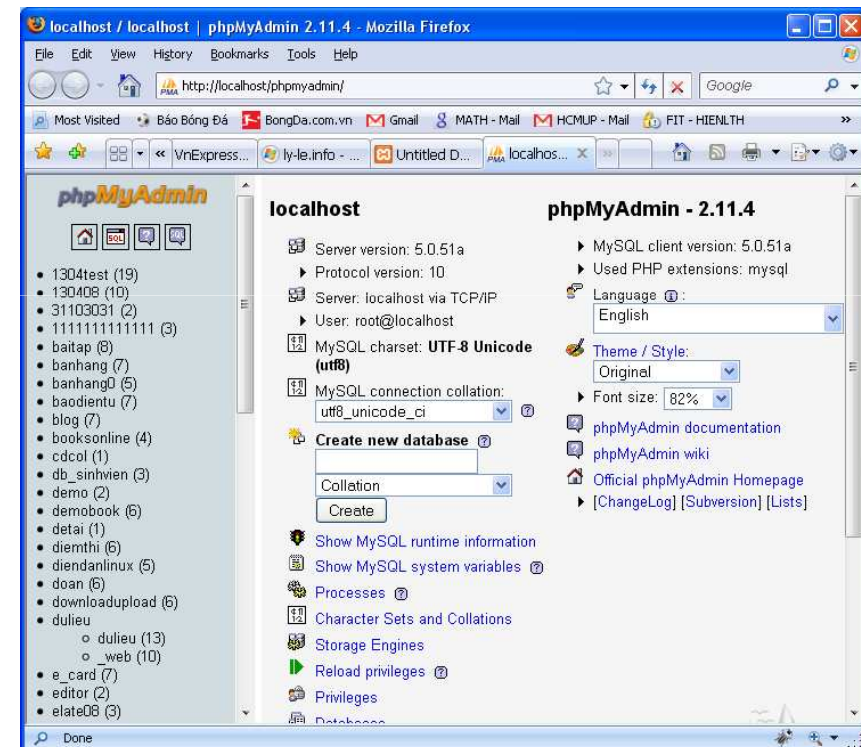
phpMyAdmin

MySQL Query Browser

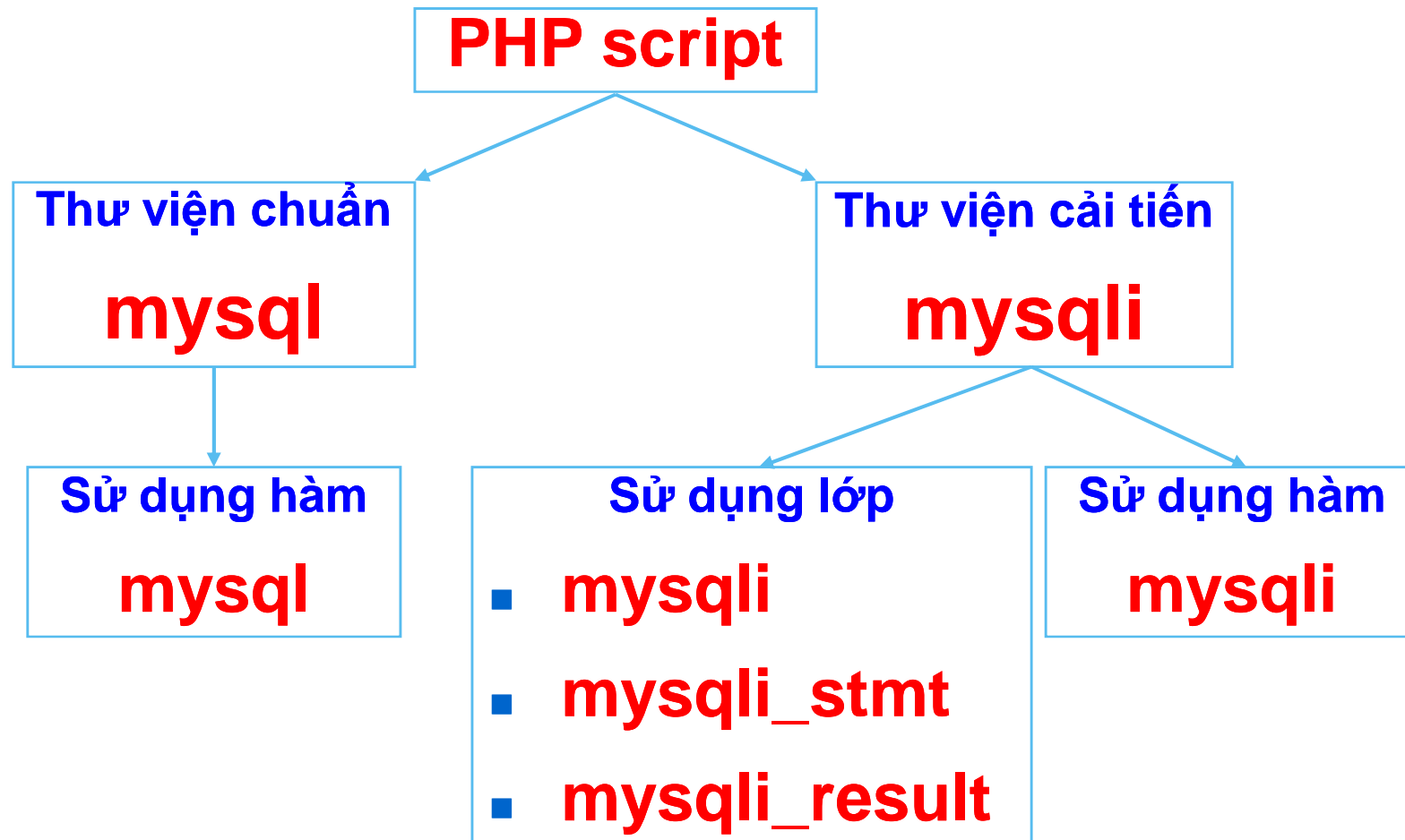
MySQL Maestros

Navicat

MySQL Manager



# 3. Kết nối MySQL từ PHP



- **Thiết lập trong php.ini**  
**extension=php\_mysql.dll**
- **Ưu điểm**
  - Hỗ trợ lập trình hướng đối tượng
  - Hỗ trợ nhân bản và phân tán CSDL
  - Nén và mã hóa dữ liệu trên kết nối
  - Tối ưu hiệu năng và mã

# 4. Quy trình kết nối vào MySQL



- 1. Mở kết nối đến CSDL**
- 2. Chọn CSDL**
- 3. Chọn bảng mã (nếu cần)**
- 4. Xử lý CSDL**
- 5. Dọn dẹp**
- 6. Đóng kết nối**

# Bước 1: Mở kết nối đến CSDL



```
// OOP mysqli
```

```
$mysqli = new mysqli('hostname', 'username',  
                    'password', 'dbname');
```

```
// mysqli
```

```
$link = mysqli_connect('hostname',  
                        'username', 'password', 'dbname');
```

## Bước 2: Chọn CSDL



```
// OOP mysqli
```

```
$mysqli->select_db('dbname');
```

```
// mysqli
```

```
mysqli_select_db($link, 'dbname');
```



## Bước 3: Chọn bảng mã (nếu cần)



```
// OOP mysqli
```

```
mysqli->query($link, "SET NAMES 'character set'")
```

```
// mysqli
```

```
mysqli_query($link, "SET NAMES 'character set'")
```

**VD: SET NAMES UTF8**

- **Truy vấn**

```
// OOP mysqli
```

```
$result = mysqli->query("query")
```

```
// mysqli
```

```
$result = mysqli_query($link, "query")
```

# Bước 4: Xử lý CSDL (tt)



- **Lấy dữ liệu từ truy vấn**

**// OOP mysqli**

**\$row = \$result->fetch\_row()**

**\$row = \$result->fetch\_assoc()**

**\$row = \$result->fetch\_array(*result\_type*)**

**// mysqli**

**\$row = *mysqli\_fetch\_row*(\$result)**

**\$row = *mysqli\_fetch\_assoc*(\$result)**

**\$row = *mysqli\_fetch\_array*(\$result, *result\_type*)**

# Bước 5: Dọn dẹp



```
// OOP mysqli  
$result->close()
```

```
// mysqli  
mysqli_free_result($result)
```

# Bước 6: Đóng kết nối



```
// OOP mysqli
$mysqli->close()

// mysqli
mysqli_close($link)
```

## 5. Các bước truy cập CSDL MySQL



1. Tạo kết nối đến database server
2. Lựa chọn CSDL
3. Xây dựng truy vấn và thực hiện truy vấn
4. Xử lý kết quả trả về
5. Đóng kết nối đến server

# Khai báo sử dụng CSDL MySQL



## • Khai báo kết nối CSDL:

```
<?php
    $dbhost = 'localhost';
    $dbuser = 'root';
    $dbpass = 'password';
    $conn = mysqli_connect($dbhost, $dbuser,
    $dbpass) or die (Không thể kết nối CSDL
    MySQL' );
?>
```

## • Chọn Database làm việc

```
$dbname = 'bookstore';
mysqli_select_db($dbname);
```

## • Giải phóng Database

```
mysqli_close($conn);
```

# Khai báo sử dụng CSDL MySQL (tt)



- Truy vấn dữ liệu:

```
$result = mysqli_query( 'câu_truy_vấn' );
```

- Giải phóng tài nguyên truy vấn

```
mysqli_free_result( $result );
```

- Sử dụng kết quả truy vấn:

```
mysqli_fetch_array( $result );
```

```
mysqli_fetch_row( $result );
```

```
mysqli_fetch_assoc( $result );
```

- Sử dụng tiếng việt:

```
mysqli_query( "SET CHARACTER SET UTF8" );
```

```
Hay mysqli_query( "SET NAMES UTF8" );
```



- Hàm **die** ( "*Chuỗi*" ): Đưa ra thông báo và kết thúc.
- Với cách viết trên, **die** chỉ thực hiện khi lệnh trước nó không thành công
- Các hàm cần thiết:
  - **mysqli\_affected\_rows()**: Số bản ghi bị tác động bởi lệnh **mysqli\_query** liền trước.
  - **mysqli\_error()**: Thông báo lỗi (nếu có)
  - **mysqli\_errno()**: Mã lỗi

- Kết nối CSDL MySQL sử dụng PDO

## PHP Data Objects

# Cấu hình để sử dụng PDO



- **Need to enable PDO in your php.ini**
- **MUST load PDO first**
- **Unix:**
  - `extension=pdo.so`
  - `extension=pdo_XXX.so`
- **Windows**
  - `extension=php_pdo.dll`
  - `extension=php_pdo_XXX.dll`

# Connecting via PDO



```
try {  
    $dbh = new PDO($dsn,  
        $user, $password, $options);  
} catch (PDOException $e) {  
    echo "Failed to connect:"  
        . $e->getMessage();  
}
```

# DSN format in PDO



- ***Driver:optional\_driver\_specific\_stuff***
  - ***sqlite:/path/to/db/file***
  - ***sqlite::memory:***
  - ***mysql:host=name;dbname=dbname***
  - ***pgsql:native\_pgsql\_connection\_string***
  - ***oci:dbname=dbname;charset=charset***
  - ***firebird:dbname=dbname;charset=charset;role=role***
  - ***odbc:odbc\_dsn***

# Connection management



```
try {  
    $dbh = new PDO($dsn, $user, $pw);  
} catch (PDOException $e) {  
    echo "connect failed:" . $e->getMessage();  
}  
  
// use the database here  
// ...  
  
// done; release the connection  
$dbh = null;
```

# Let's get data



```
$dbh = new PDO($dsn);  
$stmt = $dbh->prepare(  
    'SELECT * FROM FOO');  
$stmt->execute();  
while ($row = $stmt->fetch()) {  
    print_r($row);  
}
```

# Fetch types



- **\$stmt->fetch(PDO\_FETCH\_BOTH)**
  - Array with numeric and string keys
  - default option
- **PDO\_FETCH\_NUM**
  - Array with numeric keys
- **PDO\_FETCH\_ASSOC**
  - Array with string keys
- **PDO\_FETCH\_OBJ**
  - \$obj->name holds the 'name' column from the row
- **PDO\_FETCH\_BOUND**
  - Just returns true until there are no more rows



# Let's change data



```
$deleted = $dbh->query(  
    "DELETE FROM FOO WHERE 1");
```

```
$changes = $dbh->query(  
    "UPDATE FOO SET active=1 "  
    . "WHERE NAME LIKE '%joe%'");
```

- Quoting is annoying, but essential
- PDO offers a better way

```
$stmt->prepare('INSERT INTO CREDITS  
    (extension, name) VALUES (:extension,  
    :name)');
```

```
$stmt->execute(array(  
    ':extension' => 'xdebug',  
    ':name' => 'Derick Rethans'  
));
```

# Binding for output



```
$stmt = $dbh->prepare(  
    "SELECT extension, name from CREDITS");  
if ($stmt->execute()) {  
    $stmt->bindColumn('extension', $extension);  
    $stmt->bindColumn('name', $name);  
    while ($stmt->fetch(PDO_FETCH_BOUND)) {  
        echo "Extension: $extension\n";  
        echo "Author: $name\n";  
    }  
}
```

# Transactions



```
try {  
    $dbh->beginTransaction();  
    $dbh->query('UPDATE ...');  
    $dbh->query('UPDATE ...');  
    $dbh->commit();  
} catch (PDOException $e) {  
    $dbh->rollBack();  
}
```

# Câu hỏi và thảo luận



Đại Học Sư Phạm Tp. Hồ Chí Minh



# LẬP TRÌNH WEB PHP

## Chương 05: PROJECT

- Thiết kế giao diện, bố cục
- Hiện thị tiếng Việt trong trang Web.
- Xây dựng hàm kết nối CSDL dùng chung.
- Login.
- Upload/Download
- Trang sản phẩm: xem, đặt mua (viewer) thêm, cập nhật, xóa (admin)
- Phân trang
- Xuất bản website

# Layout



- **Tùy chọn. Ít nhất sẽ có:**
  - Banner (hình ảnh công ty, tổ chức, cá nhân, ...)**
  - Menu/Link (Các liên kết trong trang)**
  - Main (Nội dung trang web)**
  - Footer (Bản quyền – Địa chỉ)**

|  |                                   |       |
|--|-----------------------------------|-------|
| BANNER   |                                   |       |
| Link: <a href="#">Link 1</a>   <a href="#">Link 2</a>   <a href="#">Link 3</a> |                                   |       |
| LEFT   | MAIN<br>Đây là nội dung trang web | RIGHT |
| FOOTER   |                                   |       |

|  |                                   |  |
|--|-----------------------------------|--|
| BANNER   |                                   |  |
| Link: <a href="#">Link 1</a>   <a href="#">Link 2</a>   <a href="#">Link 3</a> |                                   |  |
| LEFT   | MAIN<br>Đây là nội dung trang web |  |
| FOOTER   |                                   |  |



- Thẻ meta trong file HTML/PHP  

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```
- DreamWeaver: **Ctrl + J** chọn **Encoding: UTF8.**
- CSDL: 

```
mysqli_query("SET NAMES utf8;");
```

# Hàm kết nối CSDL dùng chung



```
//File: connect.php
define("HOST", "localhost");
define("USR", "root");
define("PWD", "");
define("DBNAME", "baitap");
$conn = mysqli_connect(HOST,USR,PWD)
or die("Không kết nối được!");
    mysqli_select_db(DBNAME);
```

# Login Page



- Xây dựng file dùng để đăng nhập

ĐĂNG NHẬP

Username:

Password:



**user**

```
? username : varchar(30)
? password : varchar(32)
? email : varchar(100)
# type : int(1)
```

```
<form action="xulyform.php" method="post" name="form1">
  <p>ĐĂNG NHẬP</p>
  <p>Username:
    <input type="text" name="user" id="textfield" /><br />
    Password:
    <input type="password" name="pass" id="textfield2" />
  <br />
  <input type="submit" name="button" id="button" value="LOGIN" />
  <input type="reset" value="Clear" />
</p>
</form>
```

# Process Login Page



```
require("connect.php ");

$sql = "SELECT count(*) FROM user WHERE
  username='{$_POST['user']}' AND password
  ='" .md5($_POST['pass'])."'";
$result = mysqli_query($sql);

if(mysqli_num_rows($result)){
  echo 'Đăng nhập thành công';
  $_SESSION['user'] = $_POST[ user'];
}
else{
  //echo 'Đăng nhập lại';
  header('location:login.php');
}
```

# Upload

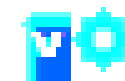


- Xây dựng file *upload.php* để upload file.

UPLOAD FILE

Chủ đề:

Files:



file

```
idfile : int(11)
chude : varchar(50)
tenfile : varchar(50)
loaifile : varchar(5)
# count : int(11)
```

```
<form action="upload.php" method="post" enctype="multipart/form-data"
name="upload">
  <p>UPLOAD FILE</p>
  <p>
    Chủ đề: <input type="text" name="chude"/><br/>
    Files: <input type="file" name="uploadedfile" /><br />
    <input type="submit" name="submit" id="button" value="Upload" />
  </p>
</form>
```

# Xử lý upload



- **Khai báo nơi lưu trữ:**

```
$target_path = "data/";
```

```
$target_path = $target_path.basename(  
    $_FILES['uploadedfile']['name']);
```

- **Upload file lên server:**

```
move_uploaded_file($_FILES['uploadedfile']['  
    tmp_name'], $target_path)
```

- **Lưu 1 số thông tin vào CSDL**

```
$sql = "INSERT INTO file (chude, tenfile,  
    loaifile) VALUES ('{$_POST['chude']}',  
    '{$target_path}', '$loaifile')";
```

# Download file



- Cách 1: Trực tiếp qua thẻ <a>
- Cách 2: Xây dựng trang download

```
function DownloadFile($file) { // $file = include path
    if(file_exists($file)) {
        header('Content-Description: File Transfer');
        header('Content-Type: application/octet-stream');
        header('Content-Disposition: attachment;
filename='.basename($file));
        header('Content-Transfer-Encoding: binary');
        header('Expires: 0');
        header('Cache-Control: must-revalidate, post-check=0, pre-check=0');
        header('Pragma: public');
        header('Content-Length: ' . filesize($file));
        ob_clean();
        flush();
        readfile($file);
        exit;
    }
}
```

- **Gọi: `DownloadFile($_GET[filename]);`**



# WEB SẢN PHẨM

PRESENTED BY

HIEN, LUONG TRAN HY  
[HIENLTH@HCMUP.EDU.VN](mailto:HIENLTH@HCMUP.EDU.VN)



- **Người dùng (viewer):**
  - Xem thông tin từng chủng loại sản phẩm.
  - Xem chi tiết 01 sản phẩm
  - Mua hàng
- **Quản trị (admin)**
  - Tạo 01 (loại) sản phẩm mới.
  - Chỉnh sửa/Xóa thông tin 01 sản phẩm
  - Gửi hàng

- Sử dụng biến truyền dạng GET, mặc định là trang đầu tiên nếu biến chưa đặt trị.

```
if (isset($_GET[page])) $page = $_GET[page];  
else $page = 1;
```

- Số lượng sản phẩm trình bày trên 01 trang:

```
$pp = ??
```

- Truy vấn CSDL chú ý: **LIMIT \$start, \$pp**

□ Trong đó  $\$start = (\$page - 1) * \$pp;$



# XUẤT BẢN SITE

PRESENTED BY

HIEN, LUONG TRAN HY  
HIENLTH@MATH.HCMUP.EDU.VN

# Xuất bản website



## Welcome Home to the Future

[Home](#) | [Sign Up For Free Hosting](#)

Fill out the form below and your free hosting account will be created.

|   |   |
|---|---|
| Username                                | <input type="text" value="hpt777"/>           |
| Password                                | <input type="password" value="*****"/>        |
| Email Address                           | <input type="text" value="hpt777@gmail.com"/> |
| Site Category                           | <input type="text" value="Personal"/> ▼       |
| Site Language                           | <input type="text" value="English"/> ▼        |
| Security Code                           | EE109   |
| Enter Security Code                     | <input type="text" value="EE109"/>            |
| <input type="button" value="Register"/> |   |

- <http://vn2k.net/signup.php>
- **Điền các thông tin theo mẫu**
  - Tên đăng nhập (username)
  - Mật khẩu (password)
  - Email
  - Loại site và ngôn ngữ
  - Điền mã xác nhận

# Xuất bản website (tt)






- Lưu giữ các thông tin về website

| Subscription To               | Terms  | Amount     |
|-------------------------------|--|------------|
| Free Hosting Plan             | \$0.00 USD for each month                                    | \$0.00 USD |
| <b>Control Panel</b>          |  |            |
| Control panel user name       | lamso_1774398  |            |
| Control panel password        | [REDACTED]   |            |
| Control panel URL             | http://cpanel.vn2k.net                                       |            |
| <b>Files and FTP Settings</b> |  |            |
| FTP user name                 | lamso_1774398  |            |
| FTP password                  | [REDACTED]   |            |
| FTP host name                 | ftp.vn2k.net   |            |
| File system path              | /home/vol4/vn2k.net/lamso_1774398/htdocs                     |            |
| <b>MySQL Settings</b>         |  |            |
| MySQL user                    | lamso_1774398  |            |
| MySQL password                | [REDACTED]   |            |
| MySQL host                    | sql103.vn2k.net  |            |
| MySQL port                    | 3306   |            |
| <b>Domain name</b>            |  |            |
| Your domain name              | http://hpt7777.vn2k.net<br>or<br>http://www.hpt7777.vn2k.net |            |
| <b>Save your settings</b>     |  |            |
| Save a copy to your computer  | <a href="#">Download my account settings</a>                 |            |

# Upload thông tin lên host

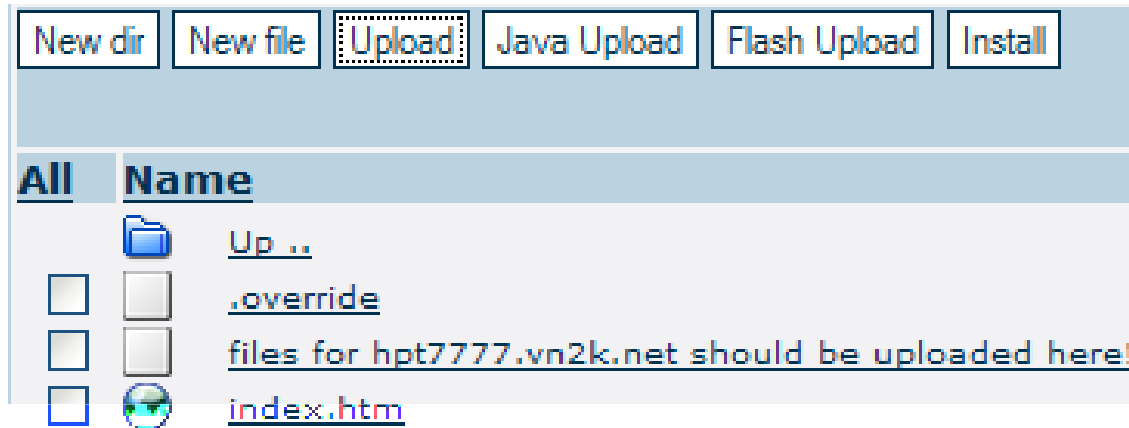


- Bằng FTP (dùng Total Commander)
- Trực tiếp: Vào Cpanel → File Manager

| All                      | Name   | Type      |
|--------------------------|--|-----------|
|                          |  <u>Up ..</u>   |           |
| <input type="checkbox"/> |  <u>htdocs</u>  | Directory |
| <input type="checkbox"/> |  <u>DO NOT UPLOAD FILES HERE DO NOT UPLOAD FILES HERE</u> | File      |

**File lưu trữ trong thư mục này**

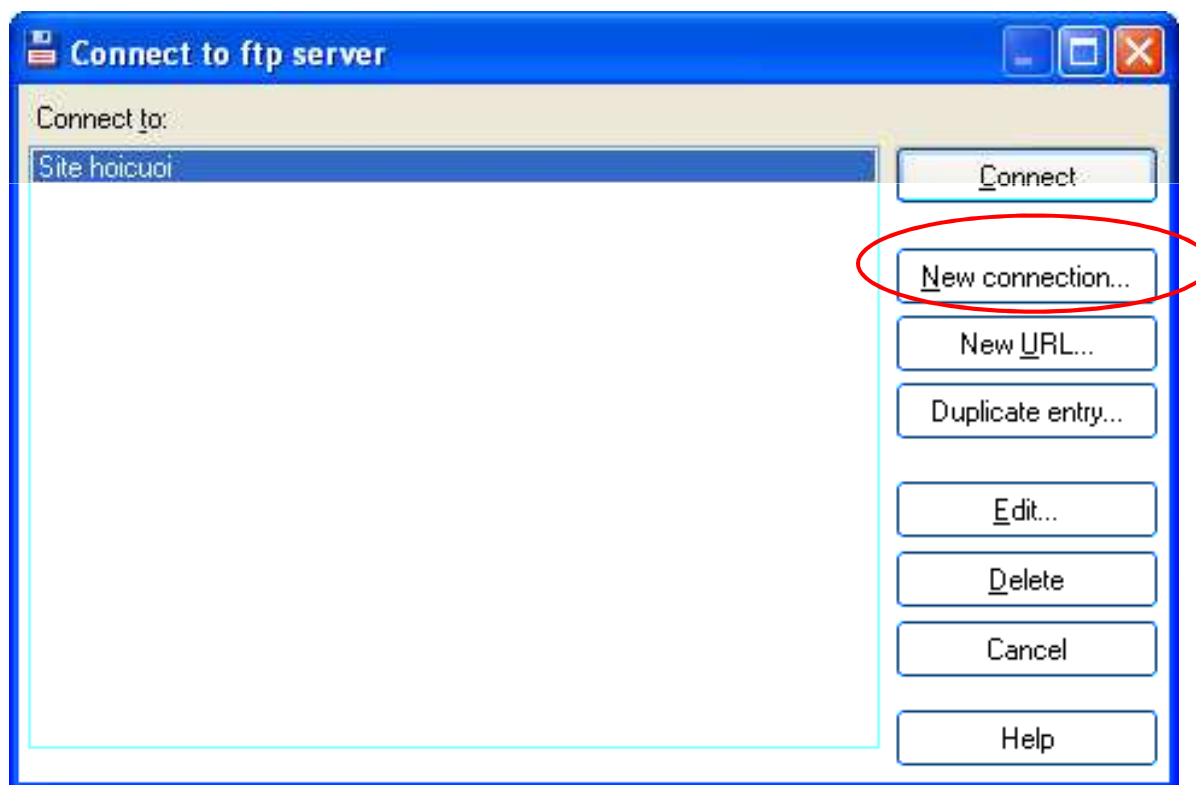
# Upload thông tin lên host



- **Chọn Upload** Bấm đây để Upload



- **Bấm nút FTP**





# Upload thông tin lên host dùng Total Commander



FTP: connection details

Session: ten bat ky

Host name[:Port]: ftp.vn2k.net

SSL/TLS Anonymous login (e-mail address as password)

User name: lamso\_1774398

Password: xxxxxx

Warning: Storing the password is insecure!

Remote Dir:

Local dir: >>

Send Commands:

Server type: Auto detect

Use firewall (proxy server) Define new Change...

Use passive mode for transfers (like a WWW browser)

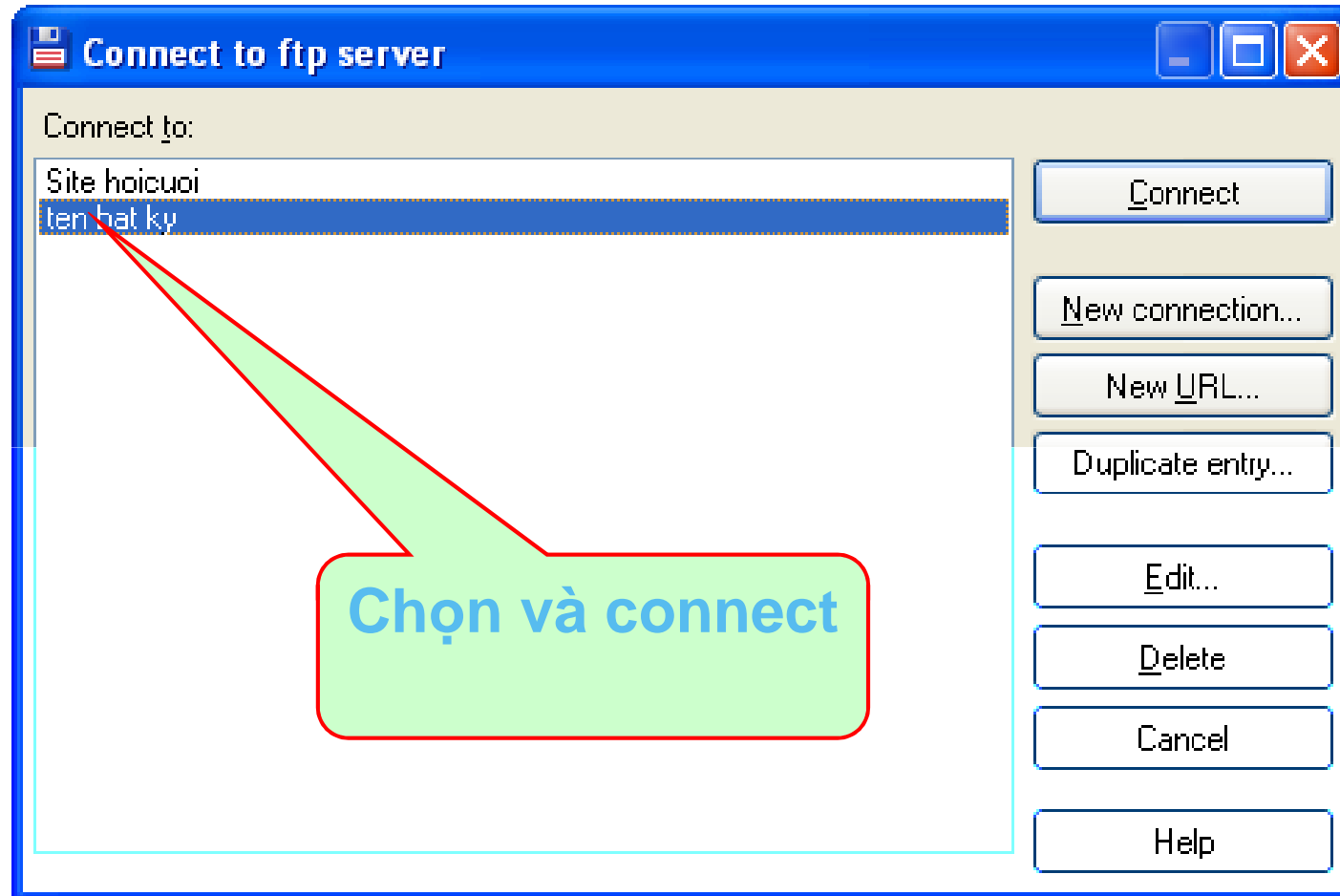
Send command to keep connection alive:  
Command: NOOP Send interval: every 90 s

Remember contents of all visited directories (in cache)

OK Cancel Help

- **Session:** Điền vào 1 cái tên tùy ý
- **Hostname, username, password** điền thông tin của bạn

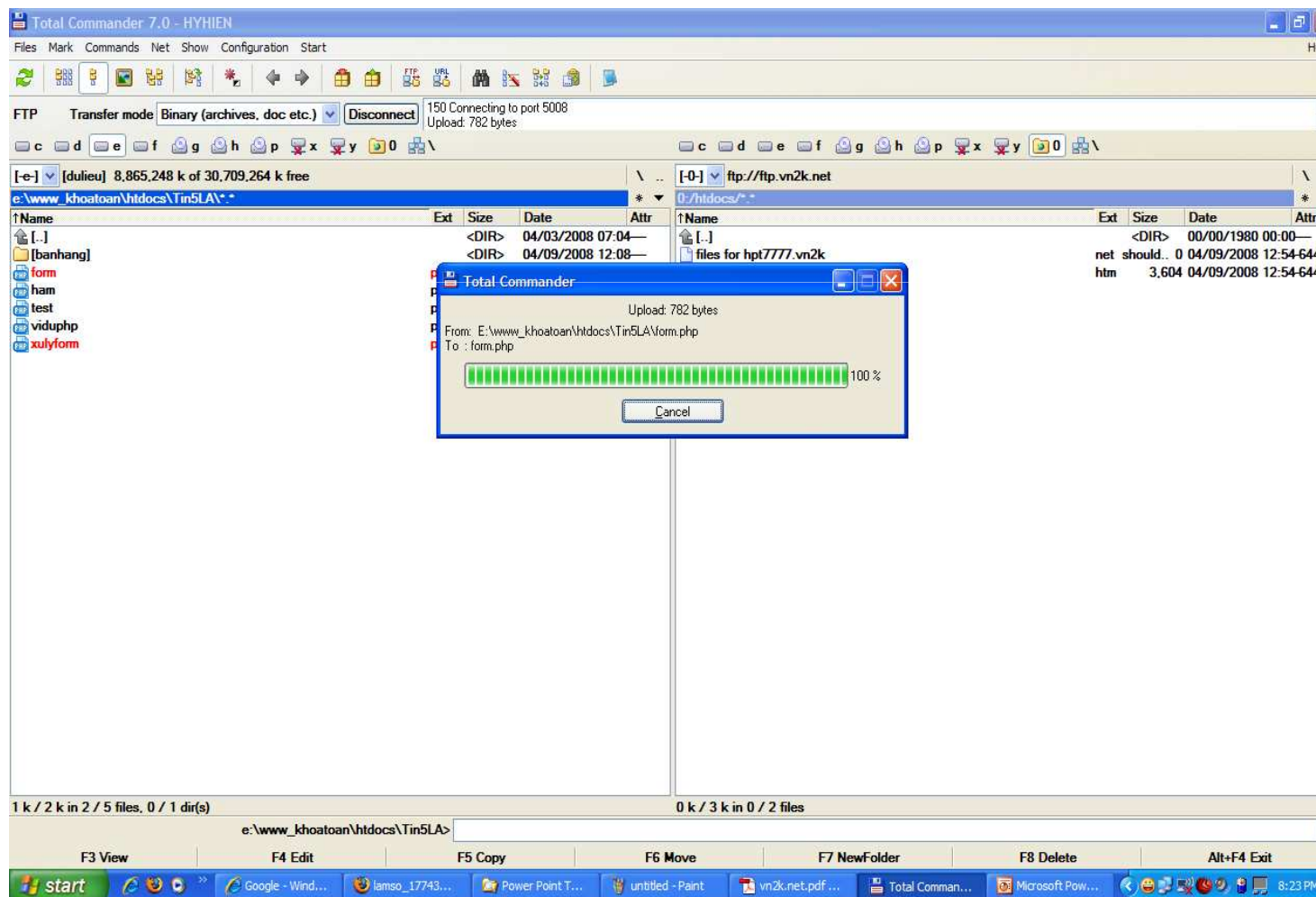
# Upload thông tin lên host dùng Total Commander



# Upload thông tin lên host dùng Total Commander



- Thực hiện chọn file và **copy** (F5)



# Tạo và sử dụng CSDL



- **Chọn MySQL Database trong cửa sổ CPanel**

You are currently using 0 of 7 databases.

Take a new database:

demo

Create Database

Remove a database:



Remove Database

Your current databases:

Database

Name: lamso\_1774398\_demo

[Backup] (Admin)

**Import Database**

**Cửa sổ PHPMyAdmin**

# Bộ đếm bằng đọc file



<?

```
$CountFile = "Counter.log";  
$CF = fopen ($CountFile, "r");  
$Hits = fread ($CF, filesize ($CountFile) );  
fclose ($CF);  
$Hits++;  
//Hiển thị ra màn hình  
$CF = fopen ($CountFile, "w");  
fwrite ($CF, $Hits);  
fclose ($CF);  
echo ($Hits);
```

?>

# Câu hỏi và thảo luận



# PHP The Second Day

**Đào Quang Minh**

**Trung tâm Tính toán Hiệu năng cao – ĐHBK Hà Nội**

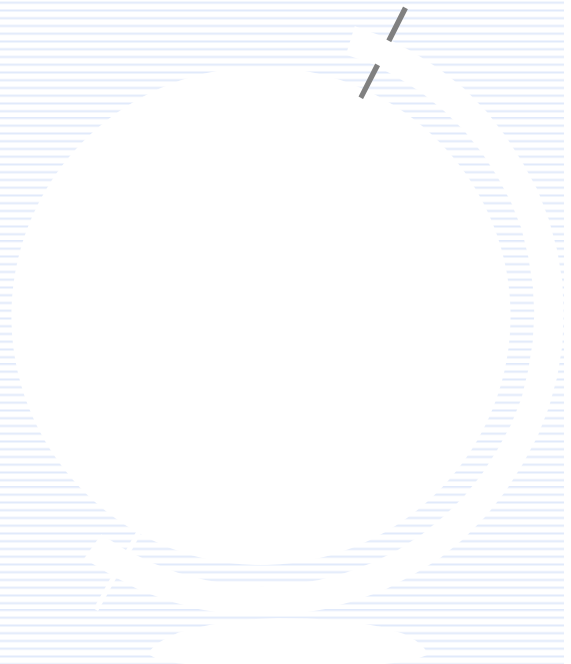
**minh0302@gmail.com**

**<http://hpcc.hut.edu.vn/~minhdq>**

## Chương 2: Làm quen với PHP (tiếp)

---

- ✓ Lập trình hướng đối tượng trong PHP
- ✓ Các phương thức truyền dữ liệu trong PHP
- ✓ Các khái niệm nâng cao





The background features a large, faint white circle centered on a light blue background with horizontal lines. Two orange lines, one horizontal and one vertical, intersect at the center of the circle, forming a crosshair. The text is positioned in the center of the circle.

# Lập trình hướng đối tượng trong PHP

# Khai báo và thể hiện lớp trong PHP

---

- ✓ Các lớp được khai báo thông qua từ khóa class, các thuộc tính khai báo dưới dạng các biến còn các phương thức được khai báo dưới dạng các hàm

```
class tên_lớp {  
    //danh_sách_các biến, hằng, lớp  
    //danh_sách_các_hàm  
}
```



(tiếp)

- ✓ Khởi tạo một đối tượng thuộc một lớp với từ khóa ***new***

```
$tên_đối_tượng = new tên_lớp();
```

- ✓ Để gọi hàm trong lớp ta sử dụng toán tử ->

```
$tên_đối_tượng->tên_hàm();
```

- ✓ Hoặc dùng toán tử :: để gọi đến một hàm hay một biến (biến phải được khai báo *static*)

```
$tên_đối_tượng::tên_hàm();
```

```
$tên_đối_tượng::$tên_biến;
```

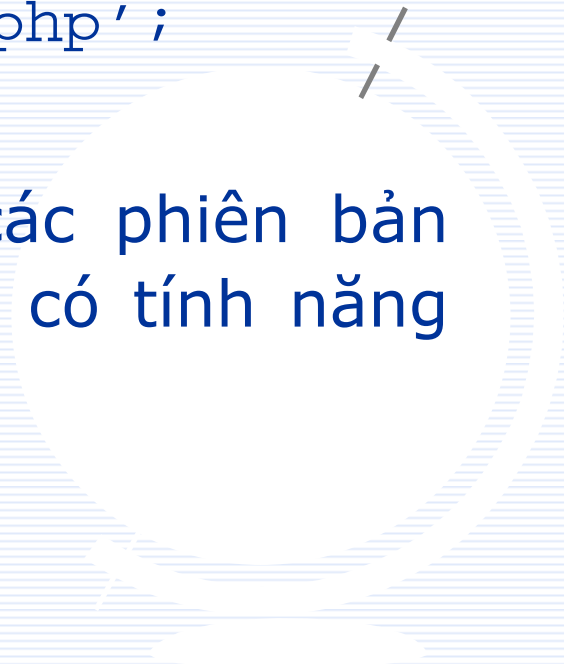


# Hàm tự động gọi `__autoload`

- ✓ Trong trường hợp chúng ta sử dụng một lớp chưa được định nghĩa thì hàm ***autoload*** sẽ tự động được gọi.

```
function __autoload($class_name)
{
    require_once $class_name.'.php';
}
```

- ✓ Điều này là chưa từng có trong các phiên bản PHP trước đây, chỉ từ PHP 5.x mới có tính năng này.



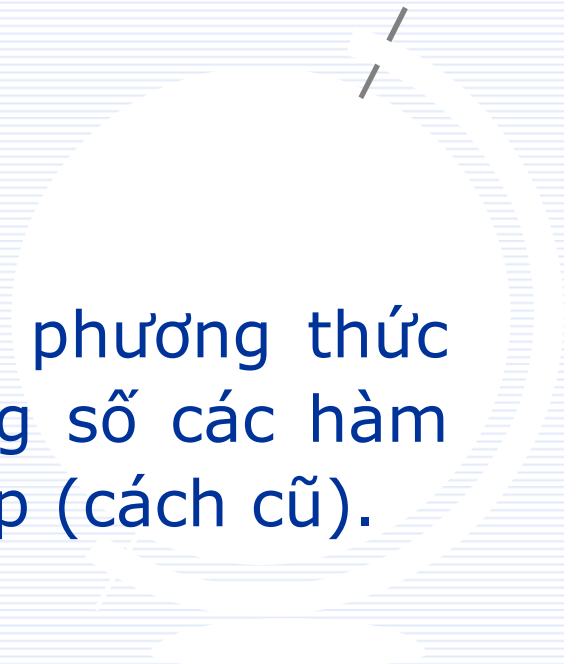
# Hàm tạo \_\_construct

---

- ✓ Cho phép bạn khai báo hàm tạo theo phương thức mới

```
class A
{
    function __construct() {
        //dòng lệnh
    }
}
```

- ✓ Nếu không tìm thấy hàm tạo theo phương thức mới này, PHP sẽ tìm hàm tạo trong số các hàm của lớp với tên hàm trùng với tên lớp (cách cũ).

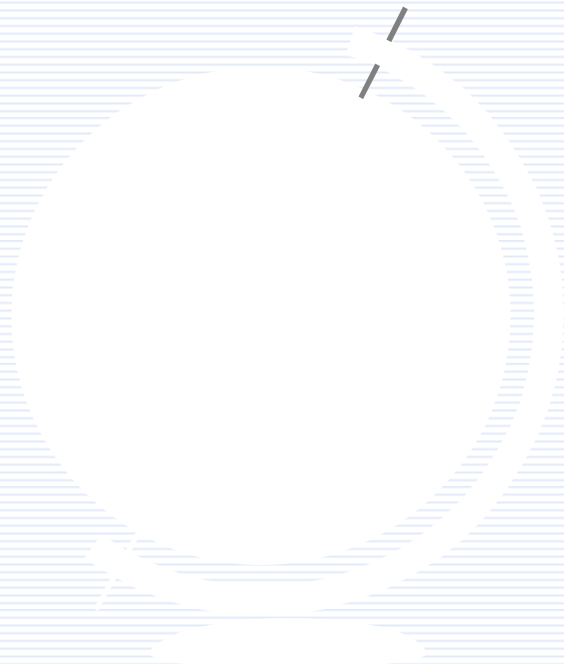


# Hàm hủy `__destruct`

- ✓ Ngược lại với hàm tạo, hàm hủy được triệu gọi khi đối tượng bị hủy
- ✓ Được gọi trong hai tình huống
  - Trong lúc thực thi mã lệnh, khi mà tất cả các liên quan đến đối tượng bị không còn nữa.
  - Hoặc khi đến cuối cùng của mã lệnh và PHP kết thúc các truy vấn.

```
function __destruct()  
{  
}  
}
```

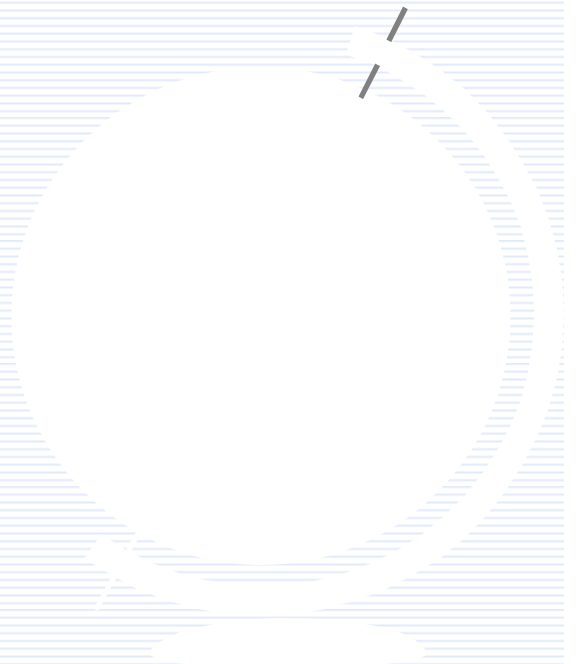
- ✓ `$tên_đối_tượng = NULL;`



# Truy cập phương thức và thuộc tính sử dụng con trỏ \$this

- ✓ Public, protected, private với các thuộc tính
  - Public: có thể truy xuất từ bên ngoài hoặc từ bên trong phương thức thông qua con trỏ đặc biệt \$this

```
class A {  
    public $a = "public_member";  
    function printHello()  
    {  
        print "Xin chao";  
    }  
    function test()  
    {  
        print $this->a;  
        print $this->printHello();  
    }  
}
```



# (tiếp)

- Protected: chỉ có thể truy xuất từ bên trong phương thức của lớp đó hoặc từ một lớp kế thừa từ lớp đó.

```
class A {
    protected $a = "protected_member";
    function printHello()
    {
        print $this->a;
    }
}
class B extends class A {
    protected $a = "new protected_member";
    function subprintHello()
    {
        print $this->a;
    }
}
```

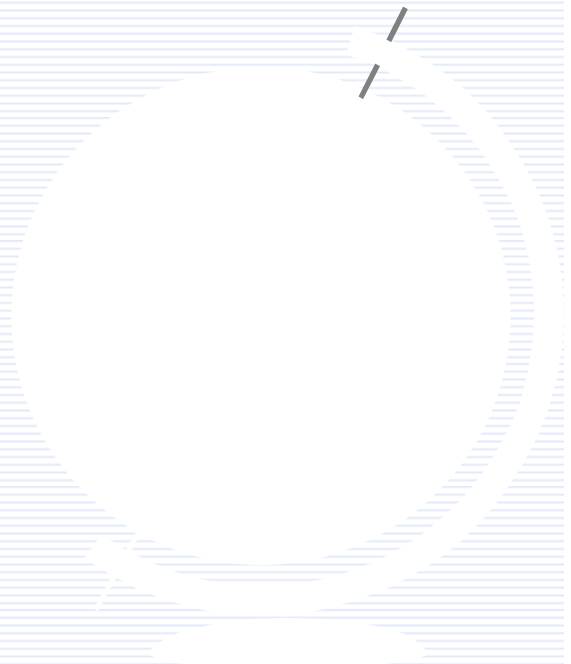




# (tiếp)

- Private: chỉ có thể được truy xuất từ bên trong lớp đó, cũng sử dụng con trỏ \$this như protected

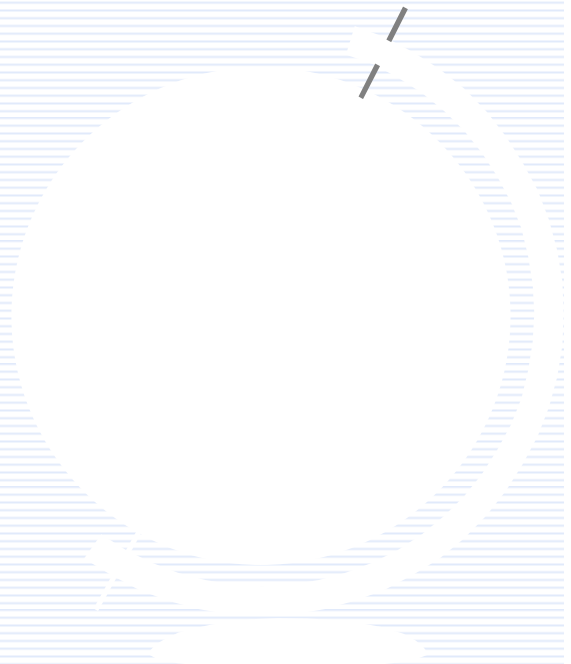
```
class A {  
    private $a = "private_member";  
    function printHello()  
    {  
        print $this->a;  
    }  
}
```



(tiếp)

- ✓ Public, protected, private với các phương thức
  - Cũng tương tự như với các thuộc tính với các định nghĩa và ràng buộc cho các kiểu public, protected và private
  - Cách khai báo cũng tương tự như khai báo các thuộc tính

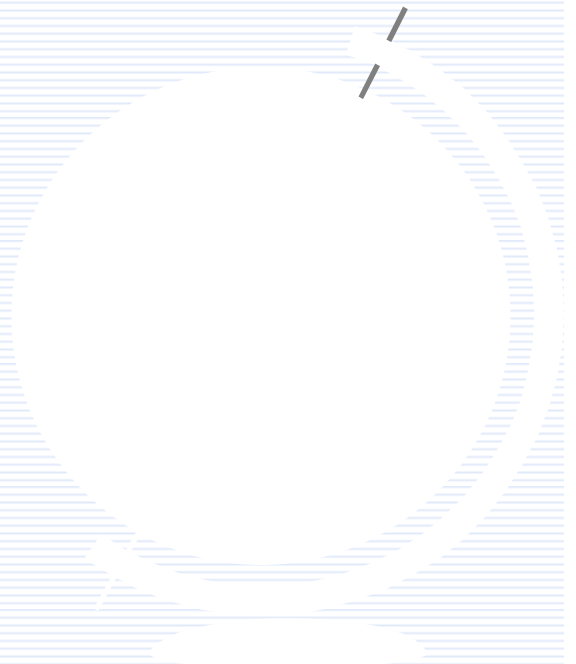
```
class A {  
    public function printHello()  
    {  
    }  
}
```



# Static

- ✓ Biến static có thể coi như một biến toàn cục được đặt trong 1 lớp nhưng có thể truy xuất từ bất kỳ đâu qua lớp đó, sử dụng toán tử **::**
- ✓ Khi gọi biến static trong hàm ta sử dụng toán tử ***self*** thay cho con trỏ ***this***

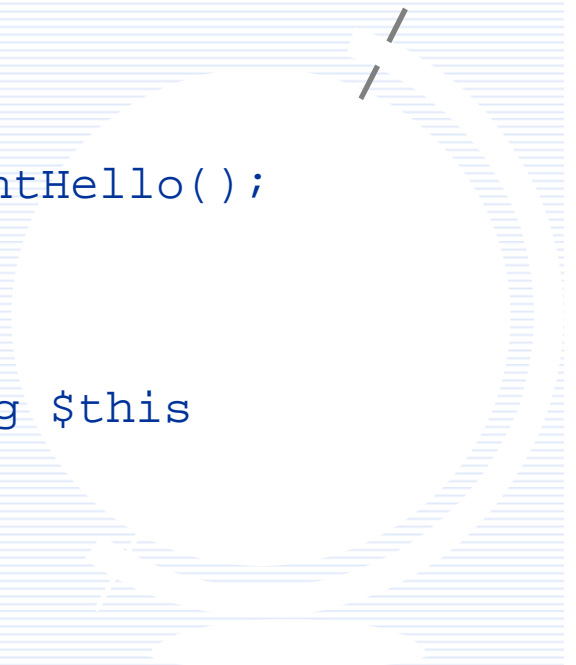
```
class A {  
    static $a = 5;  
    function printHello(){  
        print self::$a;  
    }  
}  
  
print A::$a++; //6  
A::printHello(); //5
```



(tiếp)

- ✓ Phương thức static: tương tự như biến static nhưng ngoài việc gọi nó bằng toán tử **self** nó có thể được truy xuất bằng con trỏ **this** trong 1 hàm

```
class A {  
    static function printHello() {  
        print "Xin chào";  
    }  
    function show () {  
        self::printHello();//hoặc $this->printHello();  
    }  
}  
A::printHello();//A::show(); sẽ lỗi khi dùng $this  
$obj = new A();  
$obj->show();//hoặc $obj->printHello();
```



(tiếp)

- ✓ Chú ý để tránh sự không rõ ràng giữa toán tử **this** và **self** hay **->** và **::** người ta thường quy ước như sau:
  - Khi gọi đến thuộc tính hay đối tượng kiểu static trong một lớp hay từ ngoài lớp gọi đến 1 hàm hay 1 thuộc tính của lớp nên dùng toán tử **self** và **::**
  - Khi gọi đến thuộc tính hay đối tượng bình thường trong một lớp hay từ ngoài lớp gọi đến 1 hàm hay 1 thuộc tính của lớp thì dùng toán tử **this** và **->**

# Hằng số trong lớp

---

- ✓ Có thể định nghĩa hằng trên một lớp
- ✓ Nó có giá trị không đổi và khai báo không có \$
- ✓ Tên hằng không được trùng với tên biến, lớp, hàm hay kết quả của 1 phép toán hay 1 hàm

```
const myConst = "Const_Value";
```

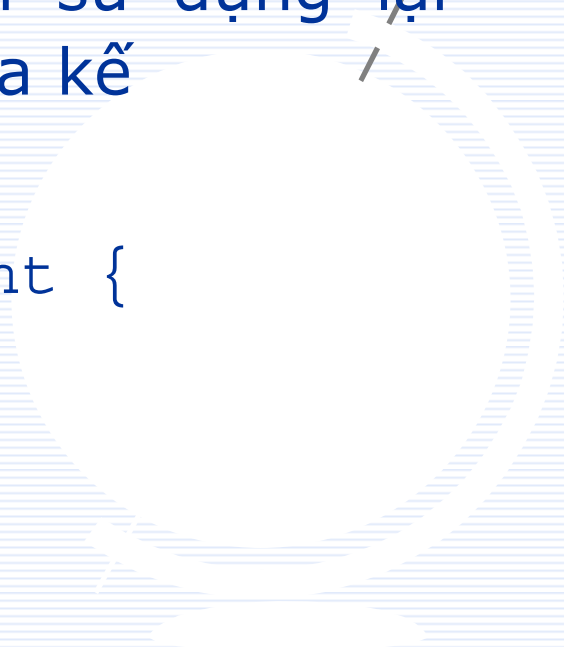
- ✓ Để lấy giá trị của 1 hằng ta chỉ có thể dùng toán tử `::` hoặc thông qua một phương thức



# Đa hình

- ✓ Đa hình trong PHP sử dụng các lớp và thừa kế giữa chúng, làm cho việc miêu tả các tình huống trong cuộc sống thực được dễ dàng hơn.
- ✓ Nó cũng làm cho dễ dàng hơn với lập trình viên khi phát triển 1 dự án bằng cách sử dụng lại những mã lệnh chính thông qua thừa kế

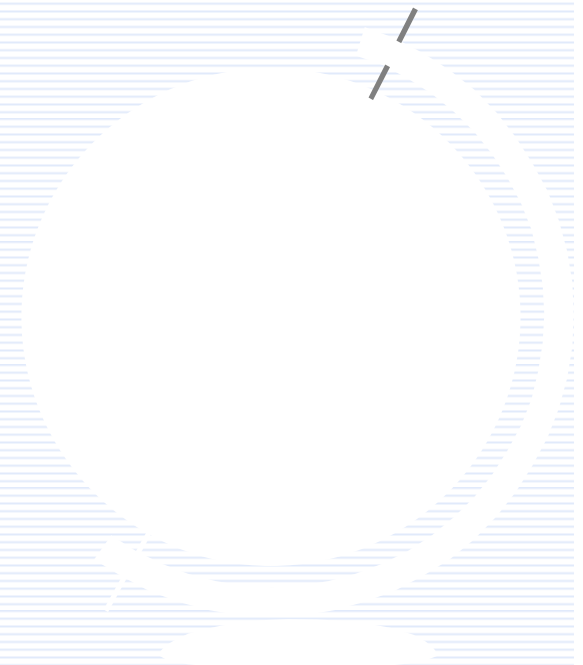
```
class child extends class parent {  
}
```



(tiếp)

---

- ✓ PHP không hỗ trợ đa thừa kế như C++
- ✓ PHP cung cấp 1 giải pháp khác cho việc tạo ra nhiều hơn 1 quan hệ của lớp bằng cách sử dụng ***interfaces*** như là Java

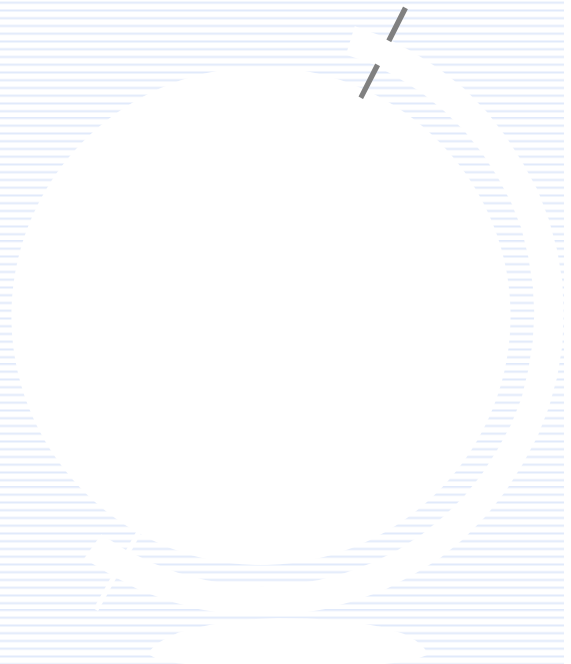




## parent:: và self::

---

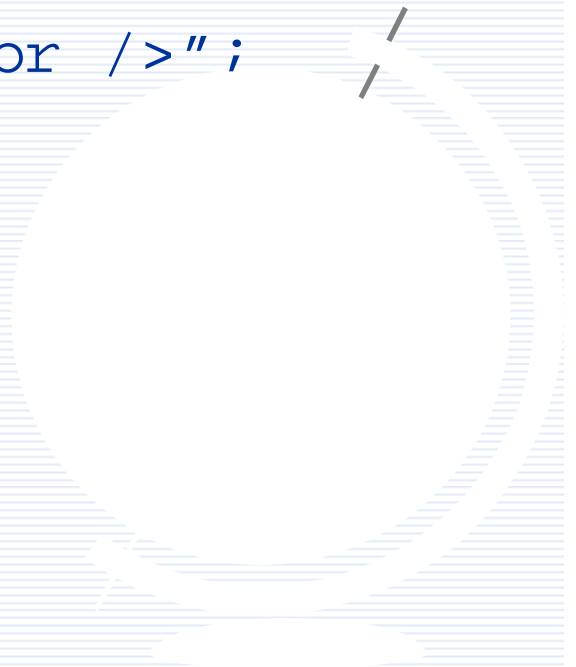
- ✓ PHP hỗ trợ 2 lớp dành riêng để dễ dàng hơn trong việc viết các ứng dụng hướng đối tượng
  - **Self::** thường được dùng để truy cập đến các đối tượng, phương thức kiểu static hay hằng số của lớp hiện tại
  - **Parent::** gọi đến lớp cha và thường được dùng khi muốn gọi hàm tạo hay các phương thức của lớp cha



# Toán tử instanceof

- ✓ Có giá trị trả về kiểu logic (true hay false)

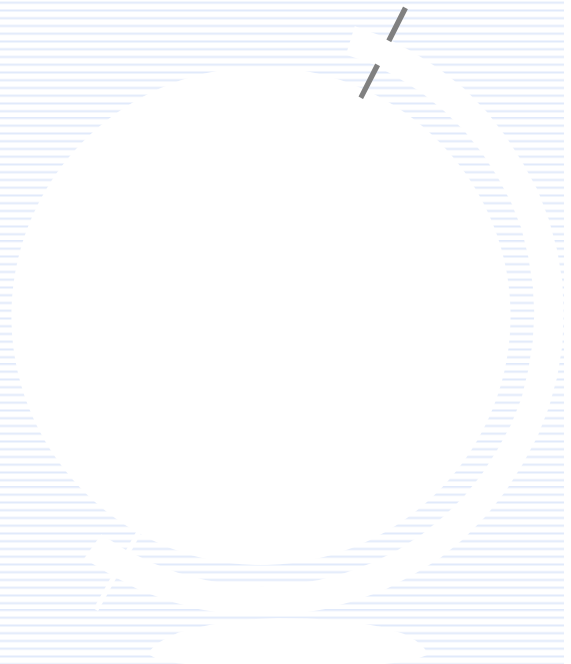
```
function checkifClass ($shape)
{
    if ($shape instanceof Rectangle) {
        print $shape->name;
        print " is a Class A <br />";
    }
}
checkifClass(new Square());
```



(tiếp)

---

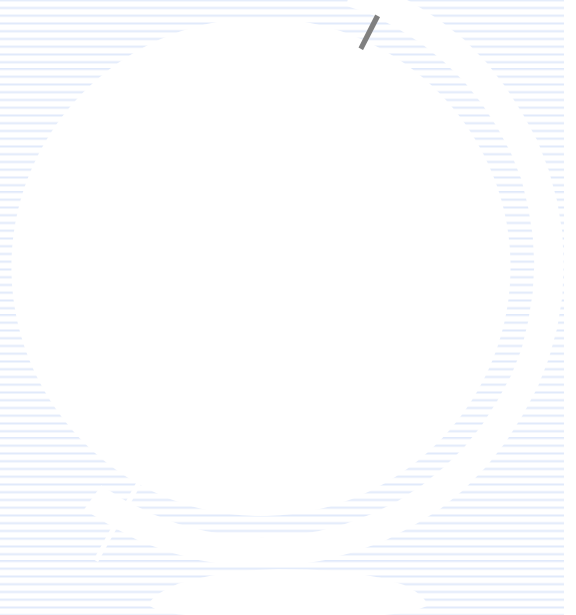
- ✓ Dùng để kiểm tra xem một đối tượng có phải thuộc về một lớp nào đó hay không?
- ✓ Toán tử này cũng dùng để kiểm tra xem một đối tượng có **implement** một **interfaces** hay không?



# Interfaces

- ✓ Là một lớp cho phép chỉ định phương thức nào của lớp đó được thực thi mà không phải định nghĩa phương thức đó làm việc như thế nào
- ✓ Nếu một lớp được khai báo là ***implements*** từ lớp ***interfaces*** mà không có các phương thức như lớp ***interfaces*** đã khai báo thì chương trình sẽ báo lỗi

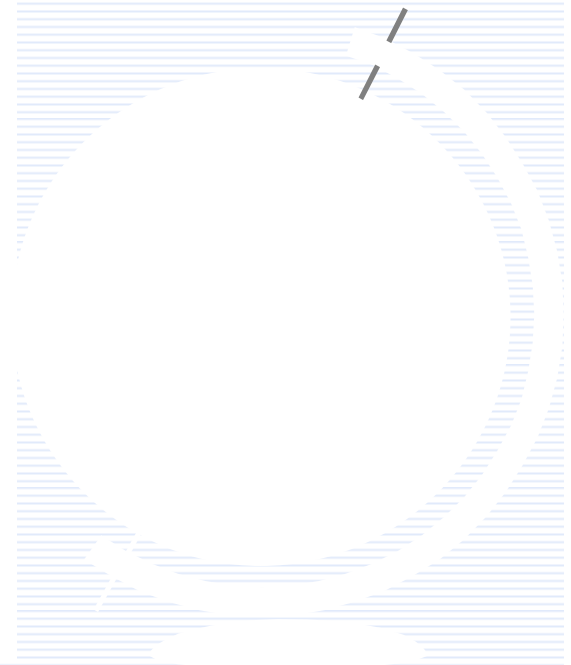
```
interface A {  
    //các_phương_thức (*)  
}  
  
class MyClass implements A {  
    //các_phương_thức (**)
```



(tiếp)

- ✓ Một lớp có thể thực thi nhiều hơn một **interfaces** nhưng phải đảm bảo các phương thức trong các **interfaces** đó không được trùng tên nhau

```
<?php
interface interf1{
    public function funcA();
    public function funcB();
}
interface interf2{
    public function funcC();
    public function funcD();
}
class myClass implements interf1,interf2
{
    public function funcA(){
        echo 'A';
    }
    public function funcB(){
        echo 'B';
    }
    public function func(){
        echo 'All';
    }
    public function funcC(){}
    public function funcD(){}
}
$a = new myClass;
$a->funcA();
?>
```

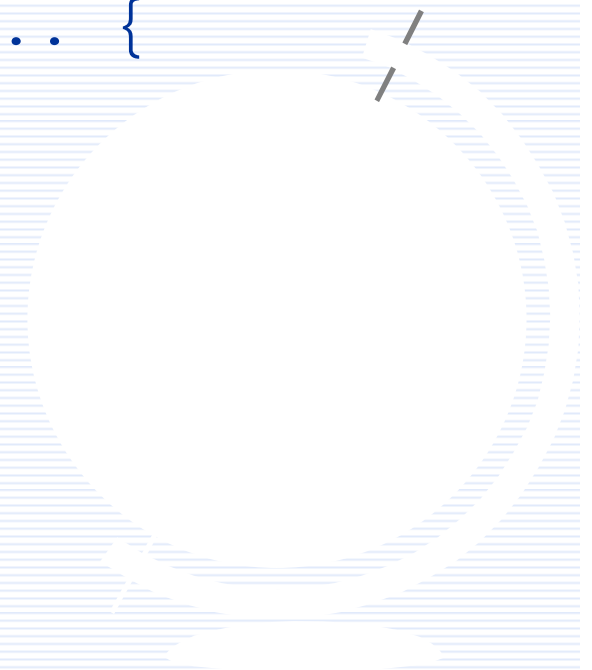


(tiếp)

✓ Mọi phương thức mô tả trong ***Interfaces*** phải được mô tả public

✓ ***Interfaces*** có thể được thừa kế từ các ***interfaces*** khác (cho phép đa thừa kế)

```
interface T1 extends T2, T3, ... {  
}
```



# Final

- ✓ **Final Method**: cũng giống như Java, **Final Method** trong PHP được dùng để chắc chắn là phương thức đó không thể được implement lại trong các lớp khác xuất phát từ nó.

```
class MyBaseClass {
    final function idGenerator()
    {
        return $this->id++;
    }

    protected $id = 0;
}

class MyConcreteClass extends MyBaseClass {
    function idGenerator()
    {
        return $this->id += 2;
    }
}
```

(tiếp)

- ✓ **Final Class**: tương tự như **Final Method**, một lớp đã được khai báo là **Final** thì mọi lớp extend từ nó sẽ không được thực hiện

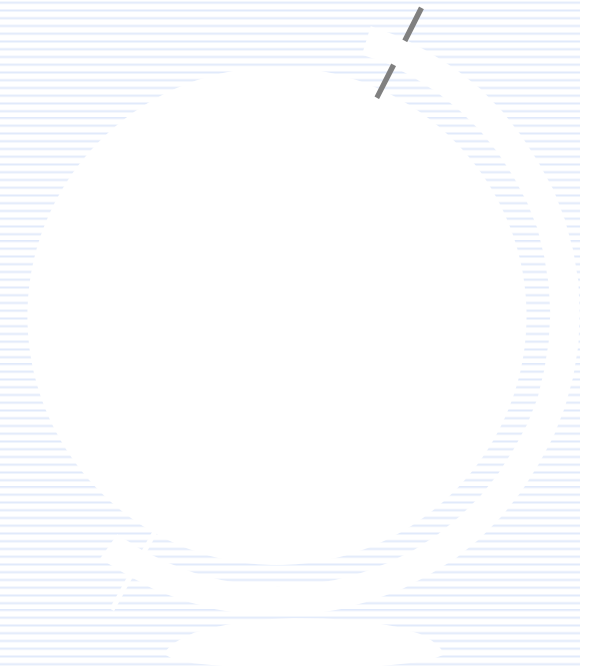
```
final class MyBaseClass {  
    ...  
}  
  
class MyConcreteClass extends MyBaseClass {  
    ...  
}
```



# Overloading

---

- ✓ Trong PHP chúng ta có thể định nghĩa lại các hàm trong lớp cha bằng các hàm cùng tên trong lớp con với chức năng hoàn toàn khác nhau
- ✓ Có thể tham chiếu chính xác các phương thức hay phương thức đã bị **overload** bằng cách dùng **parent** và toán tử tham chiếu **::**



# Exception Handling

```
try {  
    ... // Code which might throw an exception  
} catch (FirstExceptionClass $exception) {  
    ... // Code which handles this exception  
} catch (SecondExceptionClass $exception) {  
}
```

- ✓ Các mã lệnh được bắt exception trong khối try {...}
- ✓ Khi exception được ném vào catch nó sẽ được bắt lần lượt qua các khối catch () {...}
- ✓ Khi một catch đã được bắt thì những catch ở sau sẽ bị bỏ qua

(tiếp)

---

- ✓ Throw <object>: chỉ có thể ném ra một object chứ không phải bất kỳ một loại dữ liệu cơ bản nào như strings hay integers
- ✓ Có một lớp exception được định nghĩa trước đã tồn tại gọi là **Exception**, tất cả các lớp đều phải thừa kế từ nó.
- ✓ Việc ném một đối tượng mà không được thừa kế từ lớp này sẽ gây ra lỗi.



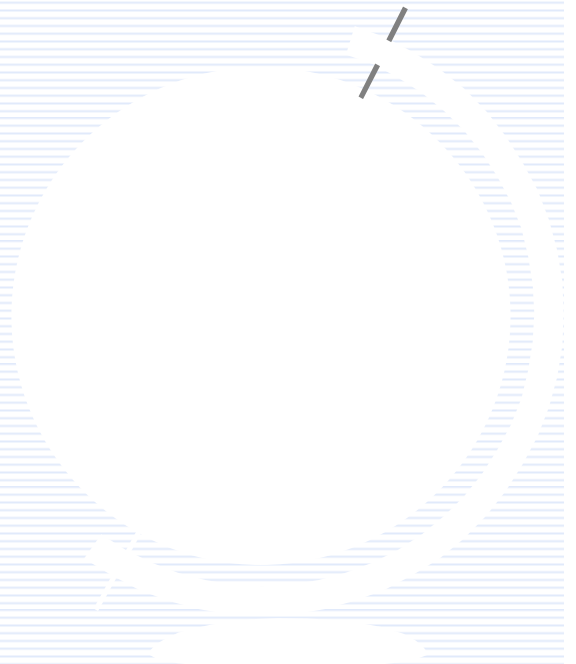


# **Các phương thức truyền dữ liệu trong PHP**

# Các đối tượng nhập dữ liệu

---

- ✓ Cho phép người sử dụng nhập dữ liệu trên trang web. Dữ liệu này có thể được gửi về server để xử lý.
- ✓ Người sử dụng nhập dữ liệu thông qua các điều khiển (controls). Có nhiều loại control:
  1. Form
  2. Oneline Textbox
  3. Checkbox
  4. Radio Button
  5. Button
  6. Combo box (drop-down menu)
  7. Listbox
  8. Hộp nhập văn bản nhiều dòng (TextArea)
  9. ...



# Phương thức truyền dữ liệu

- ✓ PHP xử lý dữ liệu qua khái niệm **form** của HTML
- ✓ **Form** có hai thuộc tính quan trọng:
  - **action="..."**: Địa chỉ trang web nhận dữ liệu khi form được **submit** (thường là địa chỉ tương đối)
  - **method="..."**; Có hai phương thức là Get (mặc định) và Post

```
<form name="" action="index.php" method = "post">  
<input type="text" name="user">  
<input type="submit" name="" submit value="submit">  
</form>
```

(tiếp)

✓ Truyền dữ liệu (biến) qua URL:

- Các biến được truyền thành từng cặp `biến=giá_trị` phân cách bởi dấu `&`
- Phân cách với địa chỉ trang ban đầu bởi dấu hỏi chấm (?)

Ví dụ: Truyền 3 biến `a`, `b`, `c` có giá trị lần lượt là `1`, `2`, `-3` vào trang <http://localhost/index.php> qua URL:

<http://localhost/ptb2.php?a=1&b=2&c=-3>

(tiếp)

- ✓ Khi ta Submit 1 form sử dụng phương thức GET, dữ liệu được truyền qua URL:
  - Tên các biến là tên đối tượng trên form
  - Giá trị biến là giá trị NSD nhập vào đối tượng
  - Để đọc giá trị các biến trong PHP: Sử dụng mảng **\$\_GET**, các chỉ số là tên biến.

Ví dụ: Trong trang index.php ở trên đọc các biến a, b, c:

```
$a = $_GET[ "a" ] ;
```

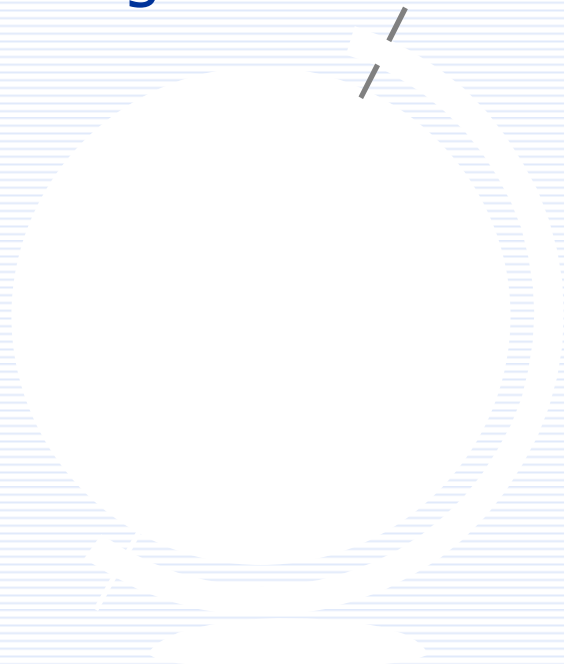
```
$b = $_GET[ "b" ] ;
```

```
$c = $_GET[ "c" ] ;
```



(tiếp)

- ✓ Truyền dữ liệu qua phương thức Post
  - Dữ liệu của Form post được truyền trong thân của yêu cầu HTTP
  - Để đọc dữ liệu sử dụng mảng **`$_POST`** với các chỉ số là tên của đối tượng trên form gửi đến.



The background features a large, faint white circle centered on a light blue background with horizontal lines. Two orange lines, one horizontal and one vertical, intersect at the center of the circle, forming a crosshair. The text is centered within this circle.

# **Các khái niệm nâng cao**

# Cookie

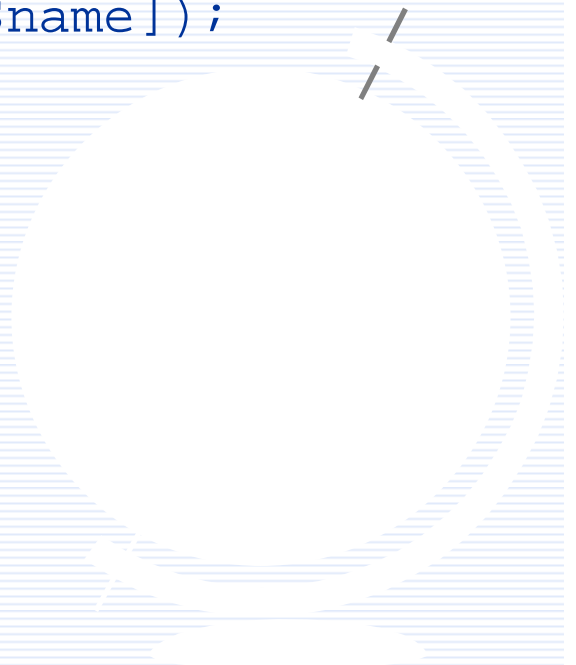
- ✓ Là một đoạn dữ liệu được truyền đến browser từ server
- ✓ Được gửi ngược lại server mỗi khi browser tải website đó từ server
- ✓ Hàm tạo Cookie:

```
function create_cookie ($name,$value) {  
    $expires = time() + 60*60*24*365;  
    setcookie($name,$value,$expires)  
}
```

(tiếp)

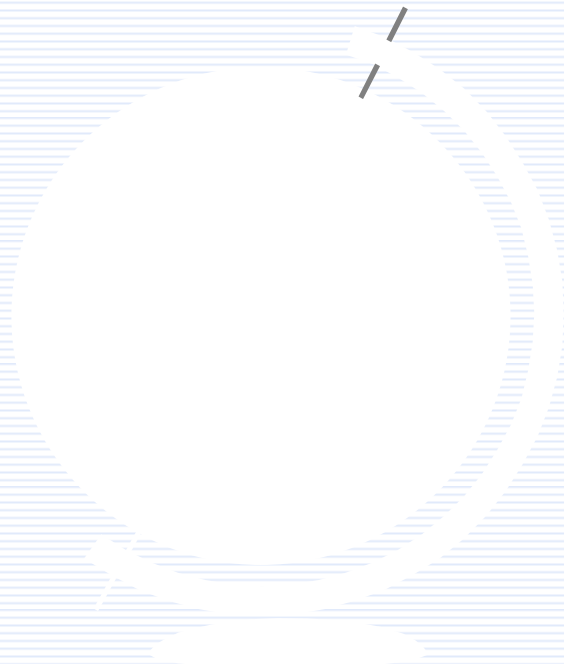
## ✓ Hàm lấy Cookie

```
function get_cookie ($name)
{
    if (isset($_COOKIE[$name]))
    {
        return urldecode($_COOKIE[$name]);
    }
    else
    {
        return FALSE;
    }
}
```



(tiếp)

- ✓ Xóa một cookie cũng tương tự như tạo một cookie nhưng thay giá trị khởi tạo bằng 0  
`create_cookie('tên_cookie', 0);`
- ✓ Cookie cũng tự động xóa sau khoảng thời gian đã được quy định bởi biến `$expires`



# Session

---

- ✓ Là khoảng thời gian người dùng giao tiếp với ứng dụng
- ✓ Bắt đầu từ khi người sử dụng truy cập vào ứng dụng lần đầu tiên cho đến khi thoát khỏi ứng dụng

```
session_start(); //phải đặt ở trên  
cùng tất cả các dòng lệnh
```



(tiếp)

✓ Khởi tạo giá trị cho session

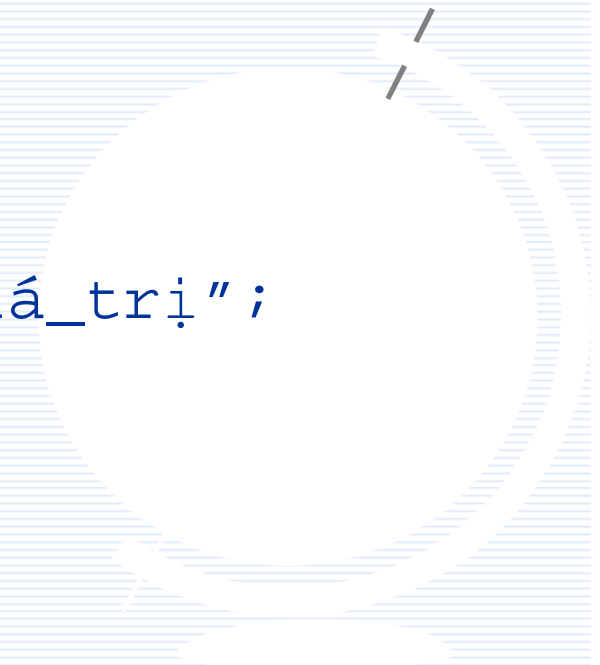
```
$_SESSION["tên_session"] = "giá_trị";
```

✓ Mỗi session có một định danh (ID) duy nhất được tạo ra trên server khi được khởi tạo. Để lấy giá trị hiện tại của session\_id

```
$id = session_id();
```

✓ Khởi tạo giá trị cho session

```
$_SESSION["tên_session"] = "giá_trị";
```



(tiếp)

---

✓ Hủy một session

```
session_unset("tên_session");
```

✓ Hủy mọi session

```
session_destroy();
```



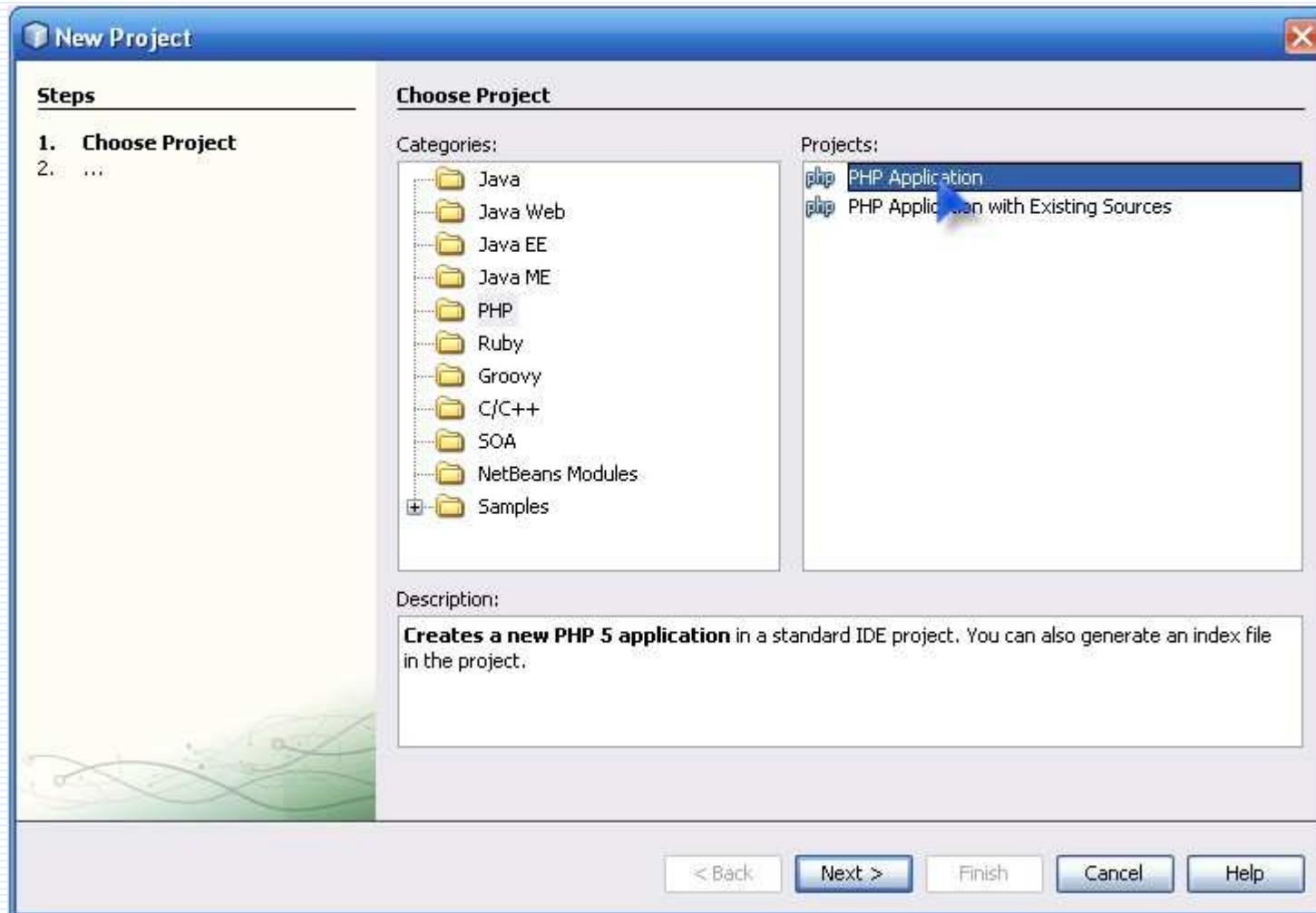


The background features a large, faint white circle centered on a light blue background with horizontal lines. Two orange lines, one horizontal and one vertical, intersect at the center of the circle, forming a crosshair. The text is positioned in the center of the circle.

# **Thiết kế Web với Netbean và Xampp**

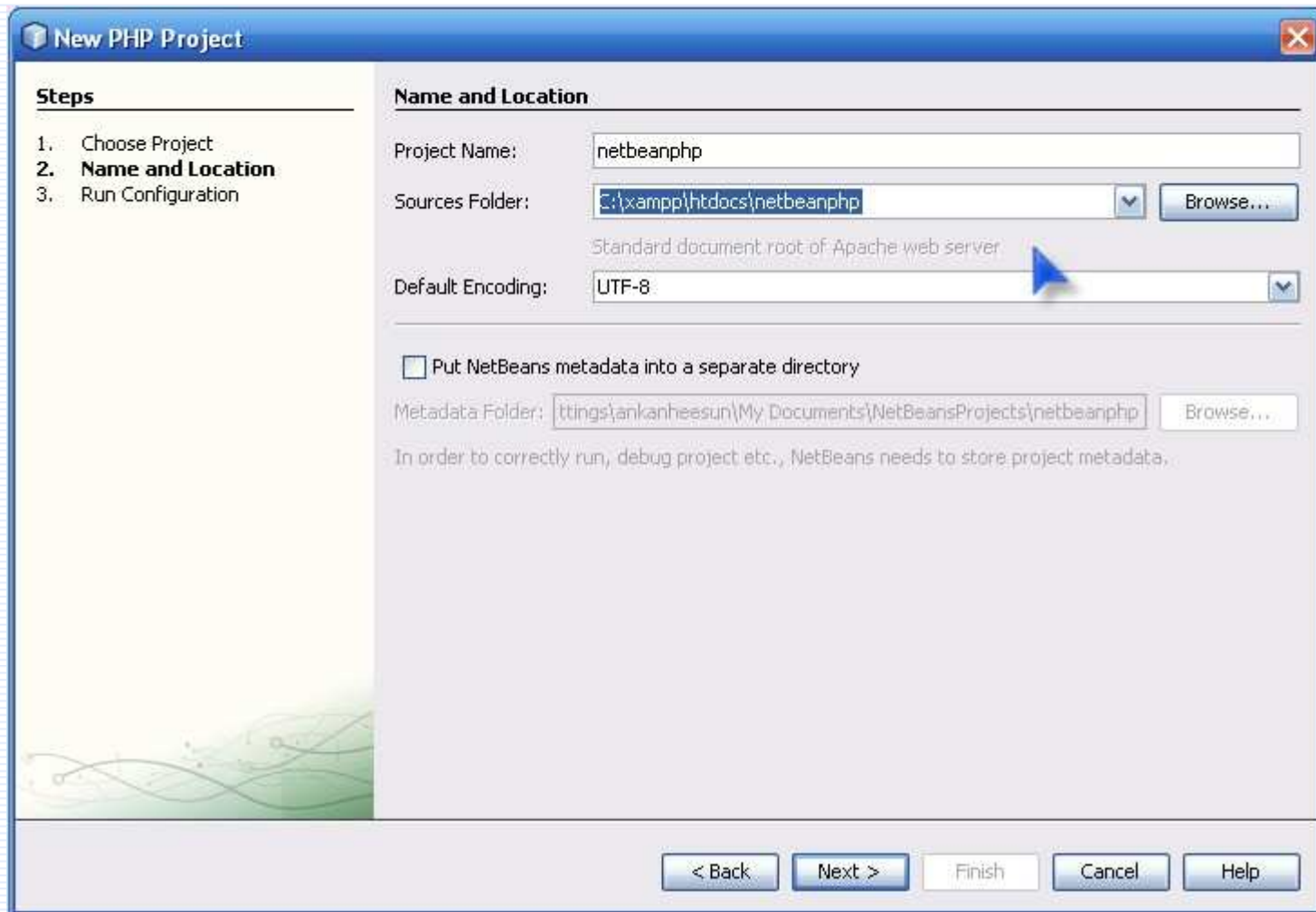
# PHP Project

- ✓ Tạo một project mới



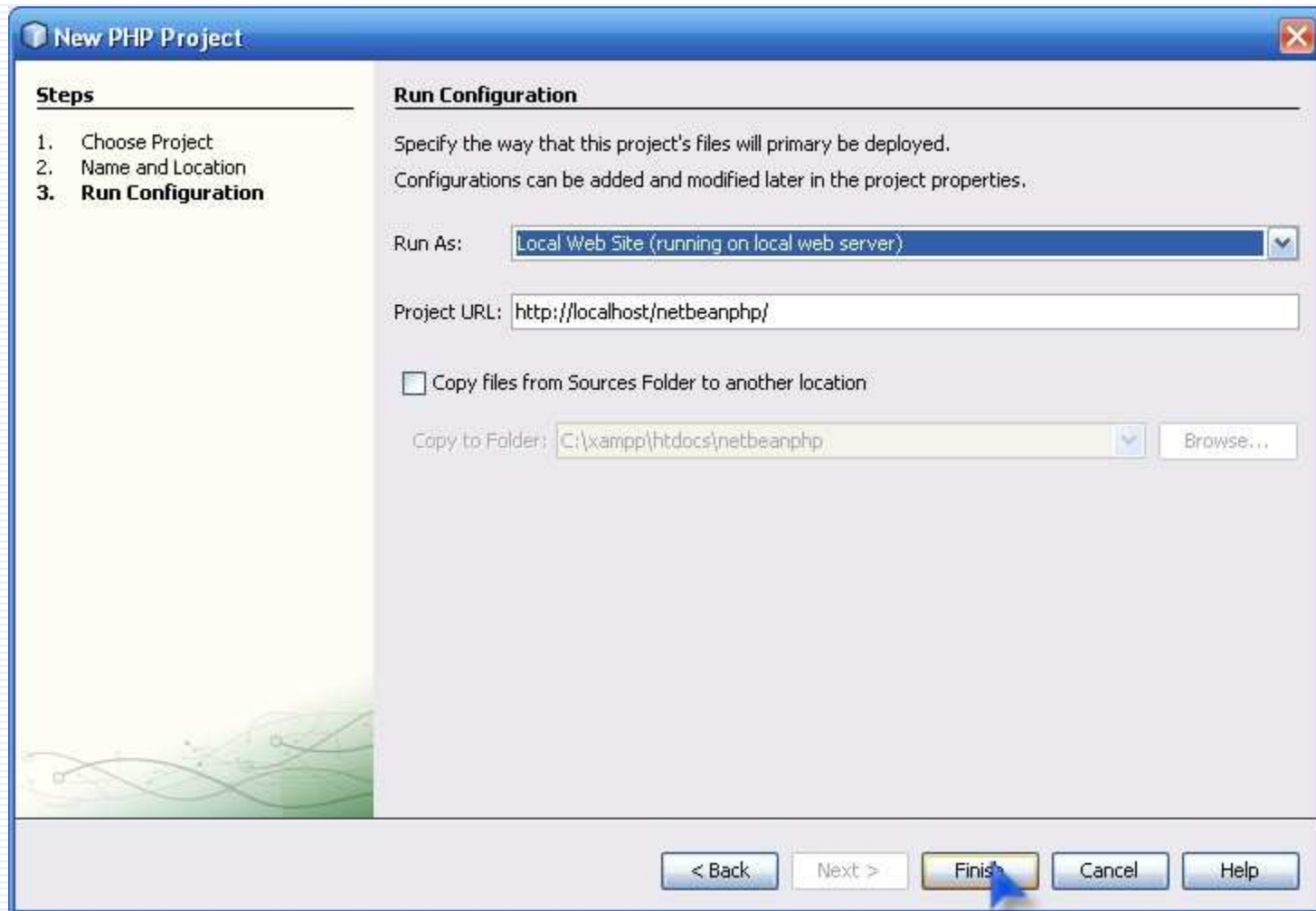
# Sources Folder

✓ ***`/opt/lampp/htdocs/coursek50/`***



# Configuration

## ✓ *Run as: Local Web Site*





**THE END**

# PHP The Third Day

**Đào Quang Minh**

**Trung tâm Tính toán Hiệu năng cao – ĐHBK Hà Nội**

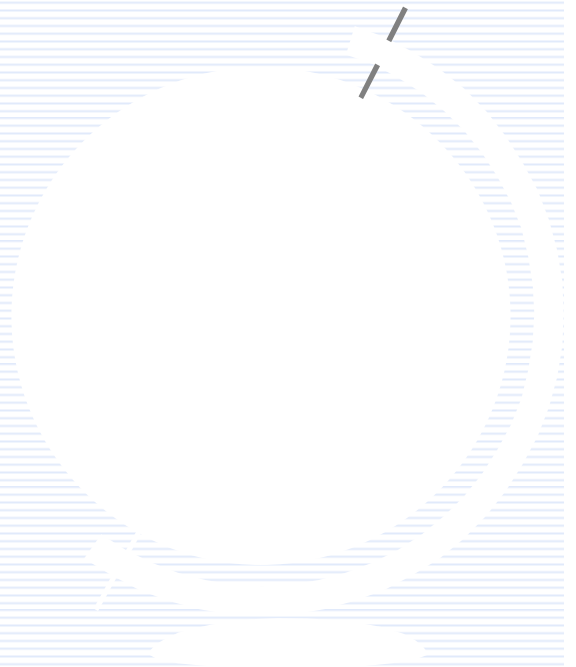
**minh0302@gmail.com**

**<http://hpcc.hut.edu.vn/~minhdq>**

# Chương 3: PHP và HTML, CSS, JavaScript

---

- ✓ PHP và HTML
- ✓ HTML và CSS (tạo giao diện)
- ✓ HTML và JavaScript



The background features a large, semi-transparent white circle centered on a light blue background with horizontal lines. Two thin orange lines intersect at the center of the circle, forming a cross. The text "PHP và HTML" is positioned to the left of the circle's center, between the two orange lines.

# PHP và HTML



# HTML Form

---

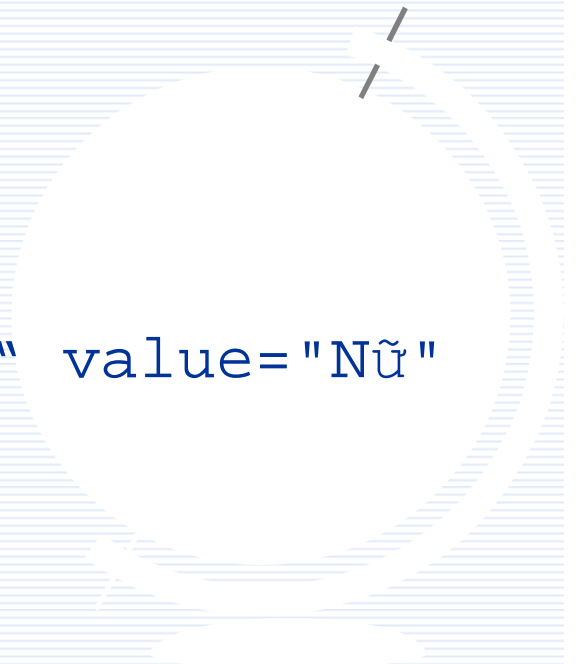
```
<form name="myform" method="post"
  action="process.php">
```

```
<p>Name <input type="text" name="txtname"
  /></p>
```

```
<p>Job <input type="text" name="txtjob"
  /></p>
```

```
<p>Giới tính <input type="radio"
name="rdbsex" value="Nam" />Nam
```

```
<input type="radio" name="rdbsex" value="Nữ"
tabindex="4" />Nữ </p>
```



(tiếp)

```
<p>Quê quán</p>
```

```
<select name="selhomeland">
```

```
  <option value="Hà Nội"
```

```
    selected="selected">Hà Nội</option>
```

```
  <option value="Huế">Huế</option>
```

```
  <option value="Hồ Chí Minh">Hồ Chí  
  Minh</option>
```

```
</select>
```



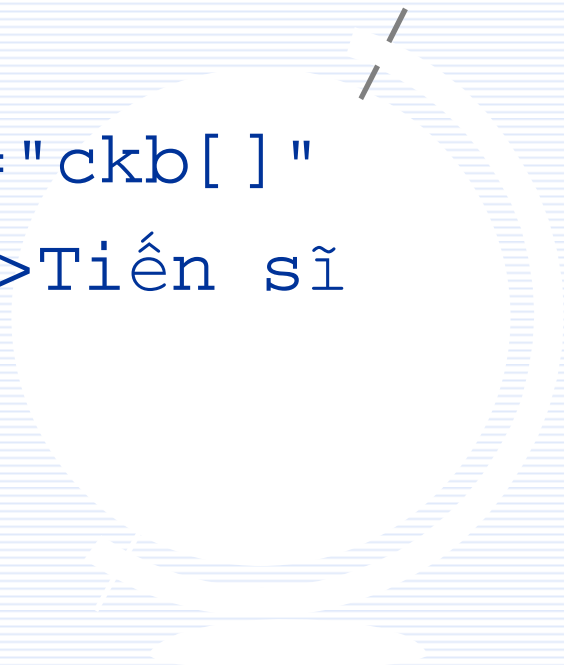
(tiếp)

```
<p>Học vị</p>
```

```
<input type="checkbox" name="ckb[]"  
  value="Thạc sĩ" />Thạc sĩ
```

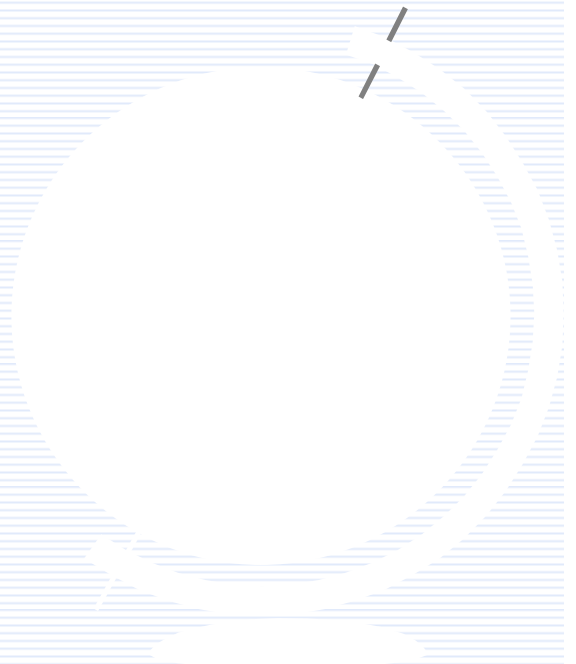
```
<input type="checkbox" name="ckb[]"  
  value="Tiến sĩ" />Tiến sĩ
```

```
<input type="checkbox" name="ckb[]"  
  value="Tiến sĩ khoa học" />Tiến sĩ  
khoa học
```



(tiếp)

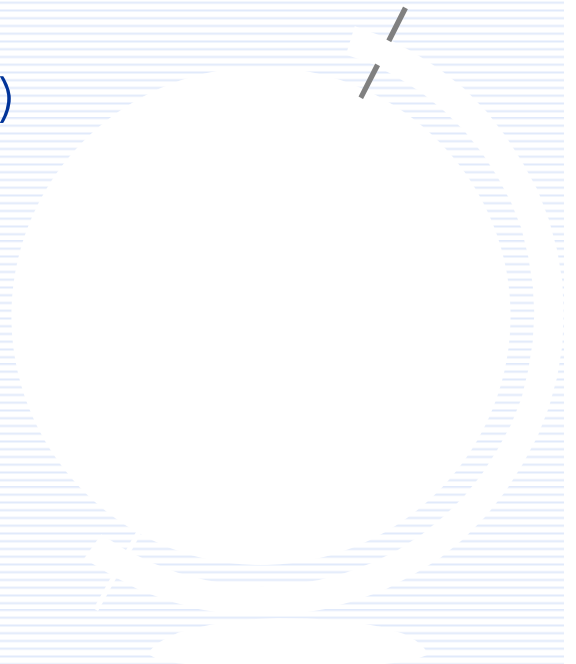
```
<p>Mô tả thêm về bản thân</p>  
<textarea name="txtdetail" cols="45"  
  rows="5" tabindex="9"></textarea>  
<p><input type="submit" value="Cập  
  nhật" tabindex="10" /></p>  
</form>
```



# PHP và HTML Form

---

```
$name = $_POST["txtname"];  
$job = $_POST["txtjob"];  
$sex = $_POST["rdbsex"];  
$homeland = $_POST["selhomeland"];  
$level = "";  
foreach ($_POST["ckb"] as $value)  
{  
    $level .= $value."<br />";  
}  
$detail = $_POST["txtdetail"];
```



# HTML File Browse

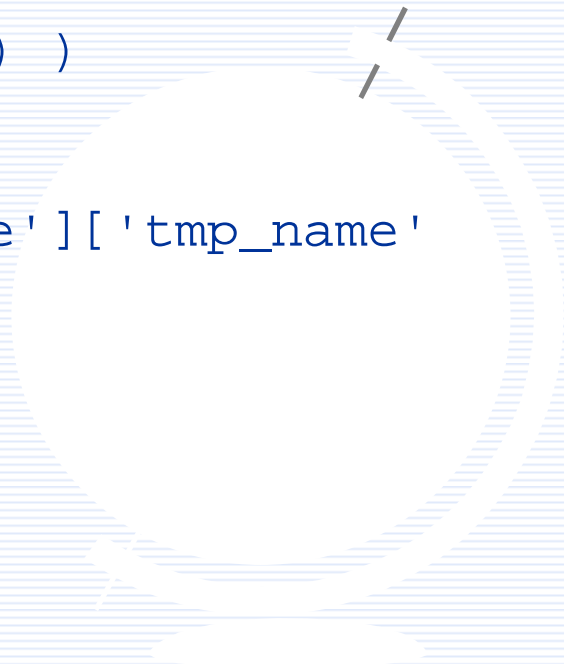
---

```
<form action="upload.php" method="post"
  enctype="multipart/form-data">
File <input name="myfile" type="file"
  size="30" />
<input type="submit" name="submitBtn" />
</form>
```



# upload.php

```
<?php
$destination_path = getcwd().DIRECTORY_SEPARATOR;
$target_path = $destination_path . basename(
    $_FILES['myfile']['name']);
if (($_FILES["myfile"]["type"] == "image/gif") ||
    ($_FILES["myfile"]["type"] == "image/jpeg") &&
    ($_FILES["myfile"]["size"] < 5120000) )
{
    if(move_uploaded_file($_FILES['myfile']['tmp_name']
        ], $target_path))
        $result = 1;
}
?>
```



(tiếp)

- ✓ Ngoài ra PHP còn có thể chèn vào bất kỳ vị trí nào trong file HTML

```
<?php
```

```
if (điều kiện)
```

```
{
```

```
?>
```

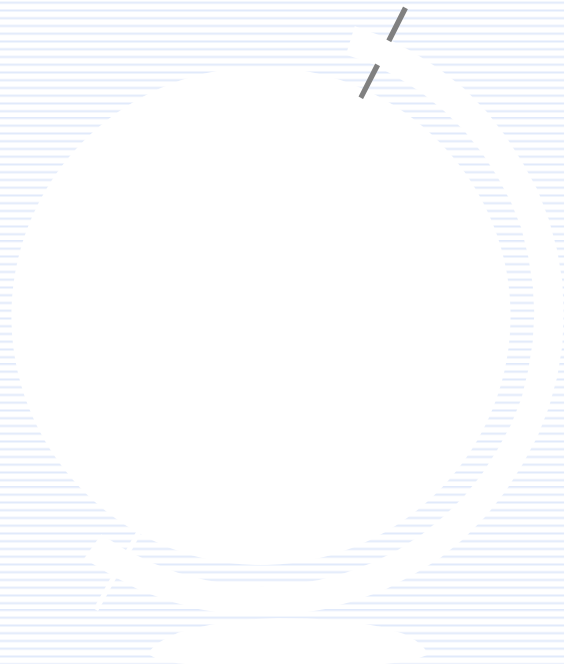
```
<form name="" action="" method="">
```

```
</form>
```

```
<?php
```

```
}
```

```
?>
```





The background features a large, semi-transparent white circle centered on a light blue background with horizontal lines. Two thin orange lines intersect at the center of the circle, forming a cross. The text "HTML và CSS" is positioned to the left of the circle's center, between the two orange lines.

# HTML và CSS

# HTML

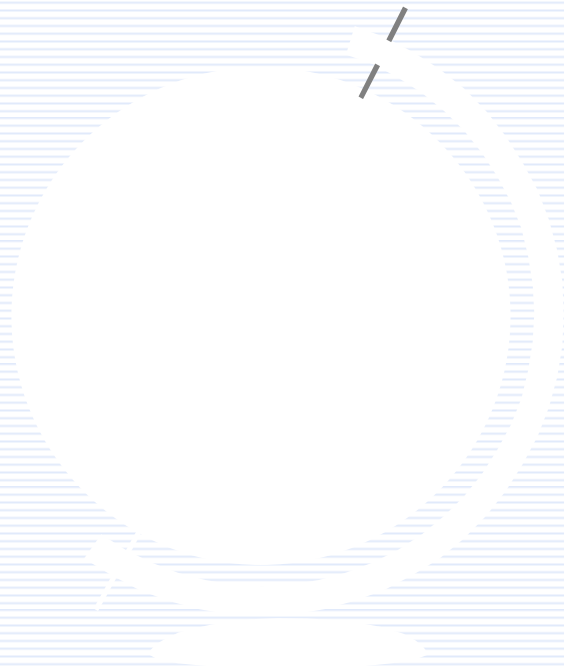
- ✓ HTML – Hyper Text Markup Language sử dụng các thẻ (tag) <..> để định dạng dữ liệu
- ✓ HTML không phân biệt chữ hoa hay chữ thường và không được trình duyệt báo lỗi cú pháp
- ✓ Cấu trúc cơ bản của 1 file html thường có dạng:

```
<html>
```

```
<head>...</head>
```

```
<body>...</body>
```

```
</html>
```



# Thẻ (tag)

- ✓ Có nhiều thẻ, mỗi thẻ có 1 tên và mang ý nghĩa khác nhau.
  - ✓ Có 2 loại thẻ: thẻ đóng và thẻ mở
  - ✓ Cách viết thẻ:
    - Thẻ mở: **<tên\_thẻ>**  
Ví dụ: <u>, <p>, <img>...
    - Thẻ đóng tương ứng: **</tên\_thẻ>**  
Ví dụ: </u>, </p>
- Chú ý: luôn có thẻ mở nhưng có thể không có thẻ đóng tương ứng. Ví dụ: <img> không có thẻ đóng



# Thuộc tính của thẻ

✓ Một thẻ có thể có các thuộc tính nhằm bổ sung tác dụng cho thẻ

✓ Mỗi thuộc tính có tên thuộc tính (tên\_TT)

✓ Viết thẻ có thuộc tính:

**<tên\_thẻ tên\_TT1="giá\_trị1" tên\_TT2="giá\_trị2"...>**

✓ Chú ý:

- Có thể thay đổi thứ tự, số lượng các thuộc tính mà không gây ra lỗi cú pháp
- Sự hỗ trợ các thẻ, thuộc tính ở mỗi trình duyệt là khác nhau. Chỉ giống nhau ở các thẻ, thuộc tính cơ bản.
- Thẻ đóng của thẻ có thuộc tính vẫn **viết bình thường** (</tên\_thẻ>)

# Soạn thảo văn bản

- ✓ Văn bản được soạn thảo như bình thường trong các file HTML
- ✓ Lưu ý:
  - Mọi khoảng trống, dấu xuống dòng trong HTML được thể hiện trên trang web là 1 khoảng trống duy nhất
  - Để gõ một số ký tự đặc biệt ta phải sử dụng mã:
    - Khoảng trống (trong trường hợp muốn có nhiều hơn 1 ký tự trống): **&nbsp;**
    - Dấu nhỏ hơn (<) và lớn hơn (>): **&lt; &gt;**
    - Dấu ngoặc kép ("): **&quot;**
    - Ký hiệu ©: **&copy;**
    - ...
- ✓ Ghi chú trong HTML:  
**<!-- Ghi chú -->**



(tiếp)

---

- ✓ `<head></head>`: là phần đầu của 1 văn bản html, định nghĩa các thông tin về tài liệu, bao gồm các thẻ như `<title>`, `<meta>`, `<script>`, v.v...
- ✓ `<body></body>`: là phần thân của văn bản html, dữ liệu trong phần này sẽ hiển thị lên browser của người dùng

## Các thẻ thông dụng

---

- ✓ `<table></table>`: dùng để tạo bảng
- ✓ `<form></form>`: dùng để tạo form
- ✓ `<img>`: dùng để chèn ảnh
- ✓ `<a href = ""></a>`: dùng để chèn liên kết
- ✓ `<font>...</font>`



# CSS

---

- ✓ CSS - Cascading Style Sheet: Mẫu quy định cách thức thể hiện các thẻ HTML
- ✓ Các đặc điểm kỹ thuật của CSS được duy trì bởi W3C
- ✓ Giúp tiết kiệm được rất nhiều thời gian và công sức cho việc thiết kế web.
- ✓ Có thể định nghĩa nhiều style vào một thẻ HTML (Cascading)



# Các loại style

---

✓ Có 4 loại style:

- *Inline Style* (Style được qui định trong 1 thẻ HTML cụ thể)
- *Internal Style* (Style được qui định trong phần <HEAD> của 1 trang HTML )
- *External Style* (style được qui định trong file .CSS ngoài)
- *Browser Default* (thiết lập mặc định của trình duyệt)

✓ Thứ tự ưu tiên: Mức ưu tiên giảm dần từ trên xuống

# Cách chèn CSS

---

✓ Đặt trong `<head>...</head>`

✓ Với Internal style:

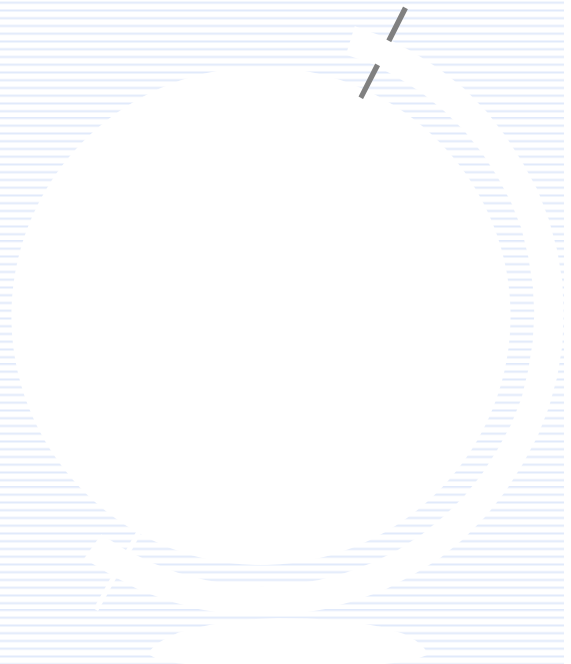
```
<style type="text/css">
```

```
<!--
```

```
    Nội dung định nghĩa style
```

```
-->
```

```
</style>
```



# Cách chèn CSS (tt)

---

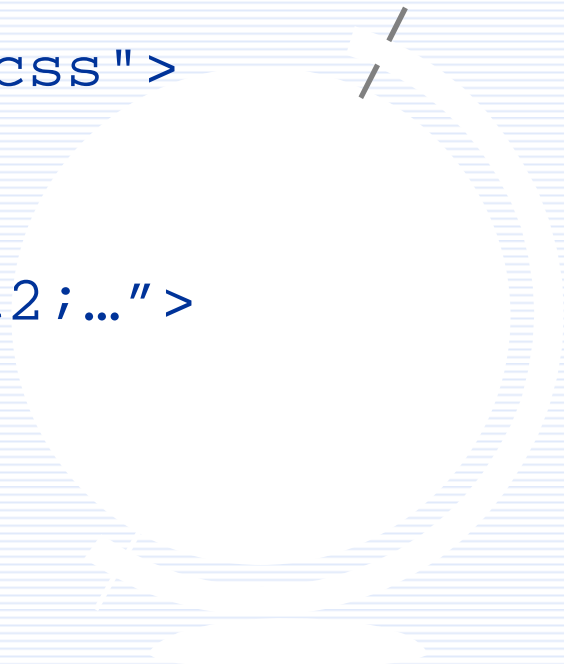
## ✓ Với External style:

- Định nghĩa style trong file riêng (thường có đuôi .CSS)
- Nhúng file CSS đã định nghĩa vào trang web:

```
<link href="địa chỉ file"  
rel="stylesheet" type="text/css">
```

## ✓ Với Inline style:

```
<tên_thẻ style="tt1:gt1;tt2:gt2;...">
```



## Chú ý

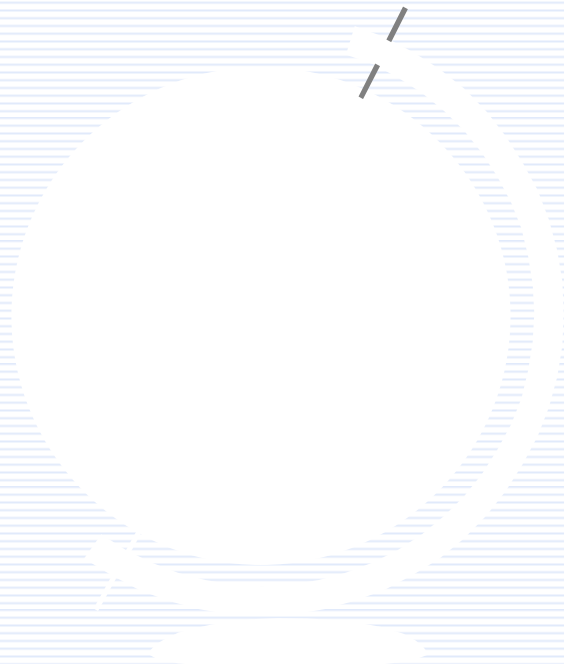
---

- ✓ CSS phân biệt chữ hoa chữ thường
- ✓ Để ghi chú trong style chúng ta sử dụng

```
/*
```

```
    đoạn ghi chú
```

```
*/
```



## Khai báo CSS

---

```
selector {  
    Property1: Value1;  
    Property2: Value2;  
}
```

# Áp dụng cho thẻ cụ thể

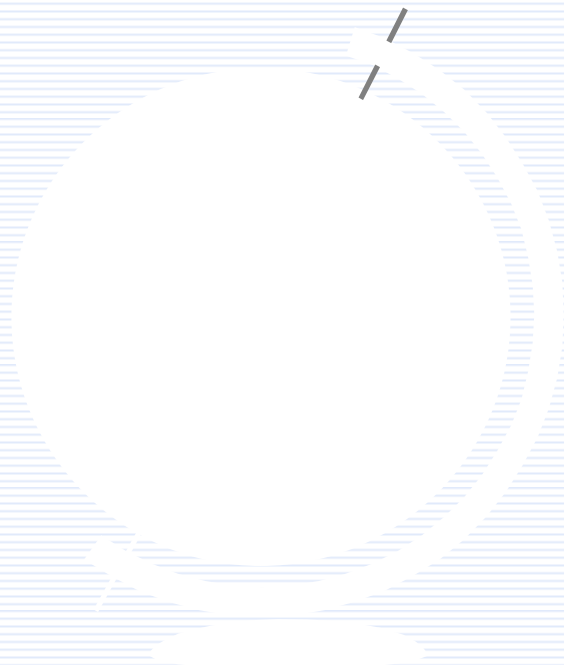
---

## ✓ Trường hợp 1 thẻ

```
p {  
    color: red;  
}
```

## ✓ Trường hợp nhiều thẻ

```
h1,h2,h3,h4,h5,h6 {  
    font-family:arial;  
}
```



# Tạo lớp

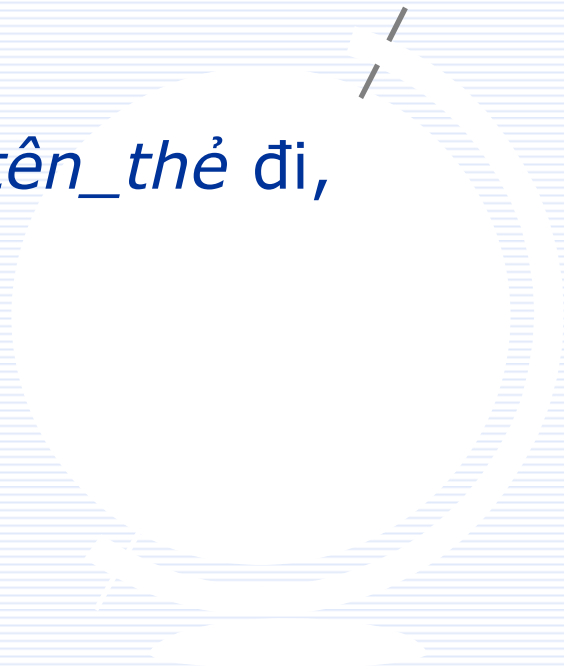
- ✓ Gắn với thẻ cụ thể: Đặt *selector* là `tên_thẻ.tên_lớp`

```
p.loai1{  
    color:red;  
}
```

```
p.loai2{  
    color:blue;  
}
```

- ✓ Không gắn với thẻ cụ thể: bỏ phần *tên\_thẻ* đi, giữ lại dấu chấm:

```
.loai3{  
    color:green;  
}
```



# Sử dụng lớp

✓ Đặt thuộc tính `class` của thẻ = "tên\_lớp"

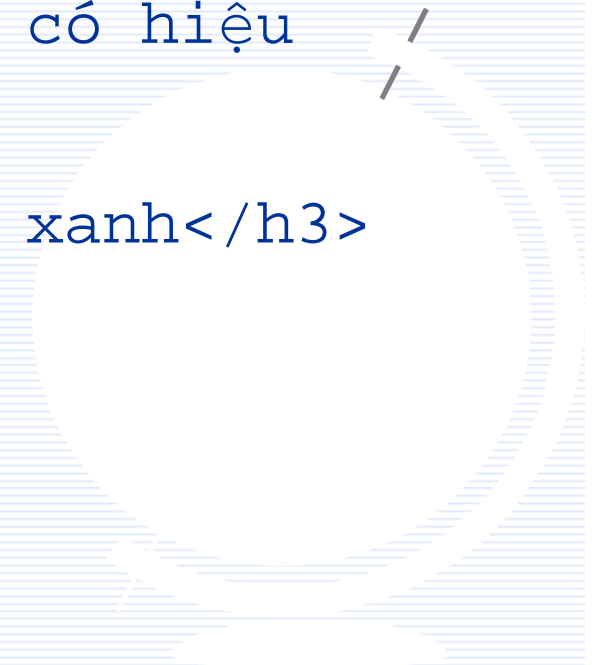
```
<tên_thẻ class="tên_lớp">
```

✓ Ví dụ:

```
<p class="loai1">Đoạn này màu đỏ</p>
```

```
<h1 class="loai2">Style không có hiệu  
lực</h1>
```

```
<h2 class="loai3">Tiêu đề màu xanh</h3>
```

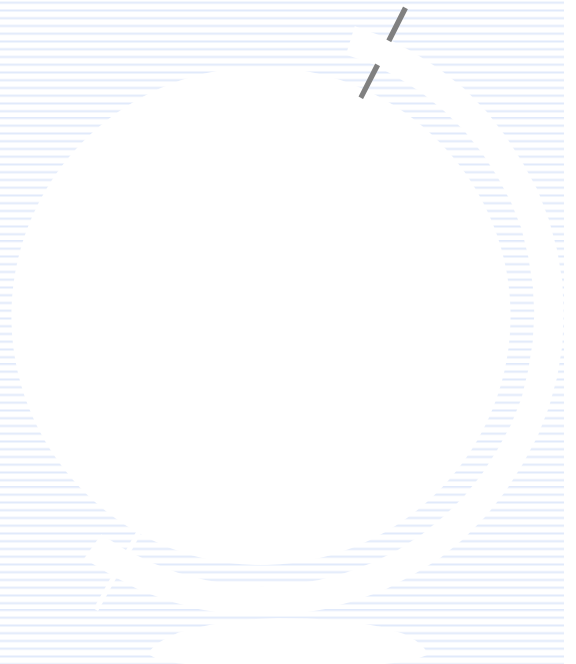




# Tạo định danh

- ✓ Tương tự như class. Thay dấu chấm (.) thành dấu thăng (#).
- ✓ Cho thẻ cụ thể: `tên_thẻ#định_danh{...}`
- ✓ Tổng quát: `#định_danh{...}`

```
p#doan1{
    color:red;
}
#loai2{
    color:blue;
}
```



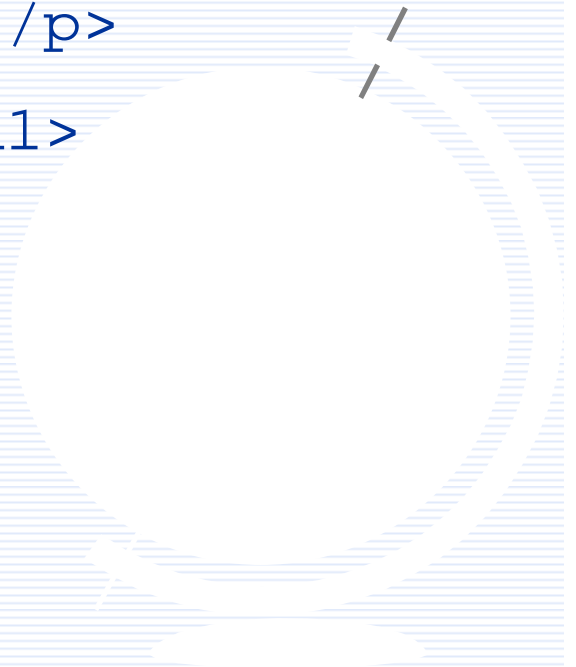
# Sử dụng định danh

---

- ✓ Mỗi định danh là duy nhất trên trang
- ✓ Đặt thuộc tính `id` của thẻ = `định_danh`  
`<tên_thẻ id="định_danh">`

```
<p id="doan1">Đoạn này màu đỏ</p>
```

```
<h1 id="loai2">Tiêu đề xanh</h1>
```



The background features a large, faint white circle centered on a light blue background with horizontal lines. Two orange lines intersect at the center: one horizontal line and one vertical line. The text "HTML và JavaScript" is positioned to the left of the center, between the horizontal orange lines.

# HTML và JavaScript

# JavaScript

---

- ✓ JS là ngôn ngữ kịch bản, được nhúng hoặc tích hợp vào tệp tin HTML
- ✓ Nếu được trình duyệt hỗ trợ, JS sẽ được thực thi
- ✓ JS được thông dịch trong trang HTML sau khi toàn bộ trang được load nhưng trước khi trang được hiển thị
- ✓ JavaScript được hãng Sun Microsystems phát triển từ Livescript của Netscape

# Chèn Javascript vào HTML

---

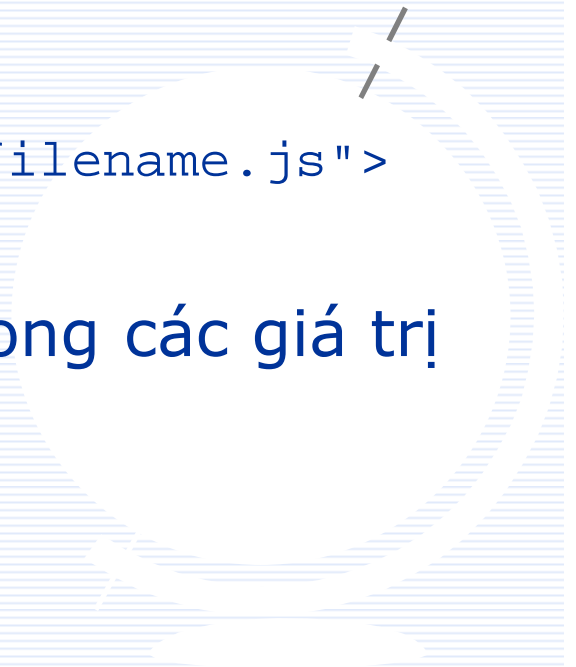
## ✓ Sử dụng thẻ SCRIPT:

```
<script language="JavaScript">  
  <!--  
    JavaScript statements;  
  //-->  
</script>
```

## ✓ Sử dụng một file JavaScript ở ngoài

```
<script language="JavaScript" src="filename.js">  
</script>
```

## ✓ Sử dụng các biểu thức JavaScript trong các giá trị thuộc tính của thẻ



# Các sự kiện JS hỗ trợ trong HTML

---

- onClick
- onChange
- onFocus
- onBlur
- onMouseOver

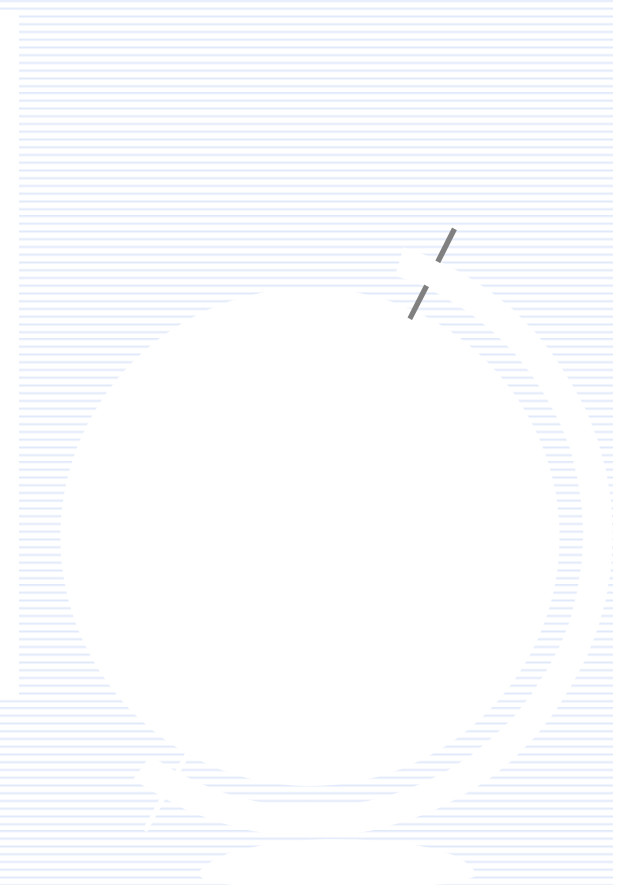
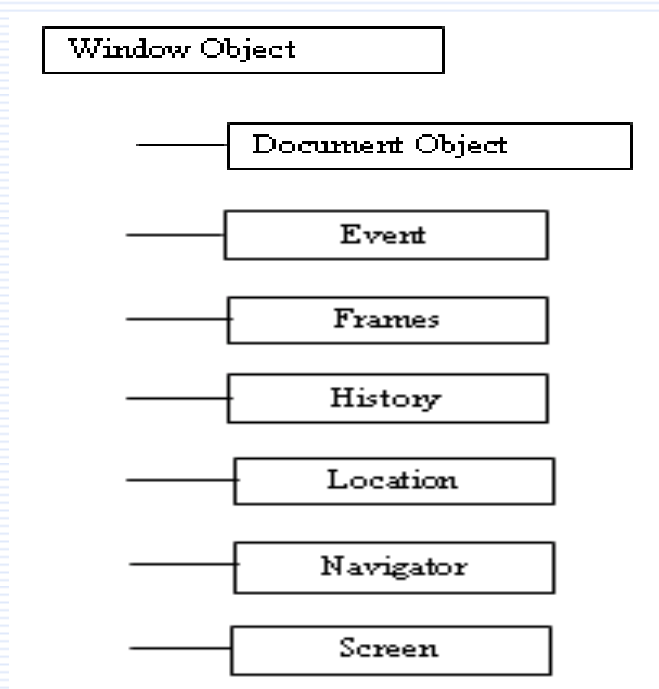
- onMouseOut
- onLoad
- onSubmit
- onMouseDown
- onMouseUp

✓ Đặt bộ lắng nghe sự kiện

```
<tên_thẻ tên_sự_kiện="tên_lệnh;">
```

# Các đối tượng trong trình duyệt

- ✓ Trình duyệt đưa ra một số đối tượng có thể được truy xuất bởi JavaScript



# HTML DOM

---

✓ Là một chuẩn được đưa ra bởi W3C, định nghĩa các truy xuất đến các văn bản như HTML

✓ Tài liệu tra cứu về HTML DOM có thể xem rất chi tiết tại địa chỉ sau:

[http://www.w3schools.com/js/js\\_obj\\_html\\_dom.asp](http://www.w3schools.com/js/js_obj_html_dom.asp)





# Website hỗ trợ CSS, HTML và PHP

---

- ✓ <http://www.w3schools.com/>
- ✓ <http://vn.php.net>
- ✓ <http://www.javascriptkit.com>
- ✓ <http://www.dynamicdrive.com>
- ✓ <http://www.javascriptbank.com>
- ✓ <http://www.dhtmlcentral.com>





**THE END**

# PHP The Fourth Day

**Đào Quang Minh**

**Trung tâm Tính toán Hiệu năng cao – ĐHBK Hà Nội**

**minh0302@gmail.com**

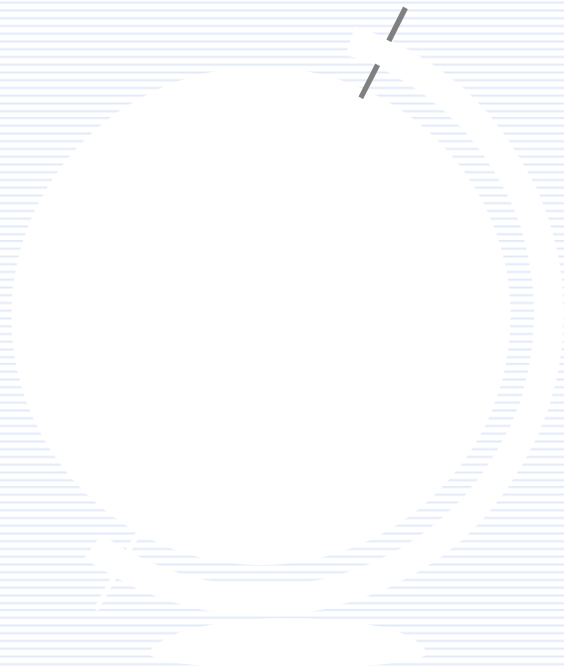
**<http://hpcc.hut.edu.vn/~minhdq>**

# Chương 3: PHP và MySQL

---

1. Các bước truy cập CSDL

2. SQL Injection



The background features a large, faint white circle centered on a light blue background with horizontal lines. Two orange lines, one horizontal and one vertical, intersect at the center of the circle, forming a crosshair. The text is positioned in the center of the circle.

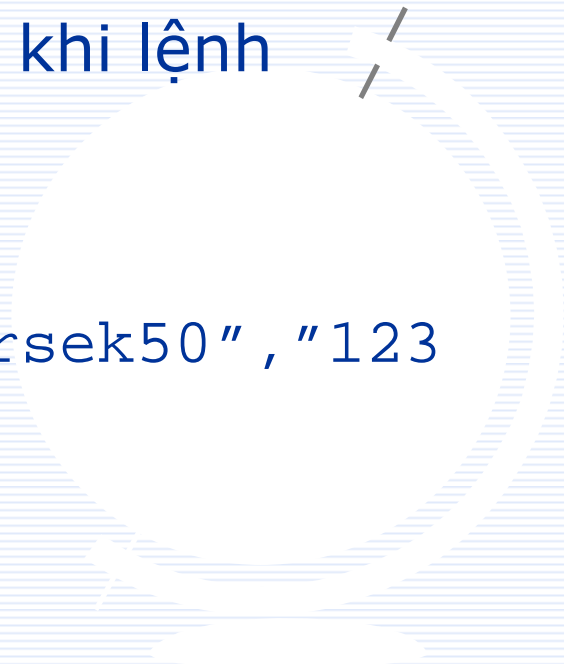
# **Các bước truy cập cơ sở dữ liệu**

# Tạo kết nối đến Database Server

```
$biến_kết_nối =  
mysql_connect("máy_chủ","tên","mật_khẩu")  
or die("Không kết nối được");
```

- ✓ Hàm `die("Chuỗi")`: Đưa ra thông báo và kết thúc.
- ✓ Với cách viết trên, `die` chỉ thực hiện khi lệnh trước nó không thành công

```
$cnn =  
mysql_connect("localhost","coursek50","123  
456");
```

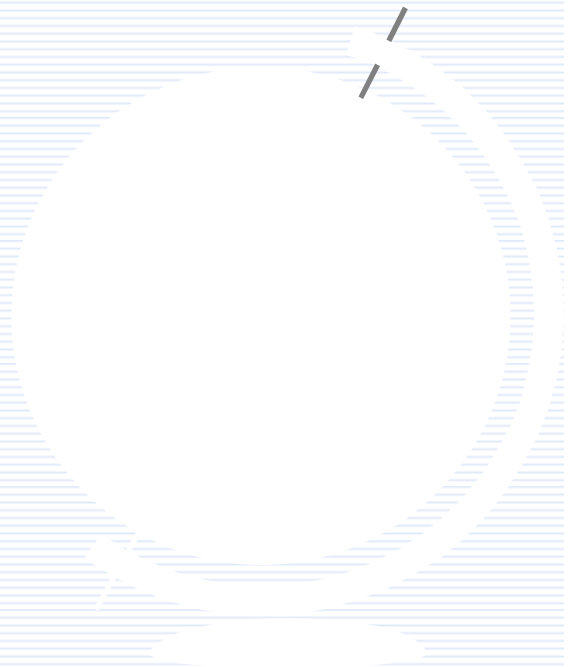


# Lựa chọn cơ sở dữ liệu

---

```
mysql_select_db("Tên CSDL")  
or die ("Chưa có CSDL");
```

```
mysql_select_db("coursek50");
```



## Viết truy vấn

---

```
$biển = mysql_query("Lệnh SQL")  
or die("Không thực hiện được SQL");
```

```
$sql = mysql_query("Select * From  
account Where username = 'coursek50'  
");
```





## Các hàm hỗ trợ

---

- ✓ `mysql_affected_rows()`: Số bản ghi bị tác động bởi lệnh `mysql_query` liền trước
- ✓ `mysql_num_rows()`: Kết quả số bản ghi của câu lệnh `mysql_query`
- ✓ `mysql_error()`: Thông báo lỗi (nếu có)
- ✓ `mysql_errno()`: Mã lỗi



## Các hàm xử lý kết quả

---

- ✓ `mysql_fetch_array($sql)`: trả về một dòng bản ghi dưới dạng một mảng với chỉ mục dạng số hoặc tên của trường.
- ✓ `mysql_fetch_row($sql)`: trả về kết quả là một mảng có thứ tự (bắt đầu từ 0)
- ✓ `mysql_fetch_assoc($sql)`: trả về kết quả là một mảng được đánh chỉ số bằng tên

Tham khảo tại: <http://vn2.php.net/mysql>



# SQL Injunction

# Thế nào là SQL Injection

---

- ✓ Là một kỹ thuật cho phép những kẻ tấn công thi hành các câu lệnh truy vấn SQL bất hợp pháp
- ✓ Bằng cách lợi dụng lỗ hổng trong việc kiểm tra dữ liệu nhập trong các ứng dụng web



## Ví dụ

```
$sql=mysql_query("Select * From  
account Where username = '$user' And  
password = '$pass'")
```

- ✓ Nếu như User nhập biến User là: "OR 1 OR user=" thì câu lệnh SQL sẽ là:

```
SELECT * FROM users WHERE user = "" OR 1  
OR user="" AND password = "$password"
```

- Và kết quả trả về sẽ là toàn bộ user trong cơ sở dữ liệu

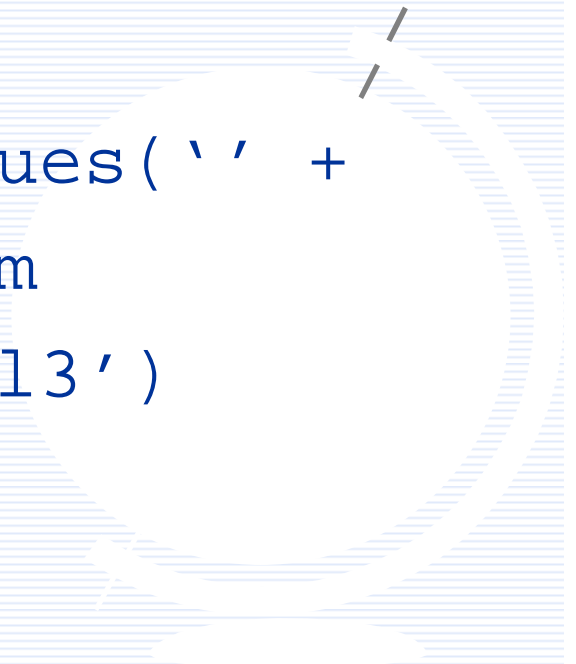
(tiếp)

```
$sql="Insert Into table_name  
Values( '$val1', '$val2', '$val3' )";
```

✓ Nếu như các biến val1 thành như sau:

```
` + (Select Top 1 Fieldname From  
TableName) +`
```

→ Insert Into TableName Values( ` ` +  
Select Top 1 FieldName From  
TableName)+ ` `, '\$val2, '\$val3' )



## Cách phòng tránh

---

- ✓ SQL Injection khai thác những bất cẩn của các lập trình viên phát triển ứng dụng web khi xử lý dữ liệu nhập vào
- ✓ Tùy vào môi trường và cách cấu hình hệ thống mà tác hại là nặng hay nhẹ



(tiếp)

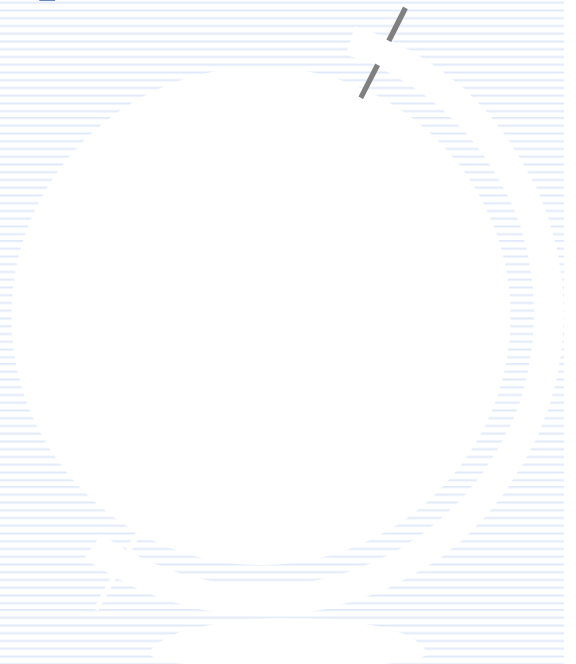
- ✓ Kiểm soát chặt chẽ dữ liệu nhập vào
  - Giới hạn chiều dài của chuỗi nhập liệu
  - Xây dựng hàm thay thế dấu 1 nháy đơn bằng 2 dấu nháy đơn
  - Xây dựng hàm loại bỏ một số kí tự và từ khóa nguy hiểm
  - Dùng hàm addslashes để thêm “\” vào trước chuỗi nhập vào



## Ví dụ

- ✓ Hàm length để giới hạn độ dài
- ✓ Hàm thay thế ` thành ``

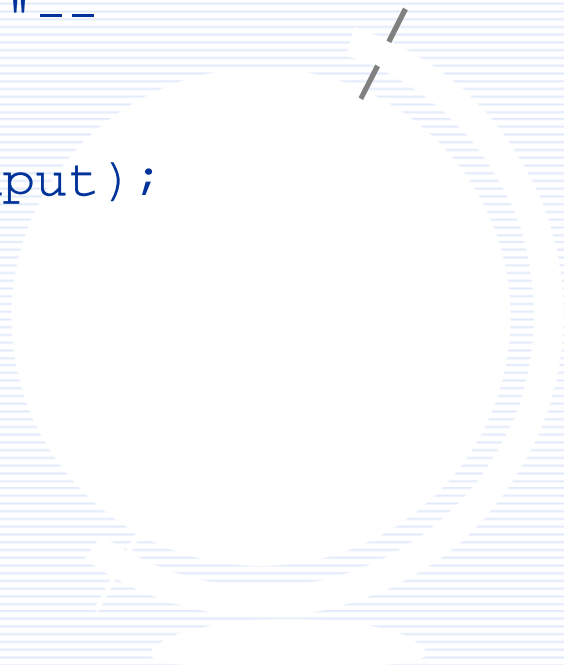
```
function replace($input)
{
    $output = str_replace("`", "`", $input);
    return $output;
}
```



(tiếp)

- ✓ Hàm loại bỏ các kí tự không mong muốn

```
function killchar($input)
{
    $len = strlen($input);
    $char = array("select", "drop", ";", "--",
    ", "insert", "delete", "xp_");
    $output = str_replace($char, "", $input);
    return $output;
}
```



The background features a large, semi-transparent white circle centered on a light blue background with horizontal lines. Two thin orange lines intersect at the center of the circle, forming a crosshair. The text 'Lập trình AJAX' is positioned to the left of the circle's center, overlapping the white area.

# Lập trình AJAX

# Giới thiệu

---

- ✓ AJAX = **A**synchronous **J**avaScript **A**nd **X**ML
- ✓ AJAX không phải là ngôn ngữ mới mà là một cách mới sử dụng các ngôn ngữ đã có.
- ✓ AJAX giúp ta tạo các trang web nhanh hơn, tiện lợi hơn, thân thiện hơn khi sử dụng.
- ✓ AJAX là tập hợp của nhiều công nghệ với thế mạnh của riêng mình để tạo thành một sức mạnh mới

# Đặc điểm AJAX

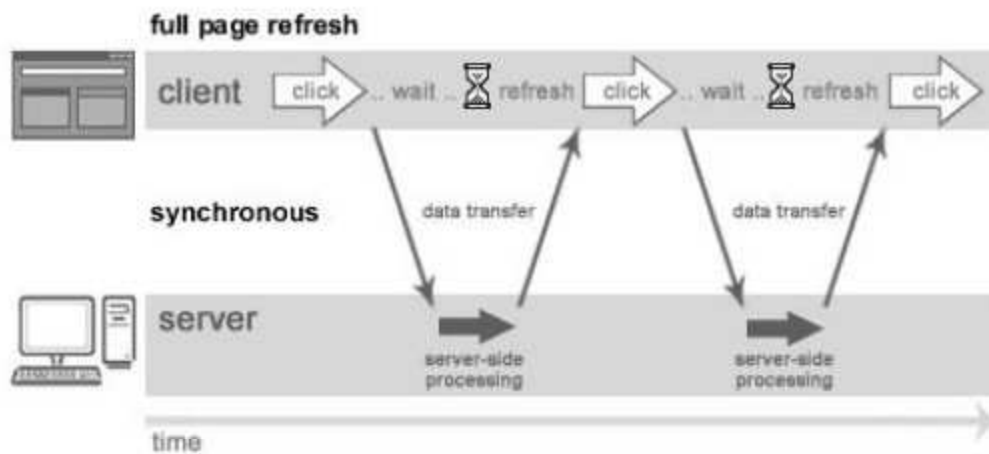
---

- ✓ Là công nghệ của web browser. Độc lập với web server.
- ✓ Sử dụng JavaScript để gửi và nhận dữ liệu giữa client và server.
- ✓ AJAX dựa trên:
  - JavaScript
  - XML
  - HTML
  - CSS
- ✓ AJAX sử dụng XML và HTTP Request

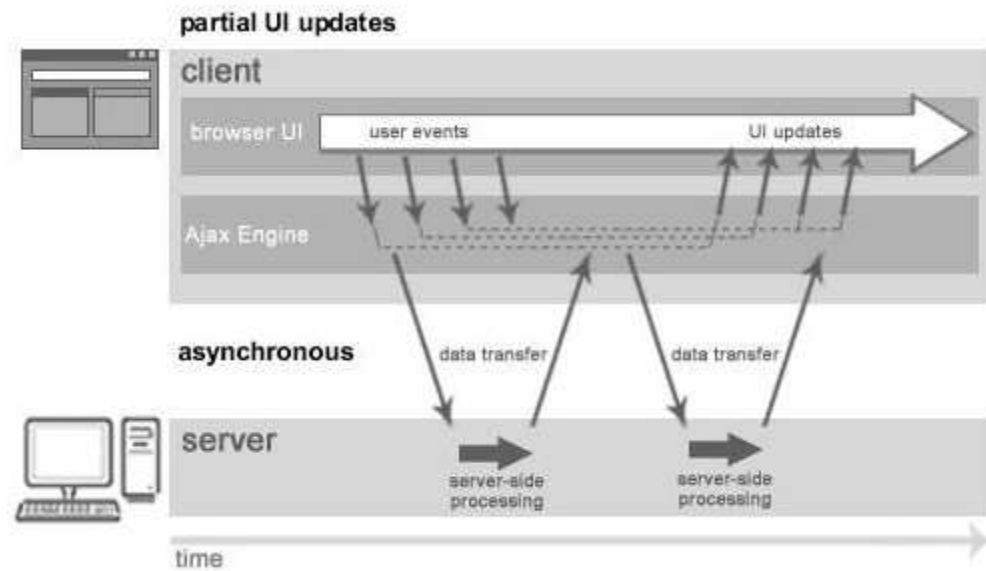


# So sánh

## Click – Wait-And-Refresh



## Ajax based



# Truy xuất dữ liệu từ server trong JavaScript

---

- ✓ Với IE6: sử dụng Msxml2.XMLHTTP:

```
XMLHttp =
```

```
    new ActiveXObject("Msxml2.XMLHTTP")
```

- ✓ Với IE5.6: sử dụng Microsoft.XMLHTTP

```
XMLHttp =
```

```
    new ActiveXObject("Microsoft.XMLHTTP")
```

- ✓ Với trình duyệt khác: XMLHttpRequest

```
XMLHttp=new XMLHttpRequest()
```



# Mã lệnh tổng quát

```
function GetXMLHttpRequest() {
    var oXMLHttp=null
    try{
        oXMLHttp=new ActiveXObject("Msxml2.XMLHttp")
    }
    catch(e) {
        try{
            oXMLHttp=new ActiveXObject("Microsoft.XMLHttp")
        }
        catch (e) {}
    }
    if (oXMLHttp==null)
        oXMLHttp=new XMLHttpRequest()
    return oXMLHttp;
}
```



# Đối tượng XMLHttpRequest

---

## ✓ Các phương thức

- **open("method","url")**: Thiết lập yêu cầu đến server (địa chỉ trang cần kết nối đến)
- **send(content)**: Gửi yêu cầu đến server.
- **abort()**: Hủy yêu cầu hiện tại

## ✓ Các thuộc tính

- **readyState**: Trạng thái hiện tại của đối tượng.
- **onreadystatechange**: Bộ xử lý sự kiện cho một sự kiện phát sinh khi có thay đổi trạng thái
- **responseText**: Chuỗi dữ liệu trả về.

(tiếp)

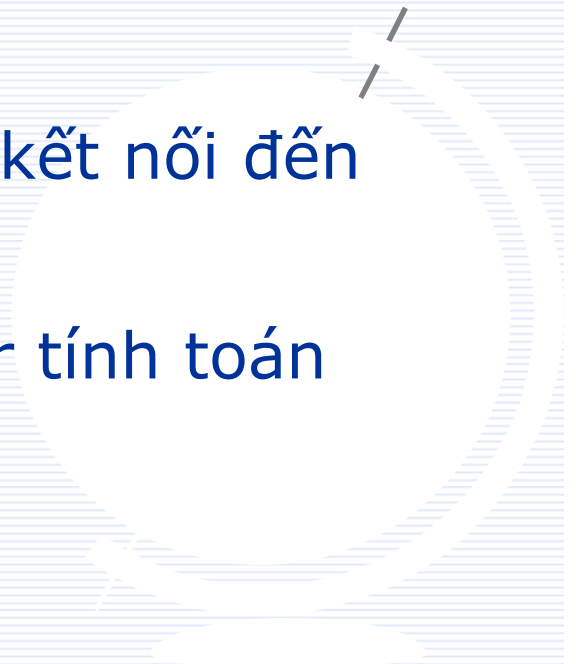
---

- ✓ **responseXML** Response trả về từ server dưới dạng XML. Đối tượng này có thể được phân tách và khảo sát như một đối tượng tài liệu DOM.
- ✓ **status** Mã trạng thái HTTP từ server (chẳng hạn 200 nếu không có lỗi, 404 cho lỗi Not Found, ...).
- ✓ **statusText** Thông điệp của mã trạng thái HTTP (chẳng hạn OK hay Not Found, ...)

# Thuộc tính readyState

---

- ✓ **readyState=0**: Chưa khởi tạo: sau khi tạo đối tượng XMLHttpRequest nhưng chưa gọi open().
- ✓ **readyState=1**: Vừa khởi tạo: sau khi gọi open() nhưng chưa gọi send().
- ✓ **readyState=2**: Vừa gửi đi: ngay sau khi gọi send().
- ✓ **readyState=3**: Đang xử lý: sau khi kết nối đến server nhưng server chưa trả lời.
- ✓ **readyState=4**: Xong: sau khi server tính toán xong, dữ liệu đã gửi về xong.

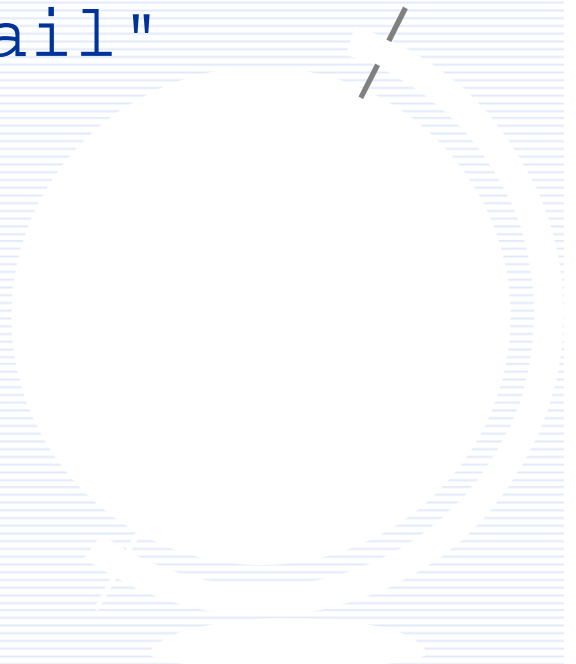


# Quá trình tương tác

---

1. Một event client-side gây ra 1 sự kiện – Ajax event. Bất kỳ một tác động nào cũng có thể gây ra Ajax event

```
<input type="text" id="email"  
name="email" onblur =  
"validateEmail()" ;>
```



(tiếp)

## 2. Một thể hiện của XMLHttpRequest được tạo ra:

- dùng phương thức open()
- tạo lời gọi hàm
- địa chỉ URL được thiết lập cùng với phương thức Http được yêu cầu
- request được tạo ra qua việc gọi send()



(tiếp)

```
var xmlHttp;
function validateEmail() {
var email = document.getElementById("email");
var url = "validate?email=" + escape(email.value);
if (window.ActiveXObject) {
xmlHttp = new ActiveXObject("Microsoft.XMLHTTP");
}
else if (window.XMLHttpRequest) {
xmlHttp = new XMLHttpRequest();
}
xmlHttp.open("GET", url);
xmlHttp.onreadystatechange = callback;
xmlHttp.send(null);
}
```

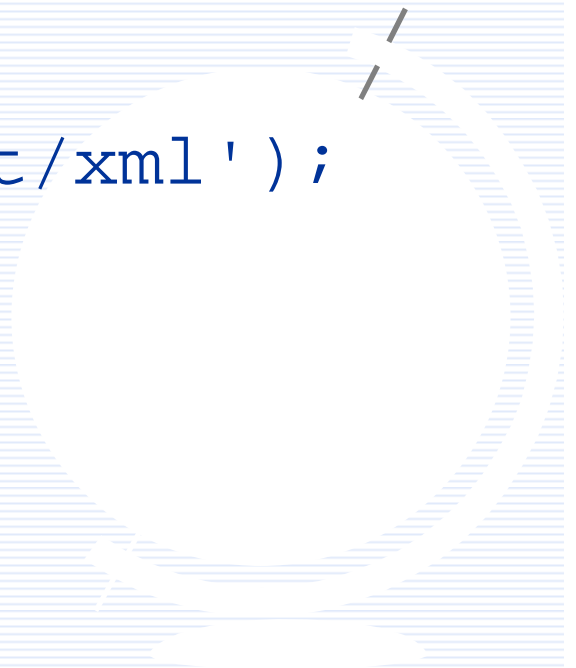


(tiếp)

---

3. Một request được tạo và gửi đến server
4. Server xử lý yêu cầu (truy cập cơ sở dữ liệu, tác vụ hệ thống...)
5. Response được trả về cho trình duyệt (xử lý ở dạng text/xml)

```
header( 'Content-Type: text/xml' );
```



(tiếp)

6. Cấu hình XMLHttpRequest để gọi hàm callback() khi kết quả xử lý được trả về

```
function callback() {  
    if (xmlHttp.readyState == 4) {  
        if (xmlHttp.status == 200) {  
            //do something interesting here  
        }  
    }  
}
```



# PHP The First Lesson

**Hanoi University of Science and Technology**  
**([hust.edu.vn](http://hust.edu.vn))**

# Chương 1: Tổng quan thiết kế Web

---

- ✓ Nhắc lại một số khái niệm
- ✓ Phân loại website
- ✓ Một số bước chính trong phát triển website
- ✓ Giới thiệu và hướng dẫn cài đặt các phần mềm hỗ trợ xây dựng website
- ✓ Viết trang PHP đầu tiên và đưa lên mạng





# Nhắc lại một số khái niệm

# Một số khái niệm cần thiết

---

## ✓ Domain Name

- Là tên được gắn với một địa chỉ IP
- Máy chủ DNS sẽ thực hiện việc gắn (ánh xạ) khi có yêu cầu truy cập
- Được chia làm nhiều cấp, phân biệt bởi dấu chấm
- Ví dụ: **hpcc.hut.edu.vn** gắn với 202.191.56.49
  - vn: Việt Nam (cấp 1)
  - edu: Tổ chức giáo dục (cấp 2)
  - hut: Tên cơ quan (cấp 3)
  - hpcc: Đơn vị trong cơ quan (cấp 4)
- Đặc biệt: tên **localhost** được gắn với 127.0.0.1



(tiếp)

---

## ✓ Máy chủ

- Là máy chuyên cung cấp tài nguyên, dịch vụ cho các máy tính khác
- Thường được cài đặt các phần mềm chuyên dụng để có khả năng cung cấp
- Một máy chủ có thể dùng cho một hay nhiều mục đích. Tên máy chủ thường gắn với mục đích sử dụng. Ví dụ: /
  - File server
  - Application server
  - Mail server
  - **Web server**
- Trên thực tế, máy chủ có cấu hình khá cao, với khả năng hoạt động ổn định

(tiếp)

---

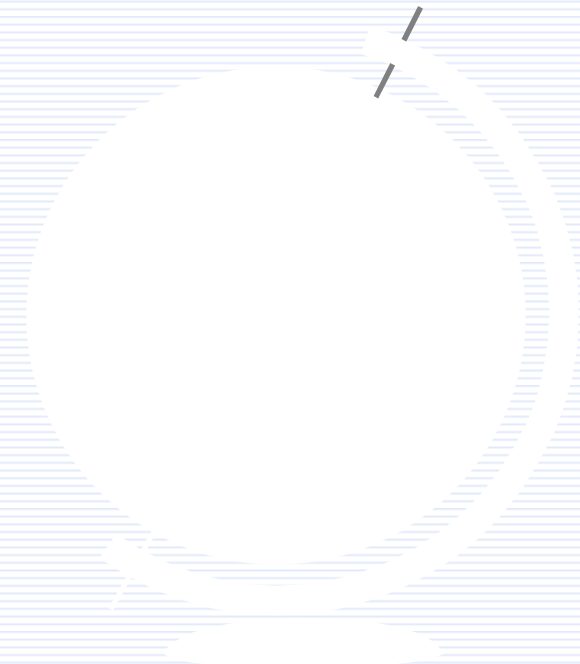
## ✓ Máy khách

- Máy khai thác dịch vụ của máy chủ
- Với mỗi dịch vụ thường có các phần mềm chuyên biệt để khai thác
- Một máy tính có thể vừa là client vừa là server
- Một máy tính có thể khai thác dịch vụ của chính nó

(tiếp)

## ✓ Cổng dịch vụ

- Là một số thuộc vào khoảng 0-65535, dùng để xác định dịch vụ của máy chủ
- Hai dịch vụ khác nhau sẽ chiếm các cổng khác nhau
- Mỗi dịch vụ thường chiếm các cổng xác định. Ví dụ theo mặc định sẽ là:
  - HTTP: 80
  - FTP: 21
  - SSH: 22
  - DNS: 53
  - SMTP: 25
  - POP3: 110



(tiếp)

## ✓ Chuỗi định vị tài nguyên (URL – Uniform Resource Locator)

- Tài nguyên: file trên mạng
- Xác định vị trí và cách khai thác các file
- Cấu trúc: giao\_thức://địa\_chỉ\_máy (hoặc IP)/đường\_dẫn/tên\_file
- VD: <http://www.hut.edu.vn/content/index.php>
- Trong trường hợp mặc định, nhiều thành phần có thể bỏ qua:
  - Giao thức, cổng: được trình duyệt mặc định
  - Tên file: được máy chủ đặt mặc định



(tiếp)

---

- ✓ Web page
  - Là một trang nội dung, có thể được viết bằng nhiều ngôn ngữ khác nhau nhưng có kết quả trả về client là HTML
- ✓ Website
  - Tập hợp các web page có nội dung thống nhất phục vụ cho một mục đích nào đó
- ✓ WWW (World Wide Web)
  - Tập hợp các website trên mạng Internet
- ✓ Web Server
  - Là máy phục vụ web (chạy trên máy chủ)
  - Một số phần mềm web server chuyên dụng như: Apache (open source) hoặc Internet Information Services – IIS (Microsoft)
- ✓ Web Browser
  - Phần mềm chạy trên client để khai thác dịch vụ web như: Mozilla Firefox, Internet Explorer, Google Chrome...

The background features a large, light blue circle with a white border, centered on a light blue background with a fine horizontal line pattern. Two orange lines, one horizontal and one vertical, intersect at the center of the circle. The text "Phân loại website" is written in a bold, black, sans-serif font across the center of the circle.

# Phân loại website

# Phân loại website

---

✓ Dựa vào công nghệ phát triển, có 2 loại:

– Web tĩnh:

- Dễ phát triển
- Tương tác yếu
- Sử dụng HTML
- Công cụ trực quan

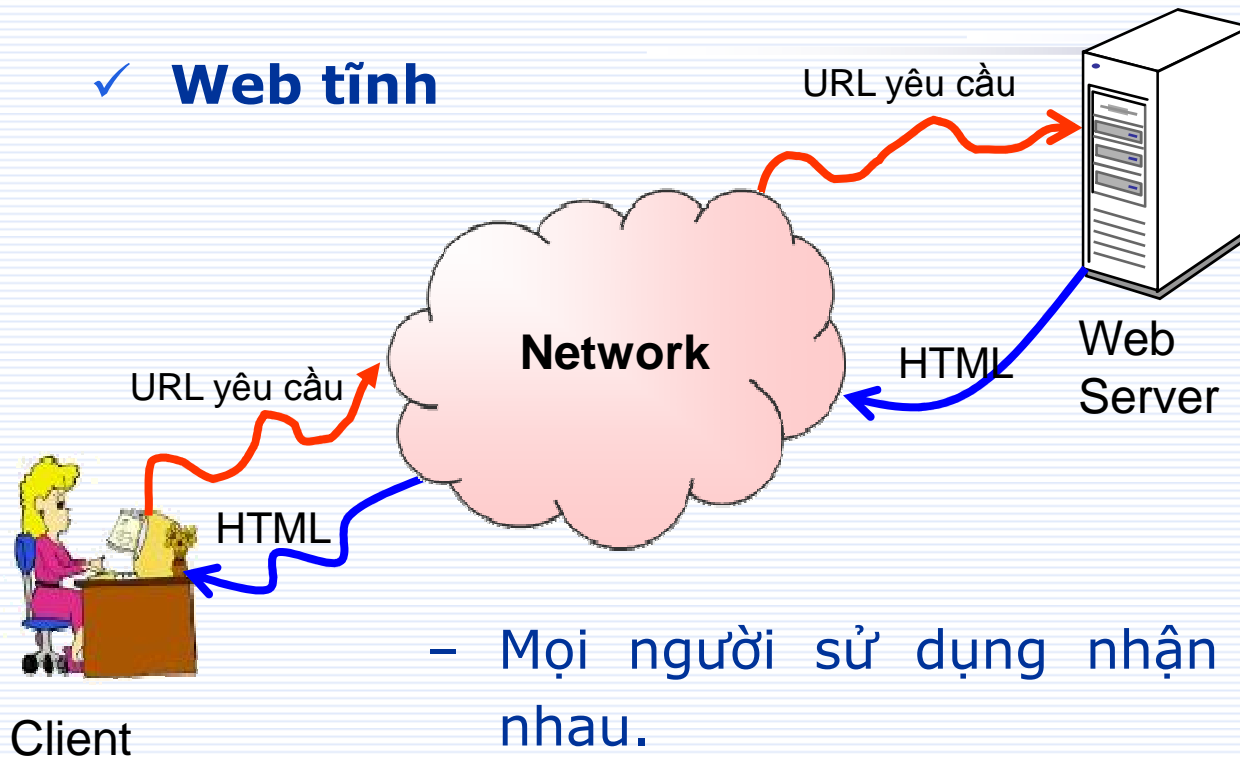
– Web động:

- Khó phát triển hơn
- Tương tác mạnh
- Sử dụng nhiều ngôn ngữ khác nhau
- Thao tác bằng tay để viết nhiều mã lệnh



(tiếp)

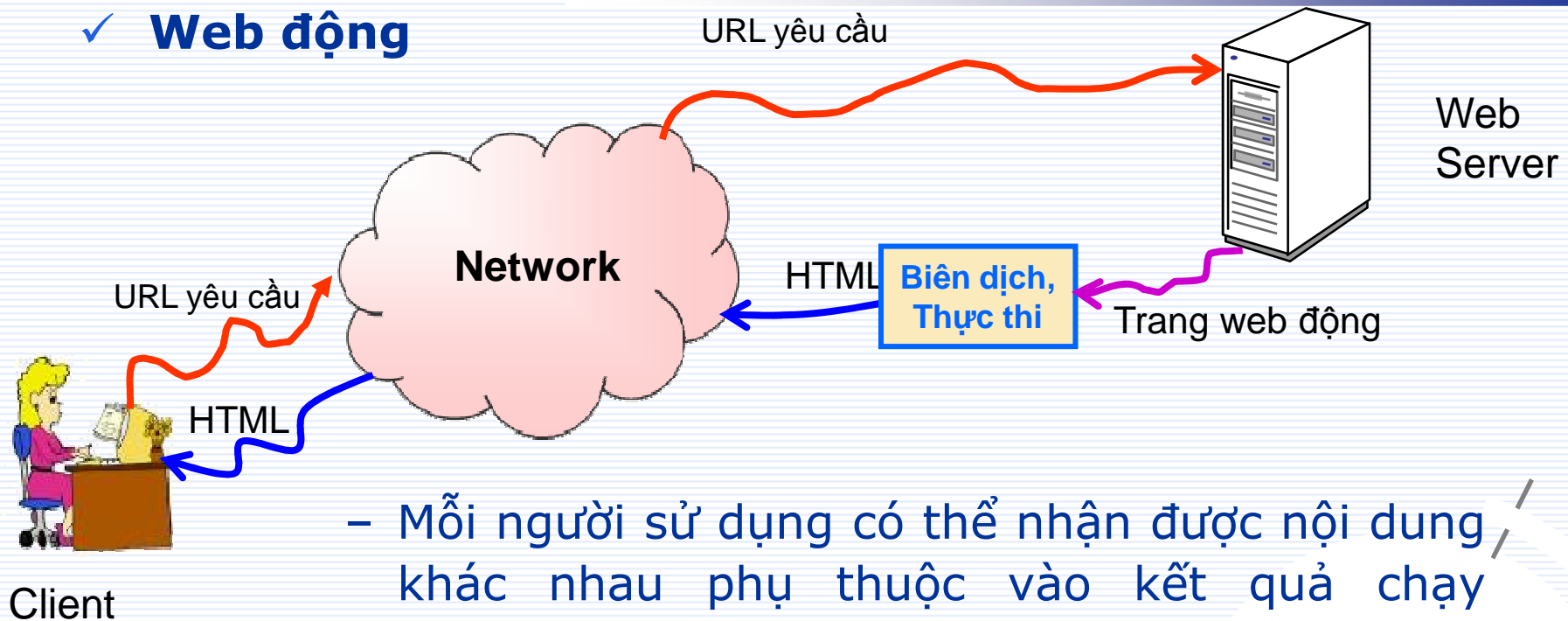
✓ **Web tĩnh**



- 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.

(tiếp)

✓ **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

(tiếp)

## ✓ Một số công nghệ viết web động

- Động phía Client:
  - JavaScript, VBScript được chạy ở client.
  - Applet
  - Flash
- Động phía Server:
  - Common Gateway Interface (CGI)
  - Java Server Pages (JSP)
  - ASP: Microsoft
    - Viết bằng VBScript, JavaScript chạy phía server.
    - Sử dụng web server IIS.
  - **PHP: Mã nguồn mở**
    - Ngôn ngữ lập trình PHP, chạy phía server.
    - Webserver: Apache, IIS
    - Bộ biên dịch: PHP
  - Perl





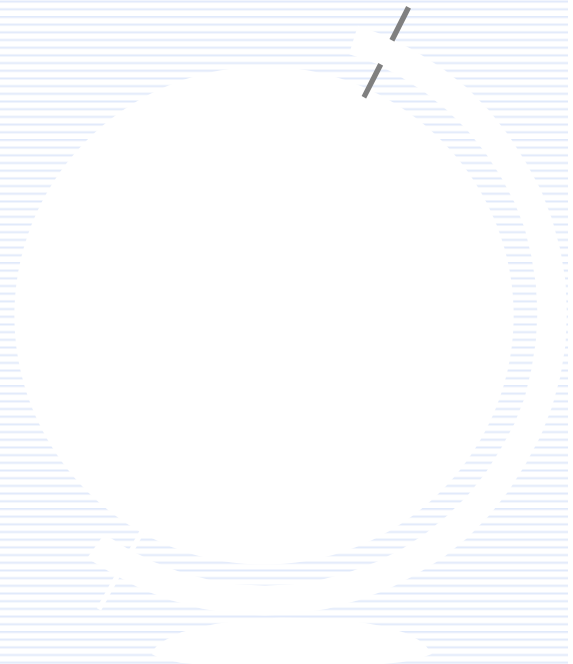
# **Một số bước chính trong phát triển website**

# Một số bước chính trong phát triển website

---

## ✓ Phân tích và thiết kế hệ thống

- Đặc tả
- Phân tích
- Thiết kế
- Lập trình
- Kiểm thử





(tiếp)

---

## ✓ Một số chú ý

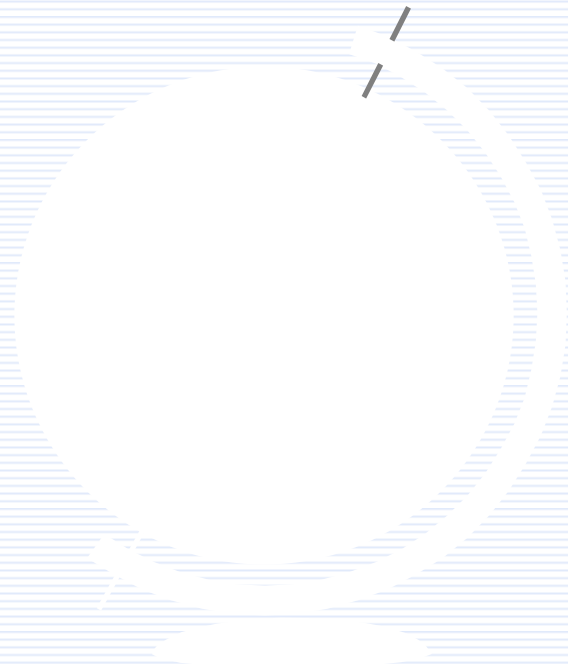
- Đặc tả: Website để làm gì? Ai dùng? Trình độ người dùng? Bố cục? Nội dung? Hình ảnh? v.v..
- Phân tích: Mối liên quan giữa các nội dung? Thứ tự các nội dung? (kịch bản website)
- Thiết kế: Sơ đồ cấu trúc website, giao diện, tĩnh hay động, CSDL, nội dung từng trang, liên kết giữa các trang v.v..
- Xây dựng: cấu trúc thư mục, các module dùng chung, v.v..

(tiếp)

---

## ✓ Kiểm thử

- Kiểm tra trên nhiều trình duyệt
- Kiểm tra trên nhiều loại mạng
- Kiểm tra tốc độ trên các loại đường truyền
- Kiểm tra các liên kết
- Thử các lỗi bảo mật
- v.v..





# **Các phần mềm hỗ trợ xây dựng website**

# Các phần mềm nền tảng xây dựng website

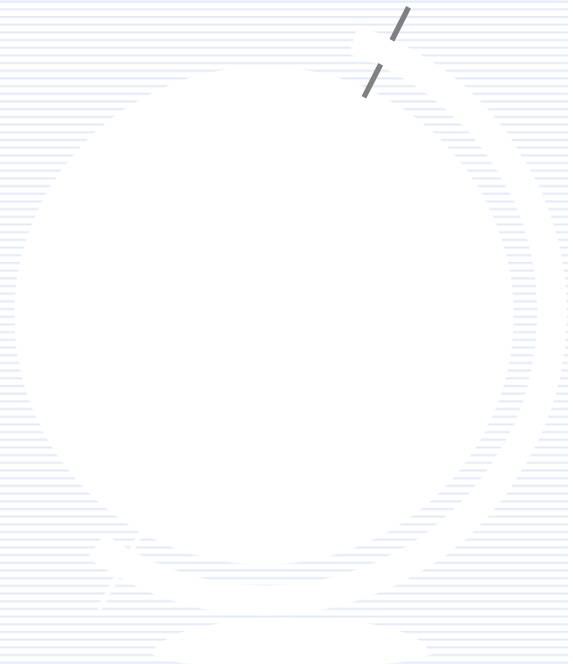
---

✓ **Operating System: Linux**

✓ **Web Server: Apache**

✓ **Database Server: MySQL**

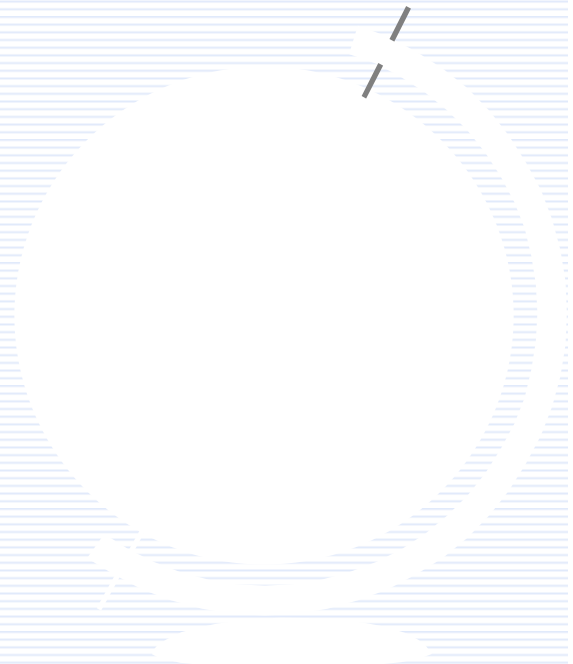
✓ **Interpreter: PHP**



# Các phần mềm hỗ trợ

---

- ✓ **Server Suites:** LAMP, XAMPP, EasyPHP...
- ✓ **Integrated Development Environment**
  - NetBean, Eclipse, Dreamweaver...
- ✓ **Database**
  - MySQL Administrator Tool
  - MySQL Query Browser



(tiếp)

## ✓ XAMPP

– Download từ website:

<http://www.apachefriends.org/en/xampp-linux.html>

hoặc:

<http://hpcc.hut.edu.vn/coursesk51/setups/xampp-win32-1.7.1-installer.exe>

– Cài đặt XAMPP: vào Terminal, gõ: ***tar xvfz xampp-linux-1.7.1.tar.gz -C /opt***

– Chú ý:

- Có thể xảy ra xung đột giữa MySQL được cài sẵn (từ trước) và MySQL của XAMPP

(tiếp)

---

## ✓ NetBean

– Download từ website:

<http://dlc.sun.com/netbeans/6.5/beta/>

hoặc:

<http://hpcc.hut.edu.vn/coursesk51/setups/netbeans-6.5-ml-linux.sh>

– Cài đặt: mở Terminal, gõ ***sh netbeans-6.5beta-linux.sh***

– Khởi động NetBean từ màn hình Desktop

(tiếp)

---

## ✓ MySQL

– Download từ website:

<http://dev.mysql.com/downloads/gui-tools/5.0.html>

– Tiến hành cài đặt thông qua cửa sổ dòng lệnh Terminal

– Truy cập cơ sở dữ liệu thông qua MySQL Administrator với username và password mà hệ thống cung cấp



The background features a large, faint white circle centered on a light blue background with a fine horizontal line pattern. Two thin orange lines intersect at the center of the circle, forming a crosshair. The text is positioned in the middle of this crosshair.

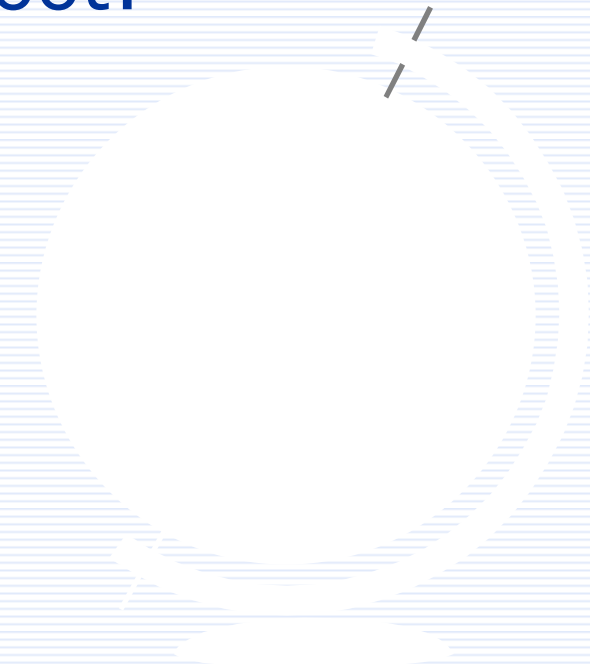
# How to write 1<sup>st</sup> PHP Page

# Viết chương trình PHP đầu tiên

---

## ✓ Cài đặt Xampp:

- Download gói cài đặt theo đường dẫn (ở phần trước)
- Khởi động xampp bằng quyền root
- Chuyển sang người dùng Root:  
`#sudo su`
- Khởi động Xampp:  
`/opt/lampp/lampp start`



# (tiếp)

- ✓ Sử dụng Nano hoặc Gedit để viết file php với những dòng lệnh sau:

```
<?php
```

```
$a = "PHP is so easy !";
```

```
echo $a;
```

```
?>
```

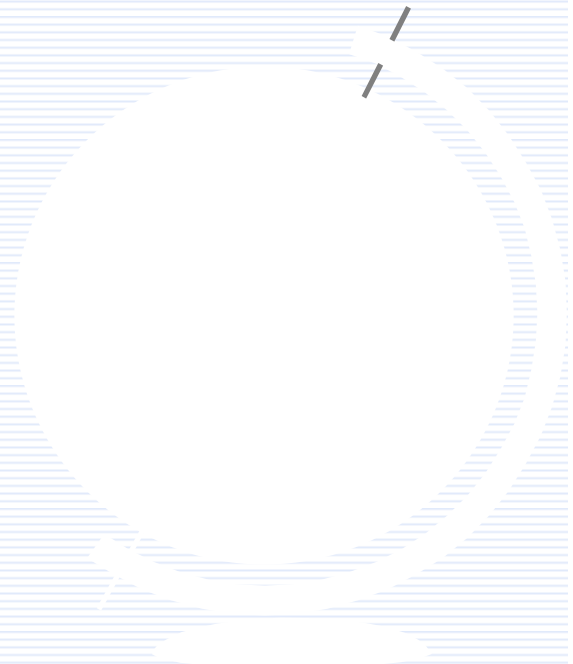
- ✓ Lưu lại file vừa viết thành: index.php
- ✓ Copy vào thư mục /opt/lampp/htdocs/coursek51/
- ✓ Chạy PHP từ web browser:  
<http://localhost/coursek51>



## Chương 2: Làm quen với PHP

---

- ✓ Tổng quan
- ✓ Cú pháp ngôn ngữ
- ✓ Cấu trúc điều khiển
- ✓ Lập trình hướng đối tượng
- ✓ Phương thức truyền dữ liệu
- ✓ Các khái niệm nâng cao





# Tổng quan về PHP

# Tổng quan về PHP

---

- ✓ PHP: viết tắt hồi quy "PHP: Hypertext Preprocessor"
- ✓ Là ngôn ngữ lập trình để viết các trang web động chạy trên server
- ✓ Bộ biên dịch PHP là phần mềm mã nguồn mở, có thể chạy cùng với các web server như Apache, IIS,...
- ✓ Thường kết hợp với hệ quản trị CSDL MySQL

(tiếp)

---

- ✓ Có khả năng hướng đối tượng
- ✓ Thông dịch
- ✓ Phân biệt chữ hoa và chữ thường
- ✓ Mỗi lệnh kết thúc bởi chấm phẩy (;)

# Cú pháp ngôn ngữ PHP



# Cú pháp ngôn ngữ PHP

---

## ✓ Chú thích trong PHP

- // hoặc # để bắt đầu một dòng chú thích.
- /\* ... \*/ để bắt đầu một đoạn chú thích

<?php

# Đây là chú thích

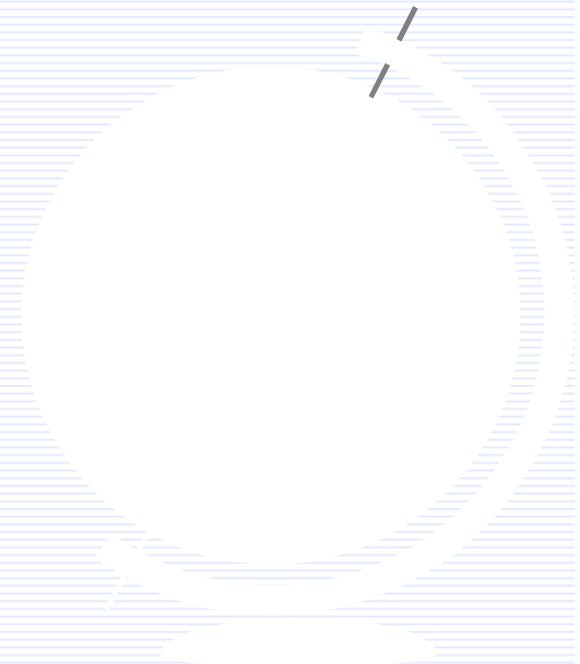
// Đây là chú thích

/\* Đây là chú thích 1

Đây là chú thích 2

/\*

?>



## ✓ Biến

- Phân biệt chữ hoa, chữ thường
  - Bắt đầu bằng dấu \$, tiếp ngay sau \$ là tên biến
- \$a = 5; \$A = 10;
- Tên biến bao gồm chữ cái, chữ số, dấu gạch nối dưới (\_) và phải bắt đầu bằng chữ cái hoặc (\_)

\$a\_1 = "Hello"; //hợp lệ

\$5b = 10; //không hợp lệ

\$trung-tam = "HPCC" //Không hợp lệ

(tiếp)

---

- Không cần khai báo trước (được tự động khai báo vào lần gán giá trị đầu tiên)
- Việc sử dụng biến chưa khởi tạo sẽ gây lỗi
- Cho phép sử dụng giá trị của biến làm tên biến khác

```
if($a < 10) //không hợp lệ khi chưa khởi tạo $a
```

```
$a = "temp";
```

```
$$a = 5; //tương đương $temp = 5;
```

(tiếp)

## ✓ PHP hỗ trợ 4 kiểu dữ liệu

### – Số:

- Số nguyên:  $-2^{31}$  đến  $2^{31} - 1$ 
  - + Hệ thập phân: `$a = 16;`
  - + Hệ 16 (hexa): `$a=0x10;`
- Số thực: từ  $1.7E-308$  đến  $1.7E+308$ 
  - + Biểu diễn: `$a = 0.017;`
  - + Dạng khoa học: `$a = 17.0E-03`

### – Chuỗi:

- Giới hạn bởi nháy đơn (') và kép (")
- Chuỗi đặt trong dấu nháy kép bị thay thế và xử lý ký tự thoát còn nháy đơn thì không. VD:

```
$a = "Hello";
```

```
$b = "$a World" //tương đương $b = "Hello World"
```

```
$c = '$a World' //tương đương $c = '$a World' (không thay đổi)
```

(tiếp)

## ✓ Chuỗi (tiếp)

- Để làm rõ các biến trong chuỗi, cần bao biến vào giữa cặp ngoặc nhọn { }
- Biến mảng, biến biển, biến trong tên có dấu gạch nối hoặc dấu chấm khi viết trong chuỗi bắt buộc phải bao lại

```
$a = "He";
```

```
$b = "$allo"; //lỗi vì PHP hiểu là $allo
```

```
$c_1 = "{$a}llo"; //đúng ($c_1 = "Hello")
```

```
$d = "{$c_1} World"; // $d = "Hello World"
```

## ✓ Chuỗi (tiếp)

- Ký tự thoát (\): được sử dụng để viết các ký tự đặc biệt trong chuỗi

Cần có chuỗi: He said: "PHP is good"!

→ \$a = "He said: \"PHP is good\"!";

- Một số ký tự buộc phải sử dụng ký tự thoát như \$, \, ", ... ngoài ra còn:

- \n: xuống dòng
- \r: trở về đầu dòng
- \t: dấu tab

(tiếp)

## ✓ Kiểu logic

– Có hai trạng thái là true và false

```
function status ($in)
```

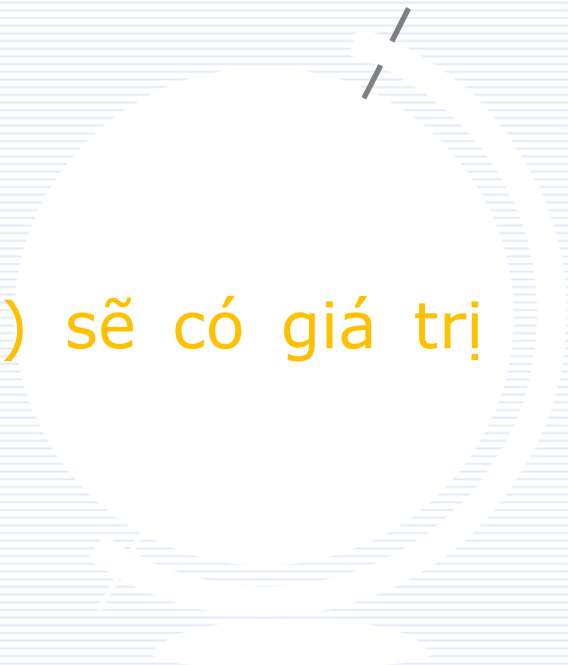
```
{
```

```
    if ($in < 5) return true;
```

```
    else return false;
```

```
}
```

// giá trị trả về của hàm status(...) sẽ có giá trị thuộc kiểu logic



## ✓ Kiểu mảng

- Cho phép chứa nhiều giá trị dữ liệu được đánh chỉ mục bằng số hay chuỗi
- Để truy cập vào 1 phần tử, đặt chỉ số trong [ ]
- Có thể sử dụng chỉ số không liên tiếp
- Có thể thêm phần tử vào cuối mảng bằng cách bỏ chỉ số

```
$a[0] = "Xin";
```

```
$a[] = "Chào"; // $a[1] = "Chào"
```



(tiếp)

## ✓ Kiểu mảng (tiếp)

– Chỉ số chuỗi

```
$a [ "ho" ] = "Nguyen" ;
```

```
$a [ "ten" ] = "Mai" ;
```

– Mảng nhiều chiều (số phần tử của mảng)

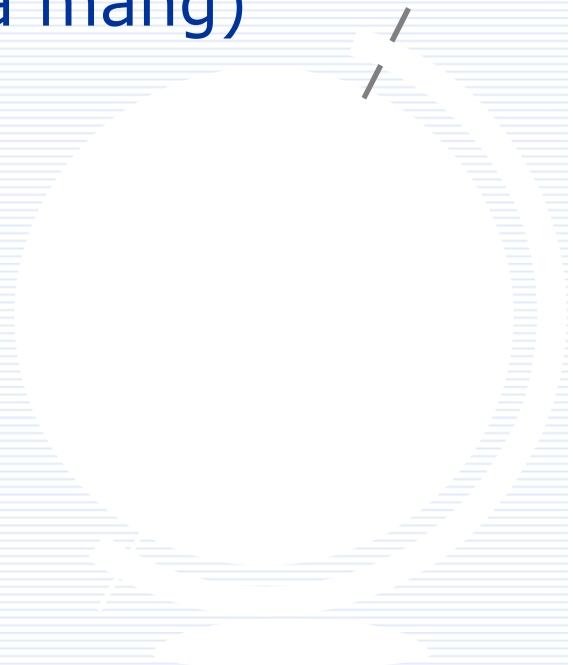
- Được coi là mảng của mảng

```
$a[0][0] = "STT" ;
```

```
$a[0][1] = "1" ;
```

```
$a[1][0] = "Ho ten" ;
```

```
$a[1][1] = "Nguyen Van A" ;
```



## ✓ Kiểu mảng (tiếp)

### – Khởi tạo mảng bằng hàm array()

- Chỉ số mặc định

```
$a=array("xin", "chào", "bạn");
```

```
/*tương đương với
```

```
$a[]="xin";
```

```
$a[]="chào";
```

```
$a[]="bạn";*/
```

- Chỉ số mặc định với chỉ số đầu khác 0

```
$a=array(5=>"xin", "chào", "bạn"); //phantu 5,6,7
```

```
/*tương đương với
```

```
$a[5]="xin";
```

```
$a[]="chào";
```

```
$a[]="bạn";*/
```

(tiếp)

✓ Khởi tạo mảng bằng hàm `array()` (tiếp)

– Chỉ số chuỗi:

```
$a=array("ten"=>"Hồng", "mau"=>"Đỏ");
```

/\*Tương đương với:

```
$a["ten"] = "Hồng";
```

```
$a["mau"] = "Đỏ";*/
```

– Mảng nhiều chiều:

```
$a = array(  
    array("TT", "Họ tên"),  
    array(1, "Nguyễn Văn A")  
);
```



(tiếp)

✓ Ép kiểu và kiểm tra kiểu

- Sử dụng cú pháp tương tự C/C++

```
$a = "123abc";
```

```
$so = (int) $a;
```

```
$kytu = (string) $a;
```

- Kiểm tra kiểu:

```
Gettype(), is_bool(), is  
_long(), ...
```

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

### – Kiểu số:

- Kết hợp: +, -, \*, /, %
- Tăng giảm: ++, --

VD: `$a++; $a--; ++$a; --$a;`

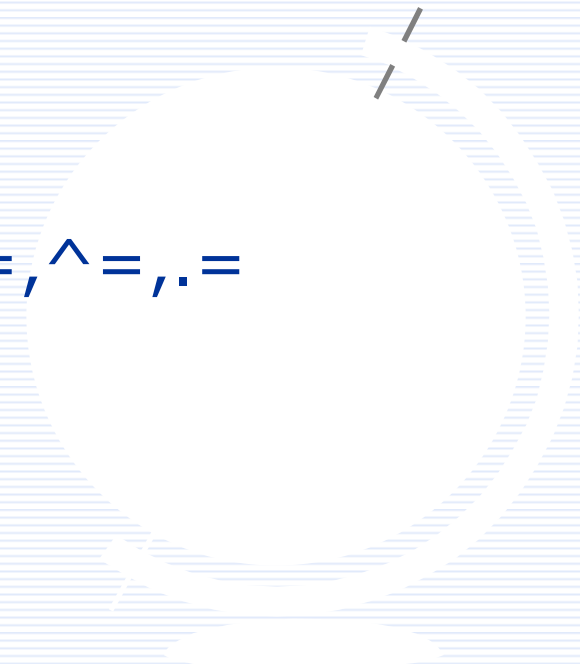
### – Chuỗi: Toán tử chấm (.)

### – Logic: AND (&&), OR (||), XOR, !

### – Gán: =, +=, -=, \*=, /=, %=, &=, ^=, .=

### – So sánh: ==, !=, <, >, <=, >=

### – Cấp phát: `new A();`



The background features a large, faint white circle centered on a light blue background with a fine horizontal line pattern. Two orange lines, one horizontal and one vertical, intersect at the center of the circle, forming a crosshair. The text is positioned in the center of the circle.

# Các cấu trúc điều khiển

# Câu lệnh rẽ nhánh **If**

```
if (BTLG)  
các_lệnh1;
```

```
if (BTLG)  
các_lệnh1;  
else  
các_lệnh2;
```

```
if (BTLG1)  
    lệnh1;  
elseif (BTLG2)  
    lệnh2;  
elseif (BTLG3)  
    lệnh3;  
else  
    các_lệnh_khác;
```

```
$biến= BTLG?Giá_trị1:Giá_trị2
```

# switch

```
switch ($biến){  
    case (giá_trị1):  
        các_lệnh1;  
    case (giá_trị2):  
        các_lệnh2;  
    default:  
        các_lệnh_khác;  
}
```

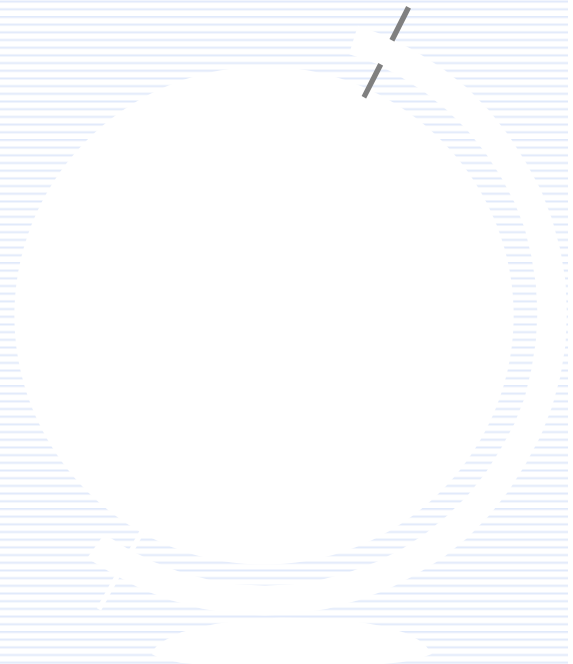


# Lặp không xác định **While...do**

---

```
while (BTLG)
{
    các_lệnh;
}
```

```
do {
    các_lệnh
} while (BTLG);
```



## Lặp xác định **for**

```
for (lệnh_khởi_tạo; BTLG; lệnh2){  
    các_lệnh;  
}
```

```
foreach ($biến_mảng as $giá_trị){  
    lệnh;  
}
```

```
foreach ($biến_mảng as $khóa=>$giá_trị){  
    lệnh;  
}
```

## Break và Continue

---

- ✓ **Break**: Kết thúc khối lệnh `for`, `while`, `do-while` hoặc `switch`
- ✓ **Continue**: bỏ qua phần ở sau `continue` trong vòng lặp, chuyển sang vòng lặp tiếp.



# Hàm tự tạo **function**

## ✓ Khái báo hàm

```
function tên_hàm(ts1, ts2,..., tsn) {  
    //các lệnh trong thân hàm  
}
```

- ✓ Để trả ra giá trị cho hàm ta sử dụng lệnh  
`return biểu_thức;`
- ✓ Để thoát khỏi hàm:
  - sử dụng `return`
  - hoặc `exit;`

## Hàm tự tạo (tt)

---

### ✓ Gọi hàm:

tên\_hàm(gt1, gt2, ..., gtn) ;

### ✓ Chú ý:

- Nếu hàm không có tham số, khi khai báo hàm, gọi hàm vẫn phải giữ lại ()
- Để khai báo hàm với các tham số mặc định, khi khai báo ta đưa ngay giá trị tham số vào.

**function** tên\_hàm(ts1=gt1, ts2=gt2, ...)

# Phạm vi biến

✓ Phạm vi biến là phạm vi ở đó biến xác định.

Trong PHP, biến có 3 phạm vi:

– Local variables - Biến cục bộ:

- Khởi tạo trong hàm
- Có tác dụng từ khi khởi tạo đến hết hàm.

– Global variables - Biến toàn cục:

- Khởi tạo ở ngoài hàm
- Có tác dụng từ khi khởi tạo đến hết file trừ ở trong hàm.
- Để đưa 1 biến toàn cục vào trong hàm, sử dụng khai báo **global** \$biên1, \$biên2,...; **\$\_GLOBAL**['biên']='giá trị'

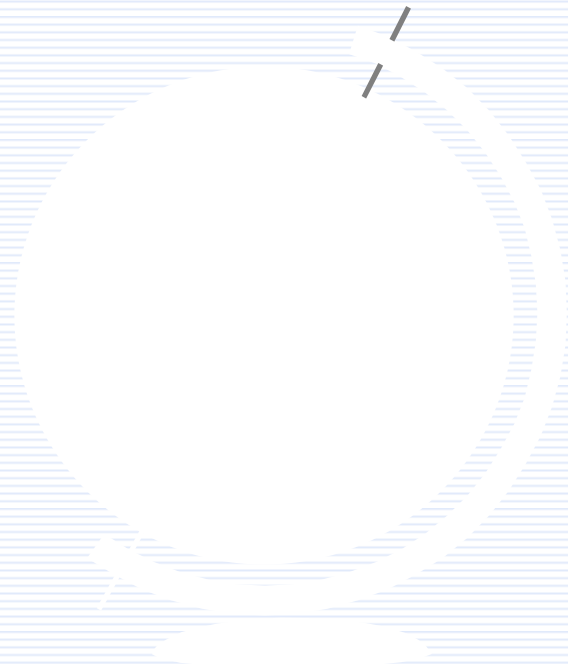
– Super Global Variables: Siêu toàn cục: Một số biến có sẵn của PHP: **\$\_SERVER**, **\$\_SESSION**...

# Biến tĩnh

---

- ✓ Khai báo trong hàm
- ✓ Giá trị được lưu trữ qua nhiều lần gọi hàm
- ✓ Chỉ được khởi tạo ở lần khai báo đầu tiên
- ✓ Để khai báo:

```
static $biến_tĩnh=giá trị;
```



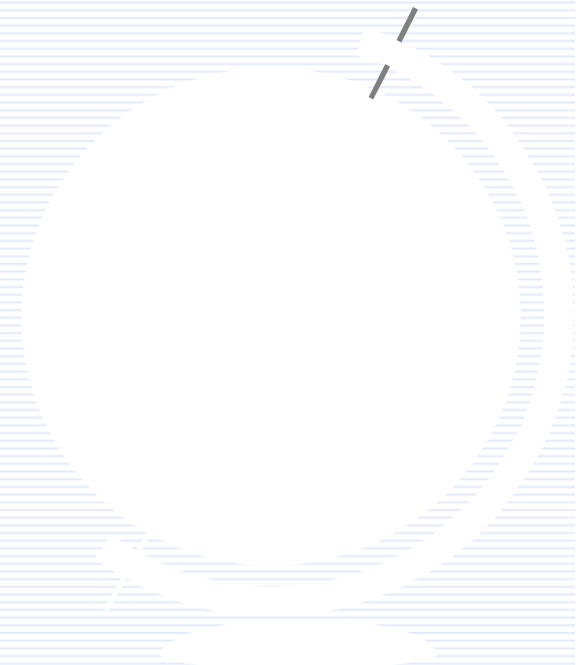
# Hằng số

---

✓ Khai báo: `define ("tên_biến", giá_trị)`

✓ Kiểm tra tồn tại hằng số bằng cú pháp:

`defined("tên_biến")`





# Chèn file

---

✓ `include("địa_chỉ_file");`

✓ `require("địa_chỉ_file");`

✓ `include_once("địa_chỉ_file");`

✓ `require_once("địa_chỉ_file");`





# The End of First Day



## Lập Trình Web

### PHP – Cơ Bản

GV: Trần Đình Nghĩa  
tdnghia1977@gmail.com



# Nội dung

1. Giới thiệu PHP
2. Cơ chế hoạt động của WebServer
3. Ví dụ minh họa
4. Cú pháp và quy ước trong PHP
5. Ví dụ và thực hành PHP cơ bản



# Giới thiệu PHP



- **PHP** : **P**HP **H**ypertext **P**roprocessor. Tên gốc là Personal Home Page.
- Là ngôn ngữ server-side-script thực thi phía WebServer trả kết quả về cho client.
- Cho phép người dùng tương tác với web site
- Thiên về xử lý nghiệp vụ.
- Tập tin PHP có phần mở rộng là **.php**
- Cú pháp ngôn ngữ giống ngôn ngữ **C++**



# Giới thiệu PHP



- Ưu điểm:
  - ↪ Đa môi trường: web-server, hệ điều hành, hệ quản trị cơ sở dữ liệu,...
  - ↪ Tốc độ truy xuất nhanh, Free, Open Source
  - ↪ Hướng đối tượng.
  
- Khuyết điểm:
  - ↪ Cú pháp rắc rối, chức năng quản lý lỗi.



# Giới thiệu PHP



## ➤ PHP Program

↪ <http://httpd.apache.org/download.cgi>

## ➤ Web Server : Apache

↪ <http://www.php.net/downloads.php>

## ➤ MySQL Database ( nếu dùng MySQL)

↪ <http://www.mysql.com/downloads/index.html>



# Giới thiệu PHP



- Download: XAMPP, WAMP, LAMP
  
- Phạm vi môn học:
  - ↪ Dùng **XAMPP** và local host
    - <http://localhost/>
    - <http://127.0.0.1>
  
- Download và cài đặt XAMPP:
  - ↪ <http://www.apachefriends.org/en/xampp.html>

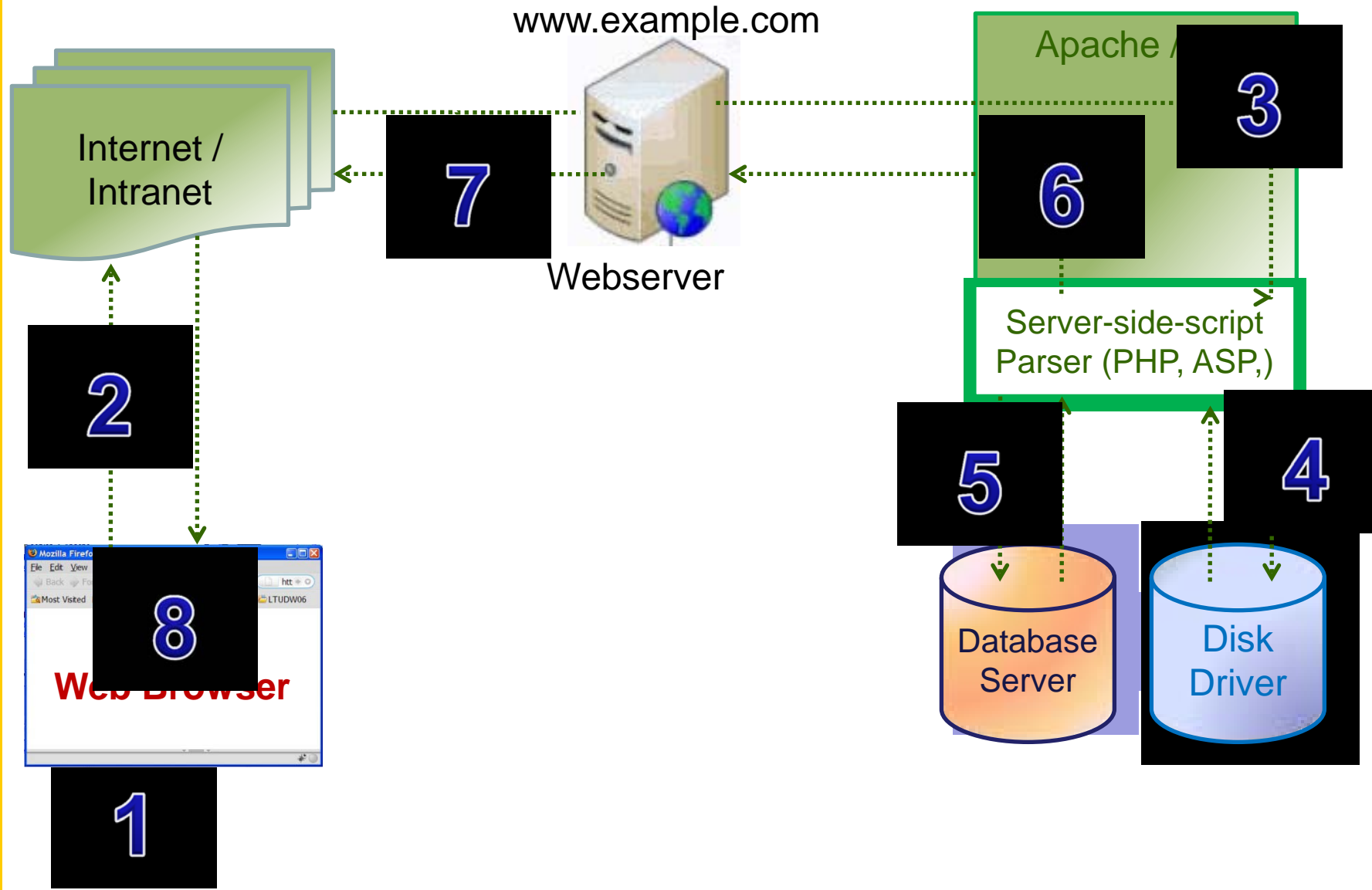




# Cơ chế hoạt động của WebServer



THIẾT KẾ VÀ LẬP TRÌNH WEB





# Ví dụ minh họa – VD1

C:\xampp\htdocs\T08\hello.php

<http://localhost/T08/hello.php>



```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Hello world</title>
  </head>

  <body>
    <script type="text/javascript">
      document.write("<br>");
      document.write("JavaScript: He
    </script>

    <?php
      echo("<br>");
      echo("PHP: Hello World
    ?>
  </body>
</html>
```

JavaScript  
Code

PHP Code



# Ví dụ minh họa – VD1



THIẾT KẾ VÀ LẬP TRÌNH WEB

JavaScript: Hello world!  
PHP: Hello World !

```
1 <html>
2   <head>
3     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
4     <title>Hello world</title>
5   </head>
6   <body>
7     <script type="text/javascript">
8       document.write("<br>");
9       document.write("JavaScript: Hello world !");
10    </script>
11    <br>PHP: Hello World ! </body>
12  </html>
```

JavaScript



# Ví dụ minh họa – VD2



```

2 <html>
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5     <title>Hello Everyone</title>
6   </head>
7
8   <body>
9     <?php
10      print("PHP hello everyone.<br>");
11    <?>
12
13    <?
14      print("<i>PHP</i> Hello everyone.<br>");
15    <?>
16
17    <script language="php">
18      print("<b>PHP</b> Heloo everyone.");
19    </script>
20  </body>
21 </html>

```

C:\xampp\htdocs\T08\helloEveryone.php  
<http://localhost/T08/helloEveryone.php>

Source of: http://localhost/T08/helloEveryone.php - Mozilla Firefox

```

1 <html>
2   <head>
3     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4     <title>Hello Everyone</title>
5   </head>
6
7   <body>
8     PHP hello everyone.<br>
9     <i>PHP</i> Hello everyone.<br>
10    <b>PHP</b> Heloo everyone.   </body>
11
12 </html>

```

PHP hello everyone.  
*PHP* Hello everyone.  
**PHP** Heloo everyone.



# Ví dụ minh họa – VD3

## JavaScript

## PHP

```

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Combo Date</title>
  </head>

  <body>
    <select name="cboNgay0" id="cboNgay0">
      <script type="text/javascript">
        for (i = 1; i <= 31; i++)
        {
          var s = '<option value="" + i + '>' + i + '</option>';
          document.write(s);
        }
      </script>
    </select>

    <select name="cboNgay1" id="cboNgay1">
      <?php
        for ($i=1; $i<=31; $i++)
        {
          $s='<option value ="' . $i . '>' . $i . '</option>';
          echo($s);
        }
      ?>
    </select>

  </body>
</html>

```

C:\xampp\htdocs\T08\comboDate.php  
<http://localhost/T08/comboDate.php>



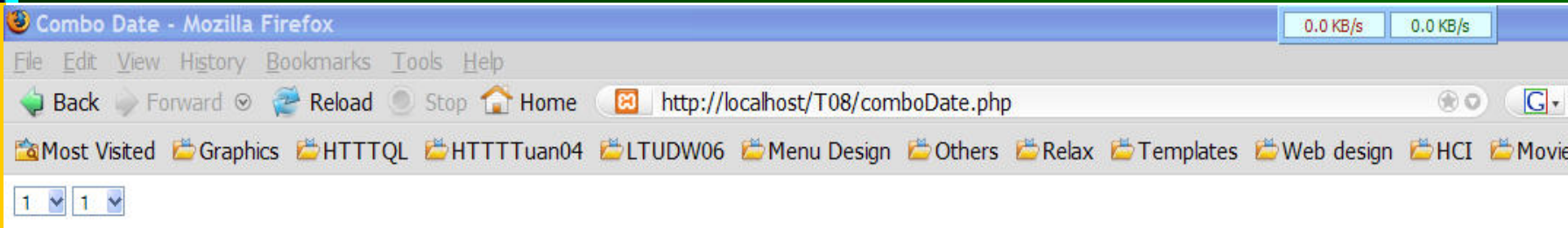




# Ví dụ minh họa – VD3



THIẾT KẾ VÀ LẬP TRÌNH WEB



```

Source of: http://localhost/T08/comboDate.php - Mozilla Firefox
File Edit View Help Html Validator

1
2 <html>
3   <head>
4     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5     <title>Combo Date</title>
6   </head>
7
8   <body>
9     <select name="cboNgay0" id="cboNgay0">
10      <script type="text/javascript">
11        for (i = 1; i <= 31; i++)
12        {
13          var s = '<option value="' + i + '>' + i + '</option>';
14          document.write(s);
15        }
16      </script>
17    </select>
18
19    <select name="cboNgay1" id="cboNgay1">
20      <option value = "1">1</option><option value = "2">2</option><option value = "3">3</option><option value = "4">4</opt
21    </select>
22  </body>
23 </html>

```

JavaScript

PHP

```

<option value = "1">1</option><option value = "2">2</option><option value = "3">3</option>
.....
<option value = "31">31</option>

```



# Ví dụ minh họa – Bug trong PHP



THIẾT KẾ VÀ LẬP TRÌNH WEB

Mozilla Firefox 0.1

File Edit View History Bookmarks Tools Help

Back Forward Reload Stop Home http://localhost/T08/bug.php

Most Visited Graphics HTTPQL HTTTTuan04 LTUDW06 Menu Design Others Relax Templates Web

Parse error: parse error in C:\xampp\htdocs\T08\Bug.php on line 10

Dw DESIGNER

File Edit View Insert Modify Format Commands Site Window Help

hello.php x helloEveryone.php x comboDate.php\* x Bug.php x va lap trinh web\Giao trinh\_LP\TW\T08\Demo\Bug.php

Code Split Design Live View Live Code Title: Bug bug bug

```
1 <html>
2   <head>
3     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
4     <title>Bug bug bug</title>
5   </head>
6
7   <body>
8     <?php
9       echo('Hello world' . <br>');
10      print('Hello everyone !');
11
12    ?>
13    <br>
14    Hello world
15  </body>
16 </html>
```

INSERT  
CSS STYLES  
AP ELEMENTS  
TAG INSPECTOR  
FILES  
ASSETS

Source of: http://localhost/T08/bug.php - Mozilla Firefox

File Edit View Help Html Validator

```
1 <br />
2 <b>Parse error</b>: parse error in <b>C:\xampp\htdocs\T08\Bug.php</b> on line <b>10</b><br />
3
```



# Cú pháp PHP cơ bản



1. Quy ước ngôn ngữ lập trình
2. Biến và kiểu dữ liệu
3. Toán tử và phép so sánh
4. Cấu trúc điều khiển
5. Hàm
6. Một số đối tượng xây dựng sẵn (build-in)





- Mã lệnh PHP được đặt trong các cặp tag:

Thẻ mở	Thẻ đóng
<?	?>
<?php	?>
<script language =“php”>	<script>

- Tất cả các câu lệnh kết thúc bằng dấu “;”
- Không phân biệt hoa thường, khoảng trắng, tab, xuống dòng trong câu lệnh.



## ➤ Cú pháp và ký hiệu tương tự như C++

//	Ghi chú 1 dòng	/* ..*/	Ghi chú 1 khối
{ }	Khối lệnh	[ ]	Sử dụng cho mảng
( )	Sử dụng cho hàm	\n, \t	Xuống hàng, ký tự Tab
'	Ký tự nháy đơn trong chuỗi	"	Ký tự nháy kép trong chuỗi

➤ Chuỗi phân biệt trong dấu nháy đơn ' và dấu nháy kép "



# Ví dụ: Phân biệt chuỗi



```
<?php
    print("I said: 'Hello' # ");
    print('I said: "Hello" # ' . '<br>');
    print("I said: \"Hello' # ");
    print('I said: "Hello\' # ');
?>
```

String PHP Demo - Mozilla Firefox

File Edit View History Bookmarks Tools Help

Back Forward Reload Stop Home <http://localhost/T08/string.php>

Most Visited Graphics HTTTQL HTTTTuan04 LTUDW06 Menu Design Other

I said: 'Hello' # I said: "Hello" #  
I said: "Hello' # I said: "Hello' #



# Biến và kiểu dữ liệu



- Kiểu dữ liệu:
  - ↪ boolean (bool)
  - ↪ integer (int)
  - ↪ double (float, real)
  - ↪ string
  - ↪ array
  - ↪ Object
- Khai báo biến: `$ten_bien = value`
- Không cần khai báo kiểu dữ liệu & khai báo biến
- Biến được tự động khởi tạo ở lần đầu gán giá trị



# Ví dụ



```
<?php
    $num =2.7;
    echo('Type of ' . $num . ' is : ' . gettype($num) . '<br/>');
    $num = 5;
    echo('Type of ' . $num . ' is : ' . gettype($num) . '<br/>');
    $str="Anyone";
    echo('Type of ' . $str . ' is : ' . gettype($str) . '<br/>');
    $bool = true;
    echo('Type of ' . $bool . ' is : ' . gettype($bool) . '<br/>');
?>
```

```
Type of 2.7 is : double
Type of 5 is : integer
Type of Anyone is : string
Type of 1 is : boolean
```

Một biến có thể có nhiều kiểu dữ liệu khác nhau



- Phạm vi biến: trong 1 file php
  - ↪ Không hoạt động trong 1 file php khác
  - ↪ Không hoạt động trong hàm (ngoại trừ sử dụng biến global.
  - ↪ Hoạt động trong tất cả các đoạn code php (bắt đầu từ lúc khởi gán)



# Ví dụ



```
The name is: <? echo($isName);?> <br>
<hr>
```

```
<?php
  if ($isDisplay == '1')
  {
    //Nghe em vào đại học
  }
  else
  {
    //Quê hương
  }
?>
```

```
<?php
if ($isDisplay == '1')
{
  // Nghe em vào đại học

  Miền Nam em ơi, còn nhớ <br />
  Kháng chiến năm nào gian khổ <br />
  Đồn giặc bủa vây thôn xóm điêu tàn <br />
  Trường: giặc đốt rồi, còn lại ánh trăng <br />
  Giữa hai trận càn, anh dạy em học chữ <br />
  Mẩu than em vẽ lên tường gạch đỏ <br />
  Những lá cờ sao bên những vòm tròn <br />
  Đầu nghiêng nghiêng theo nét chữ run run <br />

```

```
<?php
}
else
{
  // Quê hương

```

```

  Quê hương là chùm khế ngọt <br />
  Cho con trèo hái mỗi ngày <br />
  Quê hương là đường đi học <br />
  Con về rợp bướm vàng bay <br />
  Quê hương là con diều biếc <br />
  Tuổi thơ con thả trên đồng <br />
  Quê hương là con đò nhỏ <br />
  Êm đềm khua nước ven sông <br />

```

```
<?php
}
?>
```

```
<?php
$isDisplay = $_GET['display'];
//echo($isDisplay . ':' . gettype($isDisplay));
$isName = 'Cao Thanh';
?>

<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>Variable</title>
</head>
```

Hiển thị text HTML kết hợp code PHP

C:\xampp\htdocs\T08\variable.php

<http://localhost/T08/variable.php?display=1>

Miền Nam em ơi, còn nhớ  
Kháng chiến năm nào gian khổ  
Đồn giặc bủa vây thôn xóm điêu tàn  
Trường: giặc đốt rồi, còn lại ánh trăng  
Giữa hai trận càn, anh dạy em học chữ  
Mẩu than em vẽ lên tường gạch đỏ  
Những lá cờ sao bên những vòm tròn  
Đầu nghiêng nghiêng theo nét chữ run run



# Biến và kiểu dữ liệu



- Kiểu dữ liệu đơn (scalar type): boolean, integer, float / double, string
- Kiểu dữ liệu phức (compound type): array, object
- Kiểu dữ liệu đặc biệt: NULL, resource (<http://www.phpbuilder.com/manual/en/language.types.php>)
- Trong PHP, 1 biến có thể thay đổi kiểu dữ liệu trong quá trình hoạt động

↪ `$var = 1;`

↪ `$var = '1';`





- Các toán tử số học: với  $y = 3$

Operator	Description	Example	Result
+, -, *, /	Cộng, trừ, nhân, chia		
%	Modulus (div remainder)	$x = y\%2$	$x = 1$
++	Increment	$x = ++y$	$x = 4$
--	Decrement	$x = --y$	$x = 2$

- Toán tử ghép chuỗi . :

↪ `var $x = 5, $y = $x . " tiết"; // y = "5 tiết"`

↪ `var $z = 2 . "+" . 3; // z = "2+3"`



- Toán tử gán:  $x = 10$  và  $y = 5$

Operator	Example	Same as	Result (x)
=	$x = y$		5
+=	$x += y$	$x = x + y$	15
-=	$x -= y$	$x = x - y$	5
*=	$x *= y$	$x = x * y$	50
/=	$x /= y$	$x = x / y$	2
%=	$x \% = y$	$x = x \% y$	0



- Toán tử so sánh:  $x = 5$

Operator	Description	Example
=	Bằng	$x = 8 \rightarrow \text{false}$
===	Chính xác bằng (value & type)	$x === "5" \rightarrow \text{false}$
!=	Khác	$x != 8 \rightarrow \text{true}$
>	Lớn hơn	$x > 8 \rightarrow \text{false}$
<	Nhỏ hơn	$x < 8 \rightarrow \text{true}$
>=	Lớn hơn hoặc bằng	$x >= 8 \rightarrow \text{false}$
<=	Nhỏ hơn hoặc bằng	$x <= 8 \rightarrow \text{true}$



# PHP – Logical Operators



Operator	Description	Example	Result
&&	AND	X = 6 Y = 3 (X<10 && Y>1)	TRUE
	OR	X = 6 Y = 3 (X==5    Y==5)	FALSE
!	NOT	X = 6 Y = 3 !(X==Y)	TRUE



# PHP - Cấu trúc điều khiển



- if và if { } else { }
- switch
- while
- do { } while
- for
- foreach
  
- break và continue



# PHP – Cấu trúc điều khiển

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>PHP - Cau Truc Dieu Khien</title>
  </head>

  <body>
    <?php
      $i = 5;
      while($i > 0)
      {
        echo($i--);

      }

      print('<br>');

      do
      {
        echo(--$i);
      }
      while($i > 0);
    ?>
  </body>
</html>
```

Phân biệt sự khác nhau giữa 2 cấu trúc

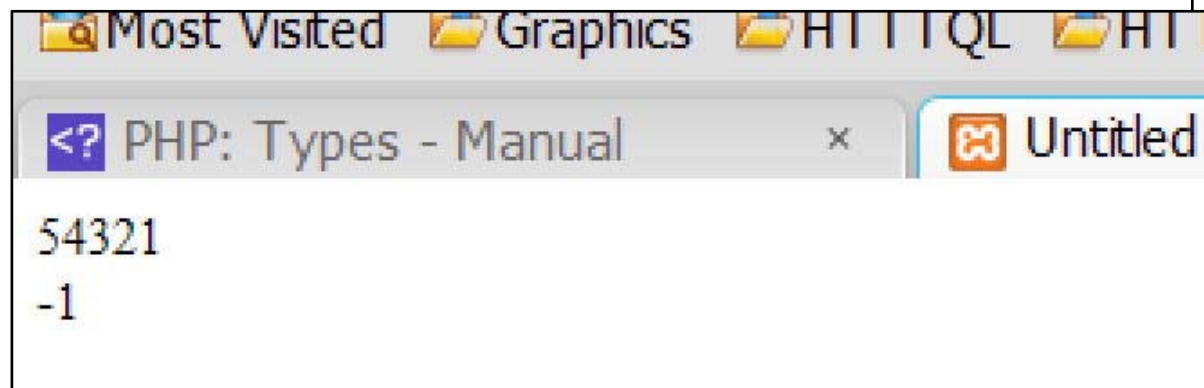
While (<ĐK>) {}

Kiểm tra ĐK đúng → thực hiện

và

Do {} while (<ĐK>)

Thực hiện → kiểm tra ĐK đúng





# Cấu trúc điều khiển – IF – IF ELSE



```

<?php
$array = array(); //gọi hàm tạo mảng
echo "<br>";
if (getenv('a') == 0) // a là số chẵn
{
    for($i = 0; $i < count($array); $i++)
    {
        $array[$i] = 2*$i;
    }
}
else
{
    $array[0] = 7; $array[1] = 9;
    foreach($array as $i)
    {
        echo ($i . " ");
    }
    print_r($array);
}
}

```

Array

7  
9

Array ( [0] => 7 [1] => 9 )



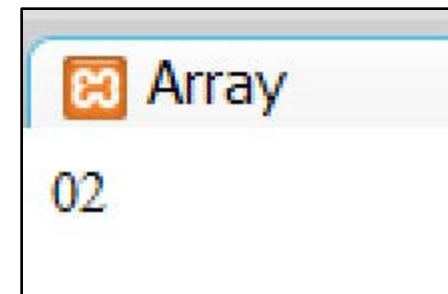
# Cấu trúc điều khiển - FOR



```
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
  <title>Array</title>
</head>

<body>
  <?php
    for($i = 0; $i < 5; $i++)
    {
      $array[$i] = $i; // gán giá trị cho mảng
    }
    for($i = 0; $i < count($array); $i++)
    {
      if ($i > 3)
        break;

      if ($array[$i] % 2 == 0)
        print($array[$i] + '<br>');
      else
        continue;
    }
  ?>
</body>
```



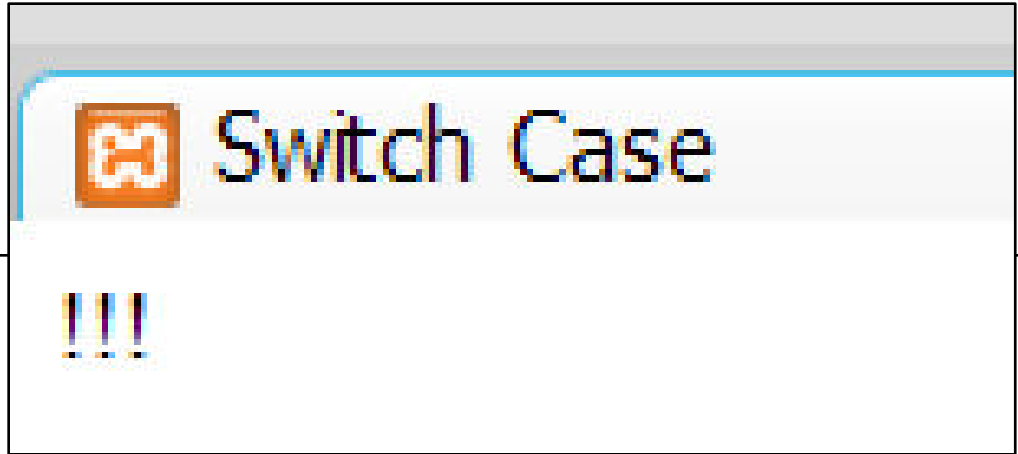




# Cấu trúc điều khiển – SWITCH CASE



```
$i= "c"; // switch can be applied to Number or String
switch ($i)
{
case "a":
    echo('a');
    break;
case "b":
    echo('b');
    break;
default:
    echo('!!!');
}
```





# PHP - Function



- Khai báo với từ khoá function
- Không cần kiểu trả về
- Nên khai báo ở đầu file PHP hoặc file riêng.

```
function tenHam($thamSo1,...)
{
    //Code
}
```

```
function tenHam($thamSo1,...)
{
    // code
    return (xyz);
}
```



# PHP – Function – Phạm vi biến

Biến toàn cục trong PHP

```

<?php
function thanhToanTien($tong=5)
{
    global $temp;
    $temp = $tong;
}

$temp = 5;
thanhToanTien(10);
echo "\$temp là: $temp" ;

```

\$temp = 5  
\$temp = 10

Thêm trị và thêm biến

```

<?php
function thanhToanTien($tong=5) //thêm trị
{
    global $variable;
}

$variable = 5;
thanhToanTien($variable);
echo "\$variable là: $variable" ;

```

\$variable = 5  
\$variable = 10



# PHP – Function – Include & Require

```
php x index.php x functions.inc x function1.php x function2.php x lib.php x
functions.inc
Split Design Live View Live Code Title: Simple function call

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Simple function call</title>
  </head>

  <body bgcolor="#ffffff" >
    <?php
      //include "functions.inc";
      require "functions.inc";
      bold("this is bold");
      $myString = "this is bold";
      bold($myString);
    ?>
  </body>

</html>
```

```
php x index.php x functions.inc x function1.php
Split Design Live View Live Code
<?php
  //functions.inc
  function bold($string)
  {
    echo "<b>" . $string . "</b>\n";
  }
?>
```



# PHP- Function

## ➤ Truyền tham số cho hàm

- ↪ Truyền tham trị
- ↪ Truyền tham biến bằng toán tử **&**

## ➤ Sử dụng từ khóa global và static trong hàm

- ↪ global: truy cập các biến ở bên ngoài hàm
- ↪ static: biến chỉ khởi gán giá trị 1 lần và được giữ nguyên nếu hàm tiếp tục được gọi

## ➤ Sử dụng từ khóa **include** và **require**

- ↪ <http://www.w3schools.com/php> (Basic reference)
- ↪ <http://vn2.php.net/manual/en/funcref.php> (Full reference)

## ➤ Google!!!



# PHP- Một số chức năng xây dựng sẵn (Build In)



- Chức năng xử lý thời gian
- Chức năng kiểm tra, đổi kiểu dữ liệu
- Chức năng toán học
- Chức năng xử lý chuỗi
  
- Chức năng kết nối & xử lý cơ sở dữ liệu
- Chức năng xử lý truy cập hệ thống file, thư mục
- Chức năng nén / giải nén file
- Chức năng liên quan tới protocol HTTP, FTP...



# Hướng đối tượng - CLASS



```
class class_name() [extends superclass_name]
{
    var $attribute;...
    function method_name()
    {
        $this->attribute = ...;
    }
    //code
}
$a = new class_name (...);
```

- <http://vn2.php.net/manual/en/language.oop5.php>





# Ví dụ



```

</php>
class Counter
{
    var $count = 0;
    var $increment = 0;
    function increment()
    {
        $this->count++;
    }
}

$c = new Counter();
$c->increment();
echo $c->count; // prints 1
$c->count = 101;

<br>
</php>
echo $c->count; //prints 101.

```



# Tham khảo

- <http://www.phpbuilder.com/manual/en/language.types.php>
- <http://www.w3schools.com/php/>
- <http://www.developer.com/lang/php/article.php/900521>
- <http://vn2.php.net/manual/en/langref.php> (reference)
- <http://vn2.php.net/manual/en/funcref.php> (reference)
- Google!!!



## Lập Trình Web

### THỰC HÀNH PHP – Cơ Bản

GV: Trần Đình Nghĩa  
tdnghia1977@gmail.com



- Download và cài đặt **XAMPP**
- Thực hành lại các đoạn code PHP trong bài giảng.



## Lập Trình Web

### PHP – Truyền và Nhận dữ liệu

GV: Trần Đình Nghĩa  
tdnghia1977@gmail.com

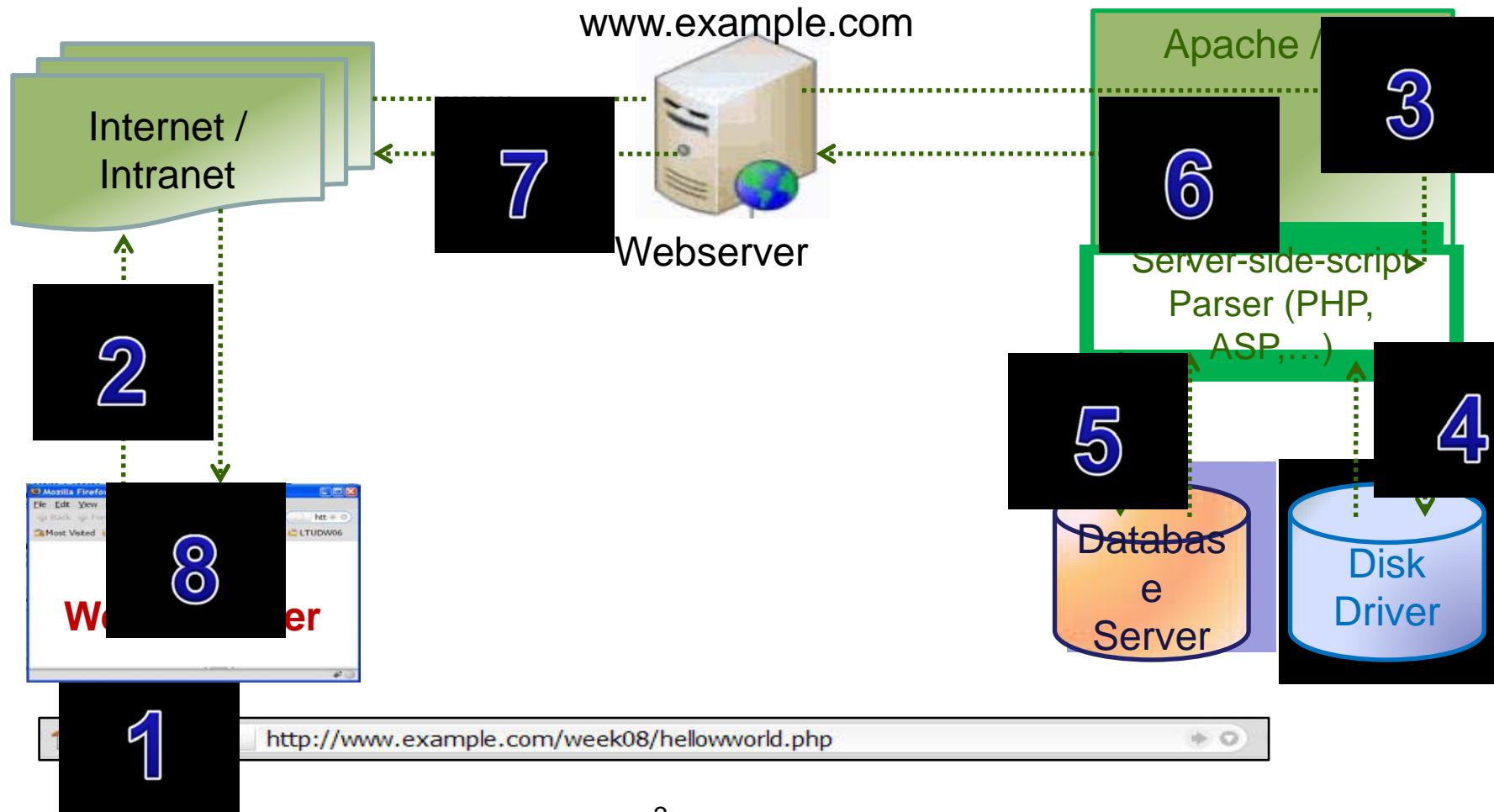


1. Cơ chế truyền nhận dữ liệu.
2. Truyền nhận qua phương thức GET
3. Truyền nhận qua phương thức POST
4. Một số ví dụ
5. Thực hành



# Cơ chế truyền nhận dữ liệu

- Web browser (client) gửi dữ liệu đến website thông qua các HTML Control (Form Field).

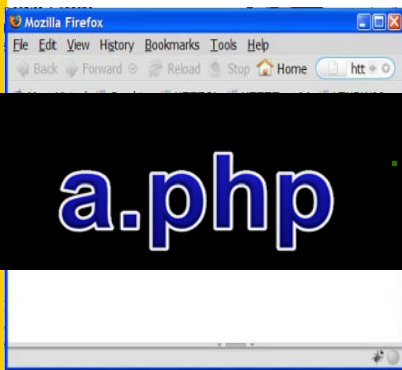




# Cơ chế truyền nhận dữ liệu

- Web server nhận và xử lý dữ liệu và trả kết quả cho client.

## Web Browser



2

Yêu cầu trang **b.php**

Internet /  
Intranet

7

www.example.com



Webserver

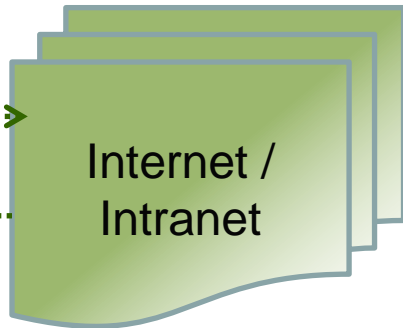
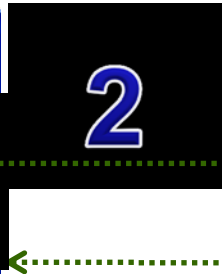
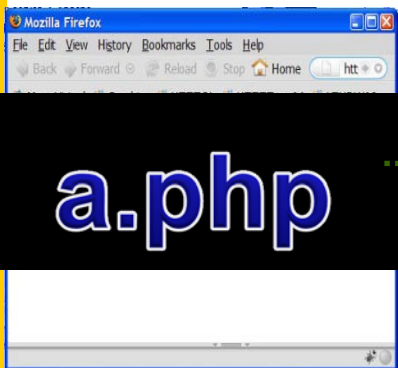


# Cơ chế truyền nhận dữ liệu



THIẾT KẾ VÀ LẬP TRÌNH WEB

**Web Browser**



www.example.com



Webserver





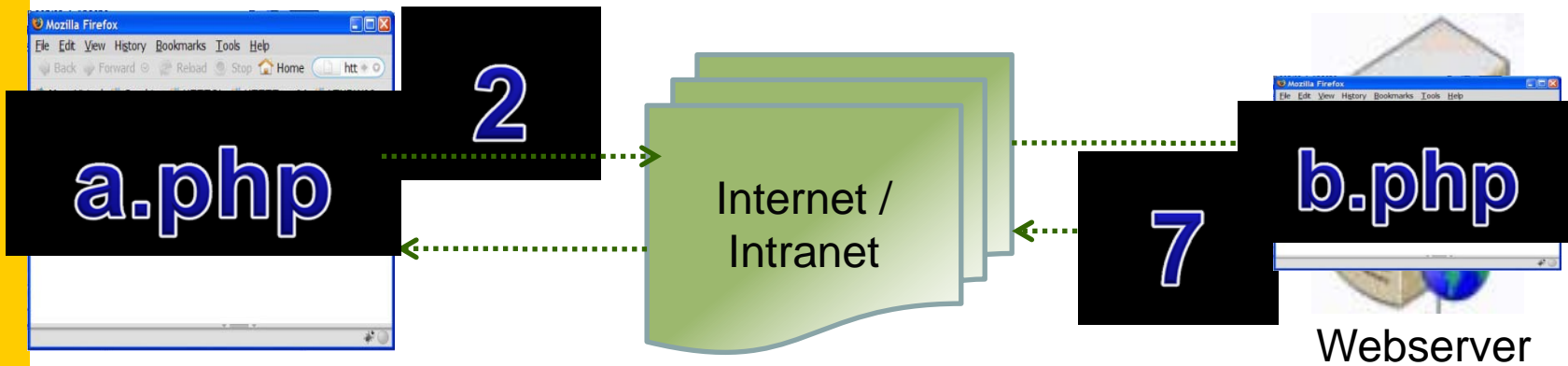
# Cơ chế truyền nhận dữ liệu



THIẾT KẾ VÀ LẬP TRÌNH WEB

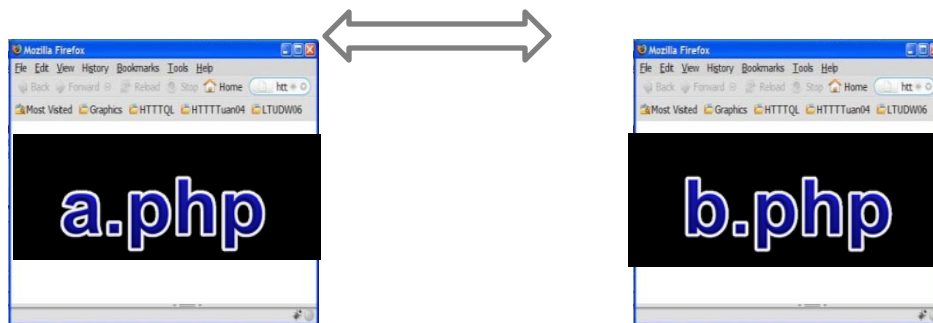
**Web Browser**

www.example.com



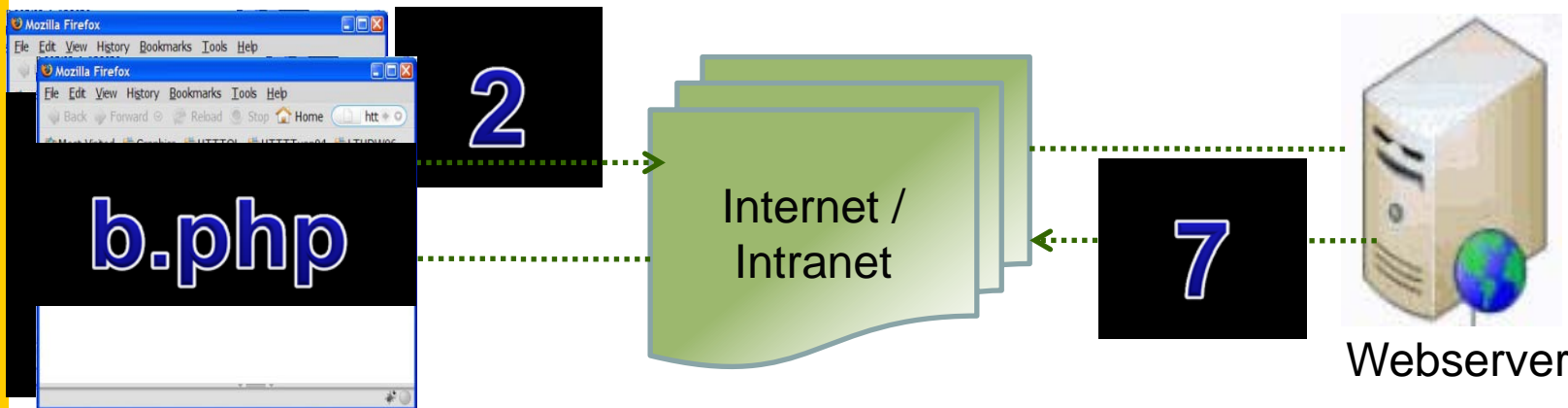


# Cơ chế truyền nhận dữ liệu



**Web Browser**

www.example.com





# Cơ chế truyền nhận dữ liệu



**a.php**



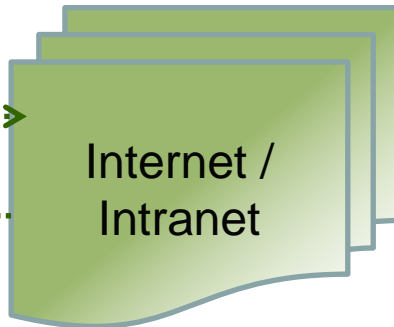
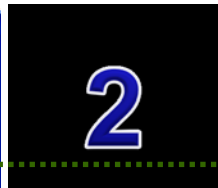
**b.php**



# Cơ chế truyền nhận dữ liệu



## Web Browser



www.example.com



Webserver





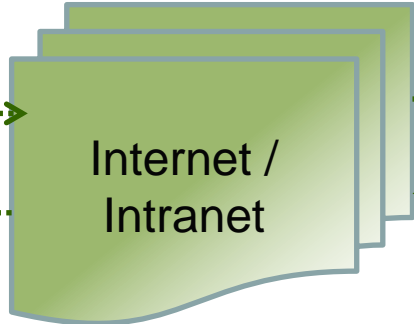
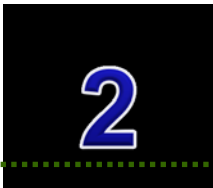
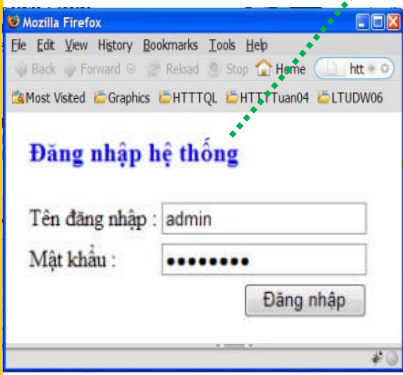
# Cơ chế truyền nhận dữ liệu



Textbox: **txtUsername** = admin

Password Box: **txtPassword** = phpadmin

**Web Browser**



www.example.com

Webserver

Yêu cầu trang **xuly.php**

THIẾT KẾ VÀ LẬP TRÌNH WEB

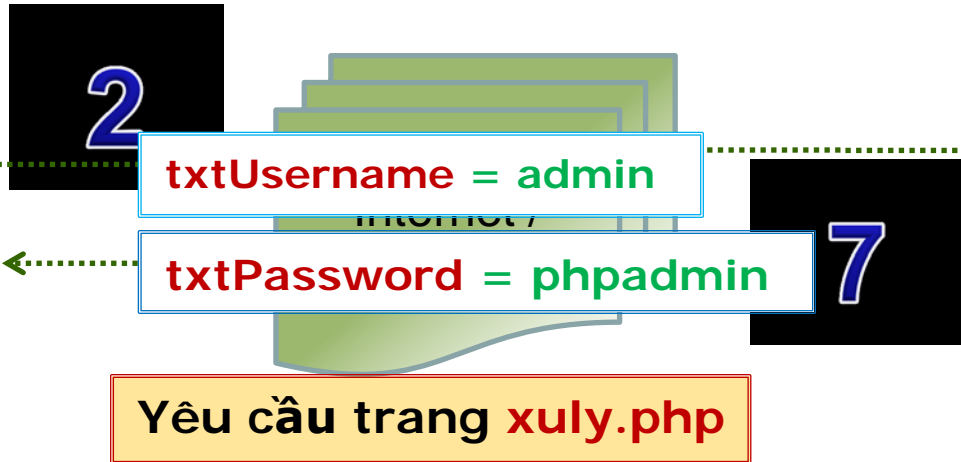
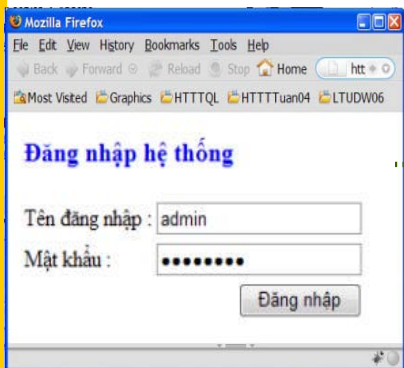


# Cơ chế truyền nhận dữ liệu



THIẾT KẾ VÀ LẬP TRÌNH WEB

## Web Browser



www.example.com



Webserver

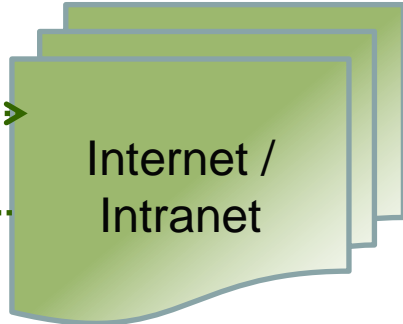
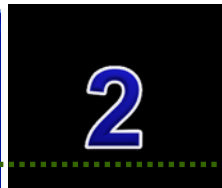


# Cơ chế truyền nhận dữ liệu



THIẾT KẾ VÀ LẬP TRÌNH WEB

## Web Browser

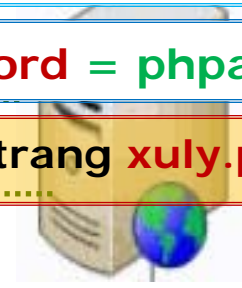


www.example.com

`txtUsername = admin`

`txtPassword = phadmin`

**Yêu cầu trang xuly.php**



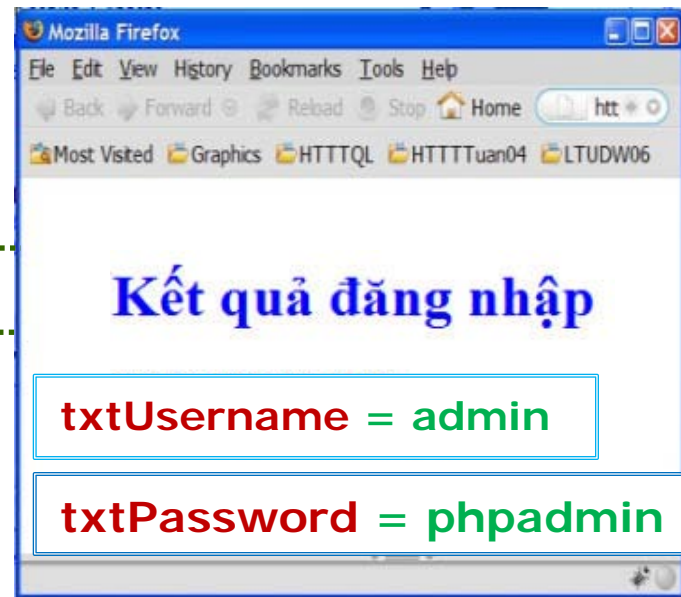
Webserver



# Cơ chế truyền nhận dữ liệu



2







# Cơ chế truyền nhận dữ liệu

➤ Trang web **nhập** dữ liệu

↳ Sử dụng đối tượng `<form>`

↳ Nhập liệu thông qua các `formfield`

↳ Thực hiện việc truyền dữ liệu qua `Submit`

```
<form action="xuly.php" method="post">
```

```
    Username: <input type="text" name="txtname"/>
```

```
    Password: <input type="text" name="password"/>
```

```
    <input type="submit">
```

```
</form>
```



# Cơ chế truyền nhận dữ liệu



- Trang web nhận dữ liệu (**URL**): Sử dụng các biến toàn cục của PHP

–`$_POST`["FieldName"]

–`$_GET`["FieldName"]

–`$_REQUEST`["FieldName"]



# Cơ chế truyền nhận dữ liệu



## timSach.php

```

<h1>Tim sach</h1>
<form action="xltimSach.php" method ="get">
  Keyword: <input type="text" name="txtTukhoa"/>
  <input type ="submit" value="Search">
</form>

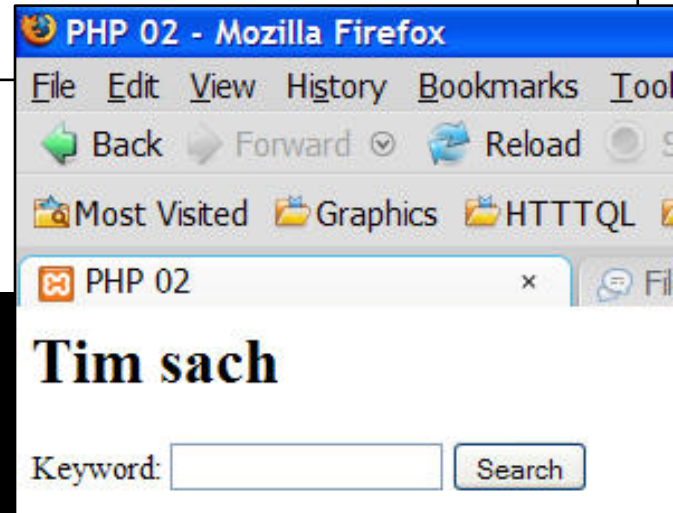
```

## xltimSach.php

```

<?php
  $keyword = $_REQUEST["txtTukhoa"];
  // ...
  <form action="xltimSach.php" method ="get">
    Keyword: <input type="text" name="txtTukhoa" value="<echo $keyword;"/> >
    <input type="submit" value="Search" />
  </form>

```





# Cơ chế truyền nhận dữ liệu



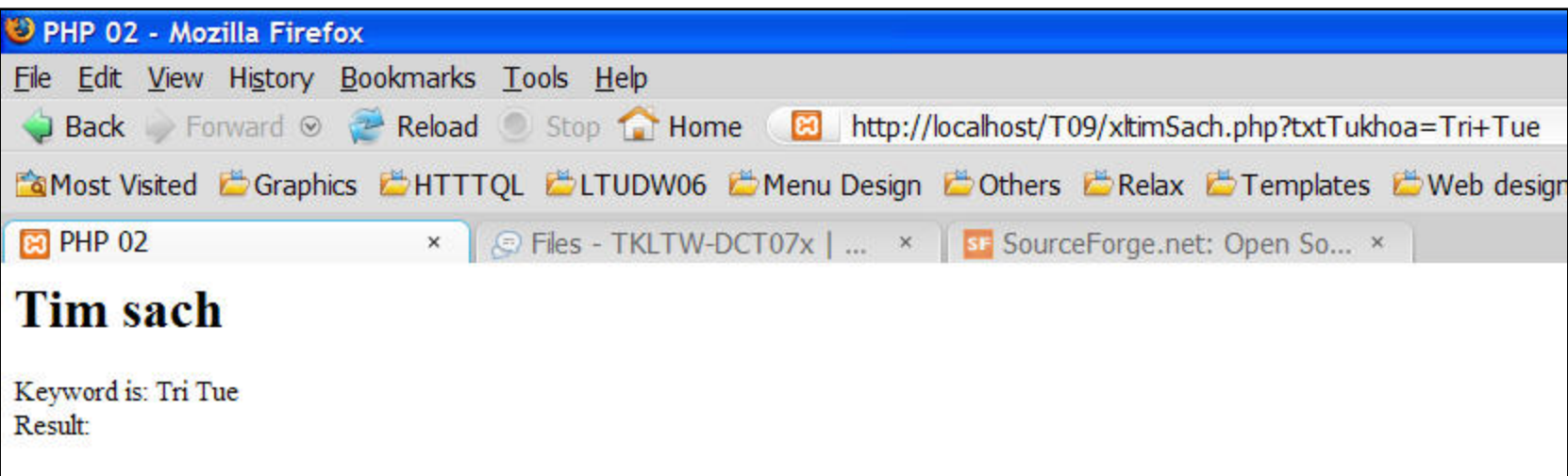
**xltimSach.php**

```

<?php
    $sTukhoa = $_REQUEST["txtTukhoa"];
?>
<h1>Tim sach</h1>
<form action="xltimSach.php" method="get">
    Keyword is: <?php echo $sTukhoa; ?> <br>
    Result:
</form>

```

THIẾT KẾ VÀ LẬP TRÌNH WEB



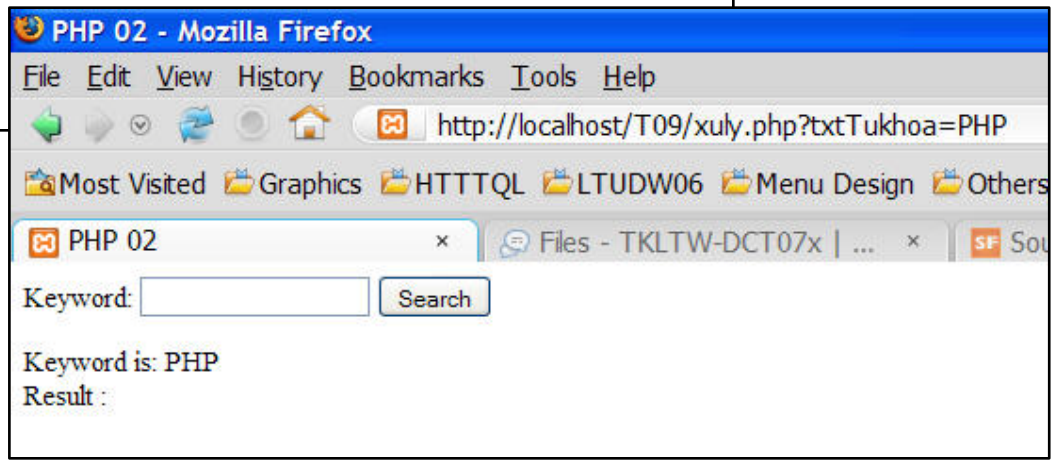
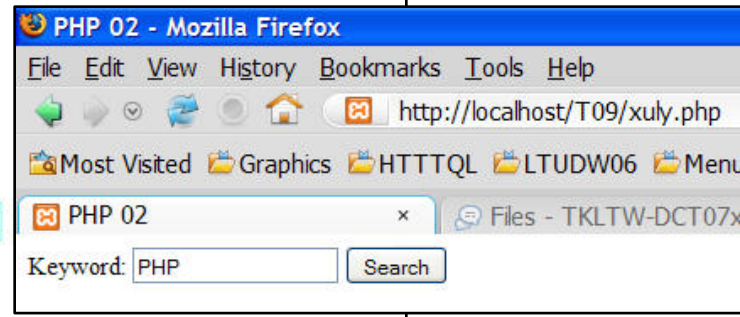


# Ví dụ: xuly.php

```
<form action="xuly.php" Method="GET" >
  Keyword: <input type="text" name="txtTukhoa"/>
  <input type="submit" value="Search"/>
</form>
```

```
<?php
  $sTukhoa = $_REQUEST["txtTukhoa"]
  if (isset($sTukhoa))
  {
    print " Keyword is: $sTukhoa";
    echo "<br> Result :";
  }
?>
```

**xuly.php**





# Phương thức GET



➤ Tham số truyền đi qua địa chỉ URL

↪ <http://domain/pathfile?fieldname1=value1&fieldname2=value2>

↪ [http://localhost/xuly.php?txtTukhoa="Tri%20tue"](http://localhost/xuly.php?txtTukhoa=)

➤ Nhận dữ liệu thông qua các biến toàn cục

↪ `$_GET["FieldName"]`

↪ `$_REQUEST["FieldName"]`



# Phương thức GET



## ➤ Ưu điểm:

↪ Có thể bookmark địa chỉ URL

↪ Có thể giả lập truyền nhận dữ liệu mà không thông qua Form

## ➤ Khuyết điểm:

↪ Không có tính bảo mật

↪ Dung lượng giới hạn

↪ URL submit được lưu lại trên Server





➤ Tham số truyền ẩn bên trong

➤ Nhận dữ liệu thông qua các biến toàn  
cục

↪ `$_POST["FieldName"]`

↪ `$_REQUEST["FieldName"]`





# Phương thức POST



## ➤ Ưu điểm:

↳ Bảo mật

↳ Không giới hạn dung lượng truyền dữ liệu

## ➤ Khuyết điểm:

↳ Trang web trả về không thể bookmark

↳ Có thể bị lỗi khi Back hoặc Refresh (expired)

↳ Security gây khó khăn cho vấn đề truyền dữ liệu



# Một số ví dụ



## Get.html

```
 <br>
<a href='nhandang.php?Ma=LGKF510'>Xem chi tiet</a>
```

## nhandang.php

```
<?php
    echo "Ma nhan dien la: " . $_GET["Ma"];
?>
```

## Get.html



## nhandang.php



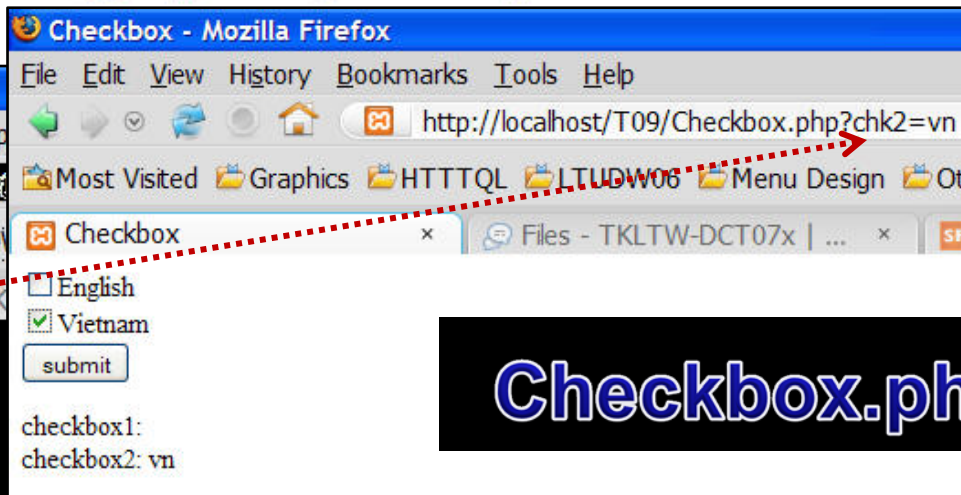
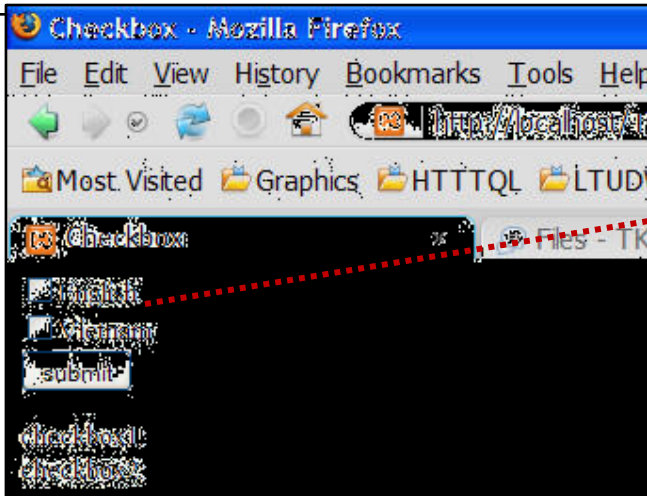


# Một số ví dụ



```
<form method = "GET" action = "Checkbox.php">
  <input type="checkbox" name="chk1" value="en">English
  <br>
  <input type="checkbox" name="chk2" value = "vn">Vietnam
  <br>
  <input type="submit" value="submit"><br>
</form>
<?php
  echo "checkbox1: " . $_REQUEST['chk1'];
  echo ("<br>");
  echo "checkbox2: " . $_REQUEST['chk2'];
?>
```

THIẾT KẾ VÀ LẬP TRÌNH WEB



**Checkbox.php**



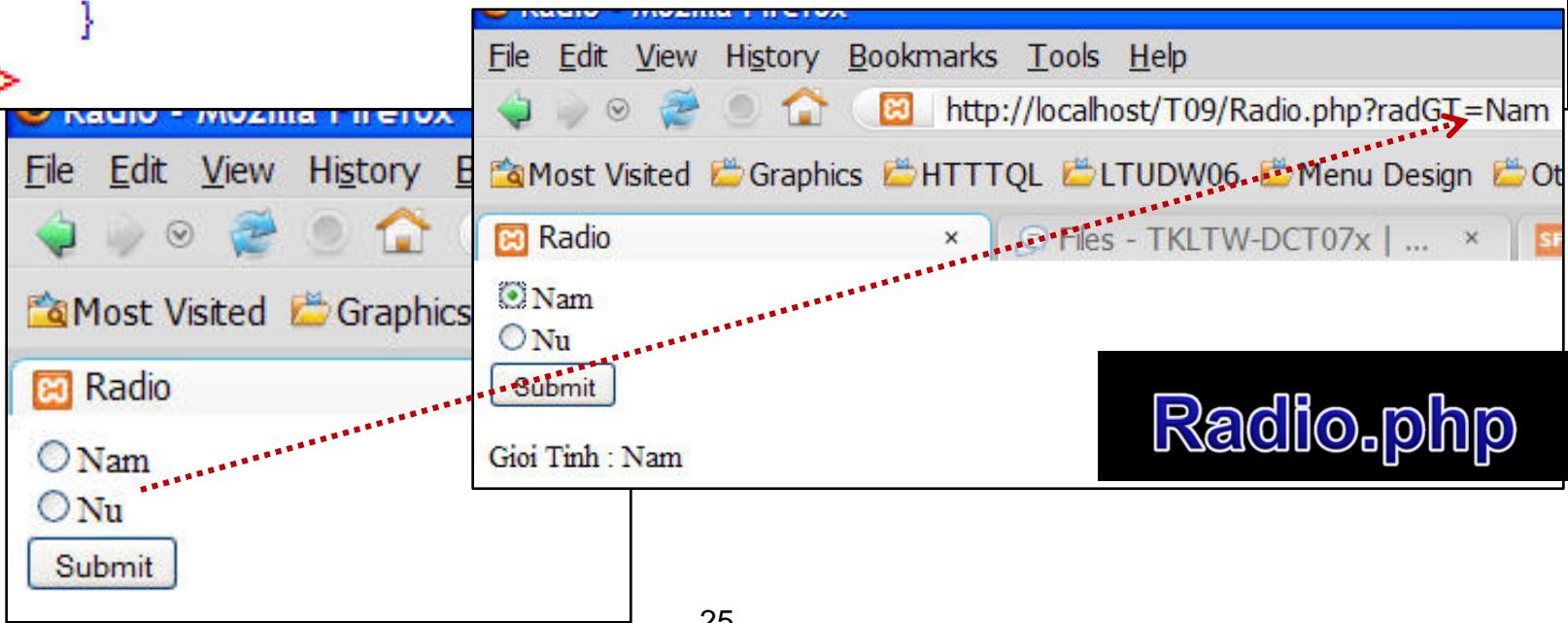
# Một số ví dụ



```

<form action="Radio.php" method="GET">
  <input type="radio" name="radGT" value="Nam">Nam <br>
  <input type="radio" name="radGT" value="Nu">Nu <br>
  <input type="submit" value = "Submit">
</form>
<?php
  if(isset($_GET['radGT']))
  {
    echo "Gioi Tinh : " . $_REQUEST['radGT'];
  }
?>

```



**Radio.php**



# Một số ví dụ



```

<form method="POST" action="combobox.php">
  <select name="Lunch">
    <option value="Pork"> Thit heo </option>
    <option value="Chicken"> Thit ga </option>
    <option> Rau cai </option>
  </select>
  <input type="submit" name="submit" value= "Chon"/>
</form>

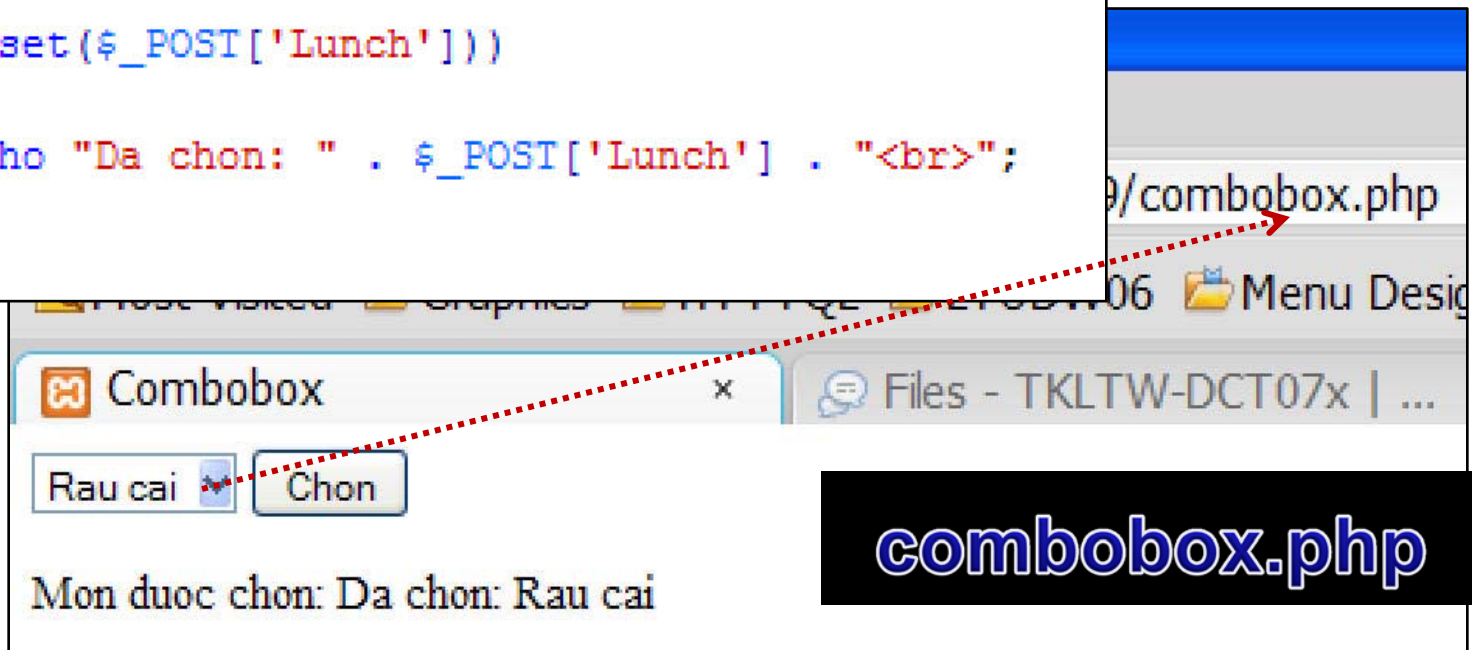
```

Mon duoc chon:

```

<?php
  if (isset($_POST['Lunch']))
  {
    echo "Da chon: " . $_POST['Lunch'] . "<br>";
  }
?>

```



**combobox.php**





# Một số ví dụ

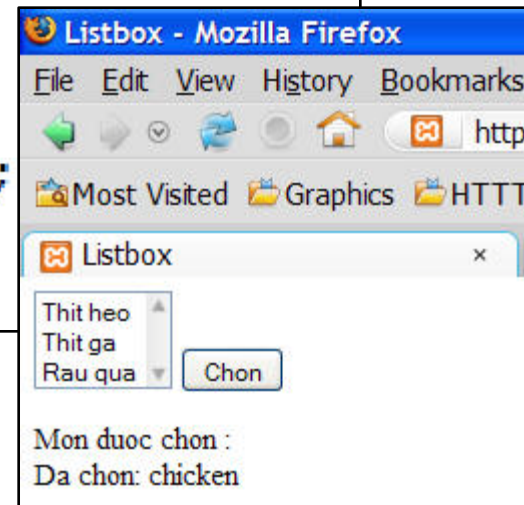


```
<form method="POST" action="Listbox.php">
  <select name="lunch[]" multiple>
    <option value="pork">Thit heo </option>
    <option value="chicken">Thit ga </option>
    <option> Rau qua </option>
  </select>
  <input type="submit" name="submit" value="Chon" />
</form>
```

Mon duoc chon : <br/>

```
<?php
  foreach ($_POST['lunch'] as $choice)
  {
    echo "Da chon: " . $choice . "<br>";
  }
?>
```

## Listbox.php





# Một số ví dụ

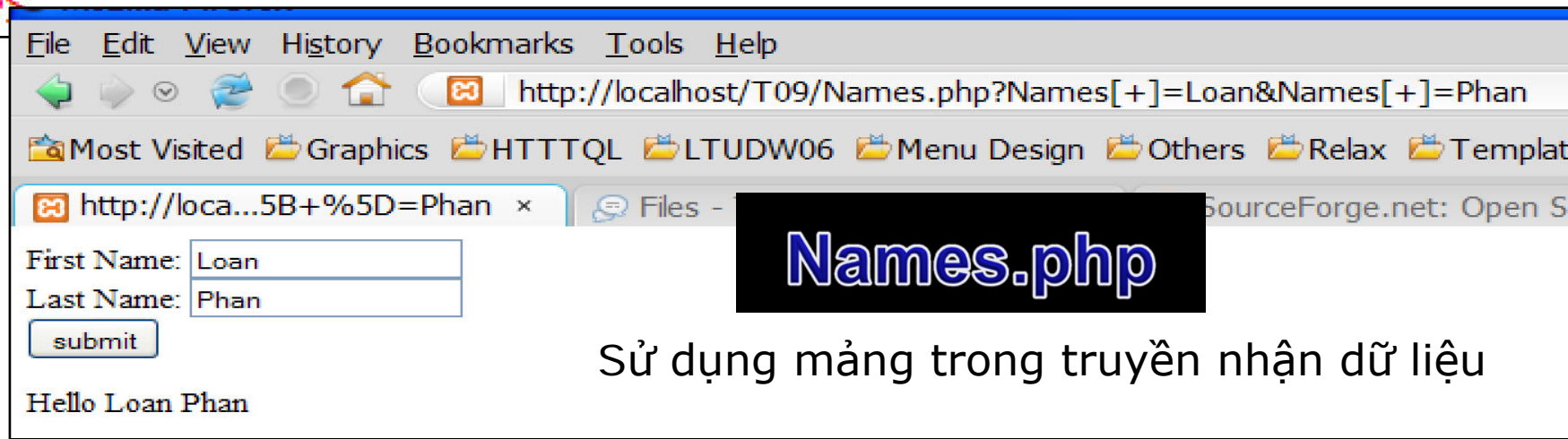


```

<form action="Names.php" name="myform" method="GET">
  First Name: <input type="text" name="Names[ ]" size=20>
  <br>
  Last Name: <input type="text" name="Names[ ]" size=20>
  <br>
  <input type="submit" value="submit">
</form>
<?php
  if (isset($_GET['Names']))
  {
    echo "Hello " . $_GET['Names'][0] . " " . $_GET['Names'][1];
  }

```

THIẾT KẾ VÀ LẬP TRÌNH WEB



Sử dụng mảng trong truyền nhận dữ liệu



# Một số ví dụ



## ➤ <http://localhost/DemoPHP/dangnhap.php>

- ↪ Client gửi dữ liệu bằng phương thức get
- ↪ File gửi dữ liệu tách rời file xử lý & hiển thị kết quả
- ↪ Trong ví dụ có HTML form + JavaScript DOM + PHP

## ➤ <http://localhost/DemoPHP/form.php>

- ↪ Client gửi dữ liệu bằng phương thức post
- ↪ File form vừa gửi dữ liệu vừa xử lý & hiển thị kết quả
- ↪ File form.php là 1 file kết hợp.



# Tham khảo

- <http://www.w3schools.com/php/>
- <http://vn2.php.net/manual/en/langref.php>
- <http://vn2.php.net/manual/en/funcref.php>
- Google ....



## Lập Trình Web

### THỰC HÀNH PHP – Truyền và Nhận dữ liệu

GV: Trần Đình Nghĩa  
tdnghia1977@gmail.com



# Bài thực hành PHP truyền nhận dữ liệu



- Thực hành lại các bài đã Demo.



## Lập Trình Web

### Lập Trình với CSDL

GV: Trần Đình Nghĩa  
tdnghia1977@gmail.com



1. CSDL mySQL - phpMyAdmin
2. Lập trình với CSDL mySQL
3. Kết nối PHP và CSDL
4. Các ví dụ
5. Thực hành



# Sử dụng công cụ phpMyAdmin - Xampp

phpMyAdmin: công cụ dạng web-based để quản lý CSDL mySQL



XAMPP Version 1.7.0 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/xampp/index.php

Most Visited Graphics HTTSQL LTUDW06 Menu Design Others Relax Templates Web design HCI Movie .Net Tutorial Lịch phát sóng kênh St...

web-dct107 | Google Grou... x XAMPP Version 1.7.0 x

## XAMPP for Windows

English / Deutsch / Français / Nederlands / Polski / Italiano / Norwegian / Español / 中文 / Português (Brasil) / 日本語

**XAMPP**  
[PHP: 5.2.8]  
Welcome  
Status  
Security  
Documentation  
Components  
phpinfo()

**Demos**  
CD Collection  
Biorhythm  
Instant Art  
Flash Art  
Phone Book  
Excel\_Writer  
Excel\_Writer  
ADODB

**Specials**  
PHP PostScript  
PHP Paradox

©2002-2006  
...APACHE  
FRIENDS...

**Welcome to XAMPP for Windows Version 1.7.0 !**

**Congratulations:**  
You have successfully installed XAMPP on this system!

Now you can start using Apache and Co. You should first try »Status« on the left navigation to make sure everything works fine.

For OpenSSL support please use the test certificate with <https://127.0.0.1> or <https://localhost>

For this release a special thanks to [Uwe Steinmann](#) for his excellent development and compilation of all current "Special" modules!

Good luck, Kay Vogelgesang + Kai 'Oswald' Seidler

localhost / localhost | phpMyAdmin 3.1.1 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/phpmyadmin/

Most Visited Graphics HTTSQL LTUDW06 Menu Design Others Relax Templates Web design HCI Movie .Net Tutorial Lịch phát sóng kênh St...

web-dct107 | Google Grou... x localhost / localhost | php... x

Server: localhost

Databases SQL Status Variables Charsets Engines Privileges Processes Export Import

**Actions**

**MySQL localhost**

Create new database @

MySQL connection collation: utf8\_general\_ci @

**Interface**

Language @ : English

Theme / Style : Original

Custom color: Reset

Font size: 82%

Please select a database

**MySQL**

- Server: localhost via TCP/IP
- Server version: 5.1.30-community
  - Protocol version: 10
  - User: root@localhost
- MySQL charset: UTF-8 Unicode (utf8)

**Web server**

- Apache/2.2.11 (Win32) DAV/2 mod\_ssl/2.2.11 OpenSSL/0.9.8i mod\_autoindex\_color PHP/5.2.8
- MySQL client version: 5.1.30
- PHP extension: mysql

**phpMyAdmin**

- Version information: 3.1.1
- Documentation
- Wiki
- Official Homepage
- [Changelog] [Subversion] [Lists]

Your configuration file contains settings (root with no password) that correspond to the default MySQL privileged account. Your MySQL server is running with this default, is open to intrusion, and you really should fix this security hole by setting a password for user 'root'.

http://localhost/phpmyadmin/

Open new phpMyAdmin window



# Tạo mới một Database



**Tên CSDL**

**Click để tạo CSDL**

MySQL Workbench - Create New Database dialog box

**Name:** webdb

**Collation:** utf8\_general\_ci

**MySQL connection collation:** utf8\_general\_ci

**Interface**

- Language: English
- Theme / Style: Original
- Custom color:
- Font size: 82%



# Tạo mới một Database



Server: localhost Database: webdb

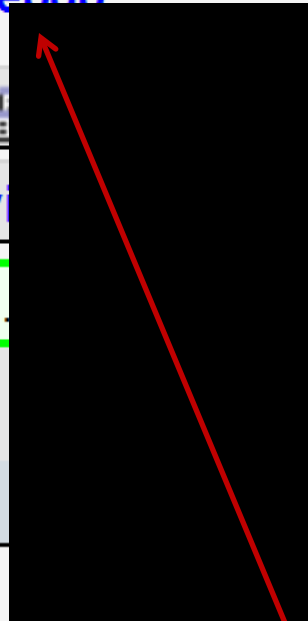
Structure SQL Search Export Import

Designer Operations Priv Stop

✓ Database webdb has been created.

```
CREATE DATABASE `webdb` ;
```

[ Edit ] [ Create PHP Code ]



**Tạo CSDL thành công**

No tables found in database

Create new table on database webdb

Name:

Number of fields:

Go





# Database & Privileges



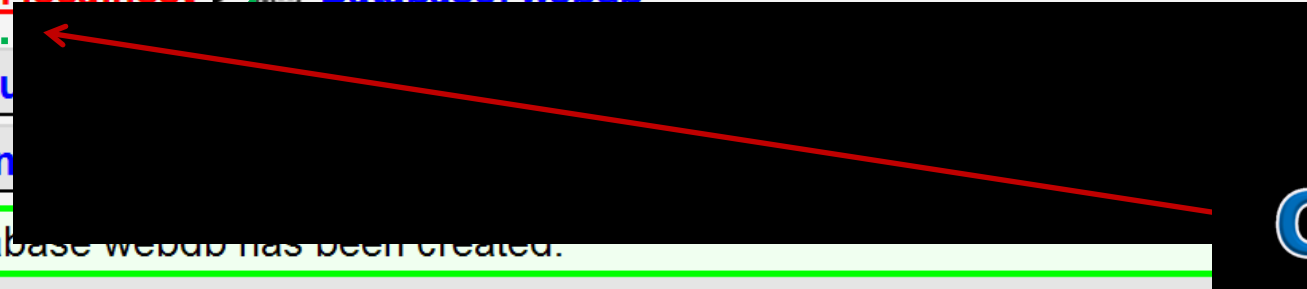
THIẾT KẾ VÀ LẬP TRÌNH WEB

Server: localhost Database: webdb

Structure Design

Database webdb has been created.

```
CREATE DATABASE `webdb` ;
```



Click

Server: localhost

Databases SQL Status Variables

Export Import

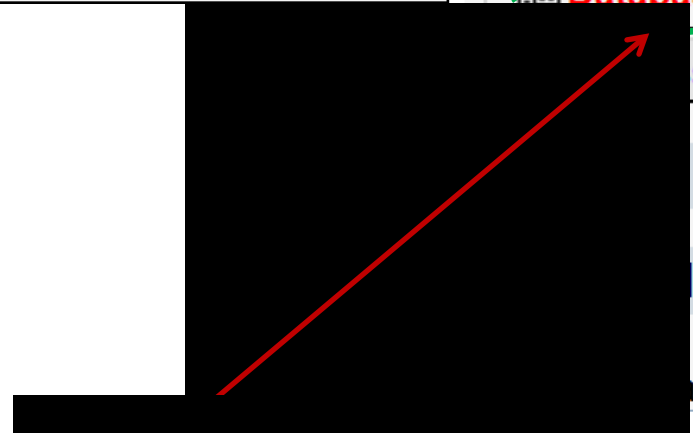
localhost

Create new database ?

Collation

Create

MySQL connection collation: utf8\_general\_ci



Click



# Database & Privileges



THIẾT KẾ VÀ LẬP TRÌNH WEB

Click

Click

Click để sửa hoặc tạo user mới

Mở cửa sổ quản lý user

Total: 7  
Check All / Uncheck All

Server: localhost

Databases SQL Status Chars

Processes Export Import

Users having access to

User	Host	Type	Privileges	Action
root	127.0.0.1	global	ALL PRIVILEGES	
root	localhost	global	ALL PRIVILEGES	



# Database & Privileges



THIẾT KẾ VÀ LẬP TRÌNH WEB

## Change Login Information / Copy User

### Login Information

User name: Use text field:

Host: Local

Password: Do not change the pass

**Đây là user mặc định của Xampp**

## Change Login Information / Copy User

### Login Information

User name: Use text field:

Host: Local

Password: Use text field

Re-type:

Generate Password:

**Thay bằng user mới**

### Create a new user with the same

- ... keep the old one.
- ... delete the old one from the user tables.
- ... revoke all active privileges from the old one and delete it afterwards.
- ... delete the old one from the user tables and reload the privileges afterwards.

**Click để hoàn tất**





# Database & Privileges

## Tạo User thành công

Server: localhost

Databases

SQL

Status

Variables

Charsets

Engines

Privileges

Processes

Export

Import

✓ Your SQL query has been executed successfully

```
CREATE USER 'myUser'@'localhost' IDENTIFIED BY '****';
```

```
GRANT ALL PRIVILEGES ON * . * TO 'myUser'@'localhost' IDENTIFIED BY '****' WITH GRANT OPTION  
MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0 ;
```

[ Edit ] [ Create PHP Code ]

Sau khi có CSDL bạn có thể dùng nó với thông số như sau:

```
$db_host = 'localhost' ;
```

```
$db_name = 'webdb' ;
```

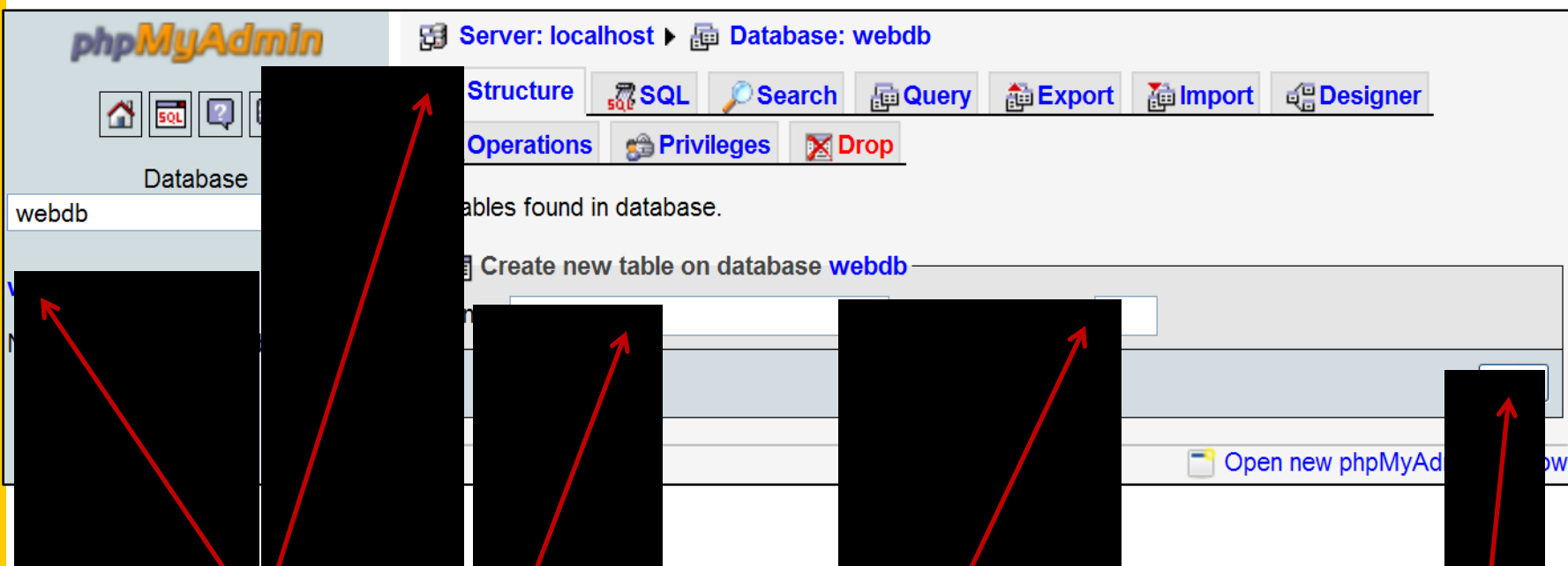
```
$db_user = 'myUser' ;
```

```
$db_pass = '123456' ;
```





# Tạo một Table trong Database



Trong Database

webdb

Structure

Tạo 1 table: User

Có tổng cộng 2 hay 3 trường (fields)

Click GO

THIẾT KẾ VÀ LẬP TRÌNH WEB



# Tạo một Table trong Database



THIẾT LẬP TRÌNH WEB

phpMyAdmin

Database: webdb

webdb (0)  
No tables found in database.

Server: localhost Database: webdb Table: User

Field	Username	Password	Name
Type	VARCHAR	VARCHAR	VARCHAR
Length/Values <sup>1</sup>	128	128	256
Default <sup>2</sup>	None	None	None
Collation			
Attributes			
Null	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Index	PRIMARY	--	--
A_I	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Comments			
MIME type			
Browser transformation			
Transformation options <sup>3</sup>			

Table comments:

Storage Engine: MyISAM

Collation:

PARTITION definition:

Save Or Add 1 field(s) Go

**Thiết lập thuộc tính cho từng trường → Click GO**



# Tạo một Table trong Database



phpMyAdmin

Server: localhost ▶ Database: webdb ▶ Table: User

Field	Type ?	Length/Values <sup>1</sup>	Default <sup>2</sup>	Collation
<input type="text" value="Username"/>	<input type="text" value="VARCHAR"/>	<input type="text" value="128"/>	<input type="text" value="None"/>	<input type="text"/>
<input type="text" value="Password"/>	<input type="text" value="VARCHAR"/>	<input type="text" value="128"/>	<input type="text" value="None"/>	<input type="text"/>
<input type="text" value="Name"/>	<input type="text" value="VARCHAR"/>	<input type="text" value="256"/>	<input type="text" value="None"/>	<input type="text"/>
<input type="text"/>	<input type="text" value="INT"/>	<input type="text"/>	<input type="text" value="As defined:"/>	<input type="text"/>

Database: webdb

webdb (0)  
No tables found in database.

Table comments:

Storage Engine:

Collation:

PARTITION definition:

Or Add  field(s)

THIẾT KẾ VÀ LẬP TRÌNH WEB

**Kiểm tra lại tất cả thông tin của các Fields trong Table.**

**→ Click **SAVE** để hoàn tất quá trình tạo Table.**



# Tạo một Table trong Database



phpMyAdmin

Server: localhost Database: webdb Table: User

MySQL returned an empty result set (0.0003 sec)

```
SELECT *
FROM `User`
LIMIT 0, 30
```

Field	Type	Null	Default	Extra	Action
<input type="checkbox"/> Username	varchar(128)	No	None		[Icons]
<input type="checkbox"/> Password	varchar(128)	No	None		[Icons]
<input type="checkbox"/> Name	varchar(256)	Yes	NULL		[Icons]

Print view Relation view

Add 1 field(s) At End of Table

+ Details...

Open new phpMyAdmin window

THIẾT KẾ VÀ LẬP TRÌNH WEB

Đã hoàn tất quá trình

→ Bước kế tiếp là th

→ Click chọn Tab : **Insert**

Ấu trúc Table : **User**.

liệu vào Table.





# Tạo một Table trong Database

## Nhập dòng dữ liệu đầu tiên trong Table : User.



phpMyAdmin

Server: localhost Database: webdb

Browser Structure SQL Search Export Import Operations Empty Drop

Database: webdb (1)

webdb (1)

user

Field	Type	Function	Value
Username	varchar(128)		
Password	varchar(128)		
Name	varchar(256)		Black Mountain

Ignore

Field	Type	Function	Null	Value
Username	varchar(128)			xyz
Password	varchar(128)			456
Name	varchar(256)		<input checked="" type="checkbox"/>	Piglet

Insert as new row and then Go back to previous page

Restart insertion with 2 rows

**1** Use TAB key to move from value to value, or CTRL anywhere

Open new phpMyAdmin window

**CLICK GO để hoàn tất nhập liệu.**



# Tạo một Table trong Database



THIẾT KẾ VÀ LẬP TRÌNH WEB

[Browse](#) [Structure](#) [SQL](#) [Search](#) [Insert](#) [Export](#) [Import](#) [Operations](#) [Empty](#) [Drop](#)

Showing rows 0 - 1 (2 total, Query took 0.0004 sec)

```

SELECT *
FROM `user`
LIMIT 0 , 30

```

Profiling [ [Edit](#) ] [ [Explain SQL](#) ] [ [Create PHP Code](#) ] [ [Refresh](#) ]

Show : 30 row(s) starting from record # 0  
 in horizontal mode and repeat headers after 100 cells  
 Sort by key: None

+ Options

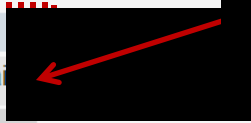
	Username	Password	Name
<input type="checkbox"/>	abc	123	Black Mountain
<input type="checkbox"/>	xyz	456	Piglet

Check All / Uncheck All With selected:

Show : 30 row(s) starting from record # 0  
 in horizontal mode and repeat headers after 100 cells

Query results operations  
[Print view](#) [Print view \(with full texts\)](#) [Export](#) [CREATE VIEW](#)

2 records đã được thêm vào



Dữ liệu đã được thêm vào Table : User.



# Tạo một Table trong Database



Server: localhost Database: webdb Table: user

Search Insert Export Import Operations Empty Drop

Database: webdb (1)

Fields: Username, Password, Name

Run SQL

THIẾT KẾ VÀ LẬP TRÌNH WEB

Có t  
liệu

câu truy vấn tro

Các câu truy vấn mySQL được viết  
RunSQL

Click GO để thực thi câu truy vấn.



# Export & Import Database



THIẾT KẾ VÀ LẬP TRÌNH WEB

The screenshot shows the phpMyAdmin interface for exporting a database. The left sidebar shows the 'Database' dropdown set to 'webdb (1)' and the 'user' table selected. The main area is titled 'View dump (schema) of database' and shows the 'Export' options. A dropdown menu is open, showing 'use' as the selected option. The 'Options' section includes checkboxes for 'Add AUTO\_INCREMENT value', 'Enclose table and field names with backquotes', and 'Add CREATE PROCEDURE / FUNCTION / EVENT'. The 'Add into comments' section is also visible. At the bottom, the 'Save as file' checkbox is checked, and the 'File name template' is set to 'TestDB'. The 'Compression' options are 'None', 'zipped', and 'gzipped'. Red arrows point from text boxes to various elements: 'Chọn Export Database' points to the database dropdown, 'Chọn các phần export' points to the 'use' dropdown, 'Chọn dạng export' points to the 'SQL' radio button, 'Đặt tên file export' points to the 'File name template' field, and 'Click' points to the 'Export' button.

**Chọn Export Database**

**Chọn các phần export**

**Chọn dạng export**

**Đặt tên file export**

**Click**



# Export & Import Database



## Chọn Import Database

Server: localhost Database: TestDB

Structure SQL Search Query Designer Operations Privileges Drop

File to import: [Browse...] (Max: 65,536 KiB)

Location of the file to import: [Browse...]

Character set of the file to import: [Default]

Imported file name: [Default]

Partial import options:

- Allow the import to continue if the file already exists.
- Number of records to import: [All]

Format of the file to import:

- SQL
- CSV
- DELIMITED
- LOCAL INFILE

File Upload dialog:

Look in: T10

- Demo
- ~\$LTW\_T10A
- LTW\_T10A
- T10A

My Recent Documents

Type: Microsoft SQL Server Query File  
Date Modified: 4/13/2009 11:03 PM  
Size: 995 bytes

TestDB

All Files

Open Cancel

THIẾT KẾ VÀ LẬP TRÌNH WEB

Chọn File TestDB.sql

Click



# Export & Import Database



## Hoàn tất quá trình Import Database

Server: localhost Database: TestDB

Structure SQL Search Query Export Import Designer Operations Privileges Drop

Import has been successfully finished, 7 queries executed.

```
-- phpMyAdmin SQL Dump
-- version 3.1.1
-- http://www.phpmyadmin.net
--
-- Host: localhost
-- Generation Time: Apr 13, 2009 at 11:03 PM
-- Server version: 5.1.30
-- PHP Version: 5.2.8
```

File to import

Location of the text file  Browse... (Max: 65,536 KiB)

Character set of the file: utf8

Imported file compression will be automatically detected from: None, gzip, zip

Partial import

Allow the interruption of an import in case the script detects it is close to the PHP timeout limit. This might be good way to import large files, however it can break transactions.

Number of records (queries) to skip from start

Format of imported file

DocSQL

SQL

Options

SQL compatibility mode

Go



1. Kết nối với CSDL
2. Xây dựng câu truy vấn dữ liệu
3. Thực thi câu truy vấn
4. Xử lý kết quả trả về từ câu truy vấn
5. Ngắt kết nối CSDL



# Tạo kết nối CSDL



## ➤ Kết nối mySQL và chọn CSDL

↪ resource `mysql_connect`

([string host], [string username], [string password])

↪ int `mysql_select_db` (string database, [resource connection] )

```
<?php
```

```
$connection = mysql_connect( "localhost" , "myUser" , "123456" );
```

```
mysql_select_db ( "webdb" , $connection );
```

```
?>
```





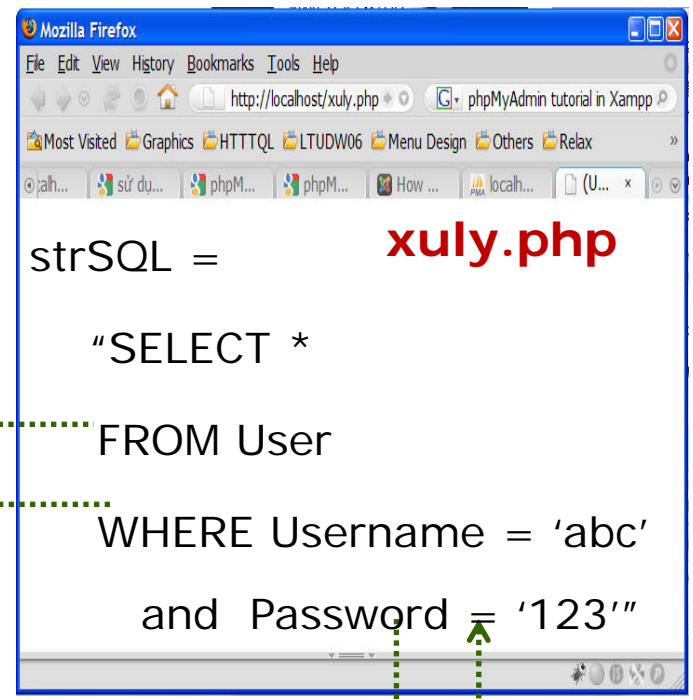
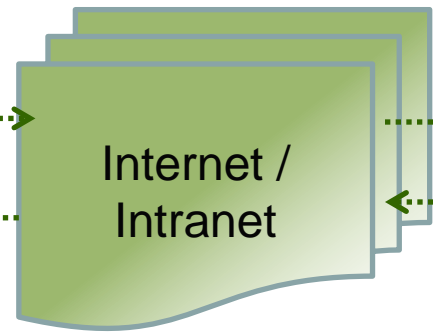
# Xây dựng câu truy vấn



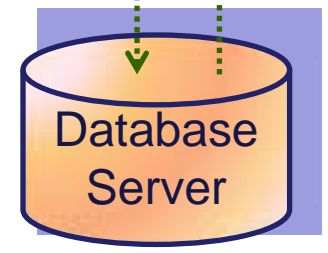
Textbox: **txtUsername** = abc

Password Box: **txtPassword** = 123

**Web Browser**



**Yêu cầu trang xuly.php**



THIẾT KẾ VÀ LẬP TRÌNH WEB



# Xây dựng câu truy vấn (tt)



Thao tác	Câu truy vấn SQL
INSERT	strSQL = "INSERT INTO Users (UserName, Password) VALUES ('abc, '123')"
DELETE	strSQL = "DELETE FROM Users WHERE UserName = 'xyz' "
UPDATE	strSQL = "UPDATE Users SET Password = '456' WHERE UserName = 'abc' "
SELECT	strSQL = "SELECT * FROM Users WHERE UserName = 'abc' "



# Thực thi câu truy vấn

➤ resource `mysql_query`

(string `SQL_command`, [resource `connection`])

```
<?php
```

```
$strSQL = "INSERT INTO Users (UserName, Password)  
          VALUES ('abc', '123')";
```

```
mysql_query($strSQL);
```

```
$strSQL="SELECT * FROM Users";
```

```
$result=mysql_query($strSQL);
```

```
?>
```



# Xử lý kết quả trả về



➤ Hiển thị dữ liệu kết quả trả về:

↪ Sử dụng hàm

- `$row = mysql_fetch_array($result);`

↪ Số lượng field

- `$num = mysql_fetch_fields($result);`

↪ Truy cập đến từng field

- `echo $row["UserName"];` hay  
`echo $row[0];`

# Ngắt kết nối dữ liệu

- int `mysql_close`([resource connection]);
- Tự động thực thi khi kết thúc mã lệnh



# Kết nối các thao tác



<?php

```
//1. Ket noi CSDL
$connction = mysql_connect("hostname","username";"password");
mysql_select_db("database_name",$connection);

//2. Chuan bi cau truy van & 3. Thuc thi truy van
$strSQL = "SELECT * FROM TableName";
$result = mysql_query($strSQL);

//4. Xu ly du lieu tra ve
While ($row = mysql_fetch_array($result))
{
    for ($i = 0; $i < mysql_num_fields($result); $i++)
        echo $row[$i];
}
//5. Dong ket noi
mysql_close($connection);
```

?>



# Các hàm PHP truy xuất CSDL MySQL



- `mysql_connect` : kết nối CSDL MySQL
- `mysql_select_db` : Chọn DB làm việc ( sau khi kết nối CSDL xong)
- `mysql_query`: thực thi câu SQL, trả về true/false hay resource (rds)
- `mysql_close` : ngắt kết nối CSDL
- `mysql_set_charset` : chỉ định sử dụng Unicode với CSDL
- `mysql_real_escape_string` : chuẩn hoá chuỗi PHP trong CSDL
- `mysql_free_result` : giải phóng vùng nhớ sử dụng CSDL
- `mysql_fetch_array` : trả về dòng recordset dưới dạng mảng
- `mysql_fetch_row`
- `mysql_num_rows`
- <http://vn2.php.net/manual/en/book.mysql.php>



# Xử lý lỗi

- int `mysql_error`(resource connection);
- string `mysql_error`(resource connection);

`<?php`

```
function showerror( ) {  
    die("Error" . mysql_errno( ) . ":" . mysql_error( ));  
}
```

```
if (!( $connection =  
    @ mysql_connect( "localhost", "fred", "shhh" )))  
    die("Couldnot connect");
```

```
if (!(mysql_select_db("winestor", $connection)))  
    showerror( );
```

`?>`





# Include File

```
<?php
```

```
    $hostName= "localhost";  
    $databaseName= "bookstoreDB";  
    $username = "root";  
    $password = "root";
```

**db.inc**

```
?>
```

```
<?php
```

```
function showError( )  
{  
    die("Error" . mysql_errno( ) . ":" . mysql_error( ));  
}
```

**error.inc**

```
?>
```

```
<?php
```

```
    include 'db.inc';  
    include 'error.inc';
```

```
?>
```



# Một số ví dụ



## ➤ Tìm kiếm và sắp xếp

↳ Tìm kiếm ([searchbooks.htm](#))

↳ Xuất dữ liệu thoả điều kiện ([showbooks.php](#))

↳ Sắp xếp records theo tên trường ([sortbooks.php](#))

↳ Cho phép người dùng chọn cột sắp xếp  
([selectsortbooks.php](#))

## ➤ Tổng hợp Thêm, Xóa, Sửa

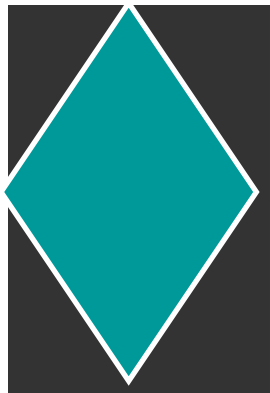
↳ Add ([addbook.php](#))

↳ Delete ([deletebook.php](#))

↳ Edit ([editbook.php](#))

# Tham khảo

- <http://helpdeskgeek.com/how-to/phpmyadmin-how-to/>
- <http://vn2.php.net>
- <http://vn2.php.net/manual/en/book.mysql.php>
- Google ....



## Lập Trình Web

### THỰC HÀNH Lập Trình với CSDL

GV: Trần Đình Nghĩa  
tdnghia1977@gmail.com



# Bài thực hành PHP – Lập trình với CSDL



- Code lại các bài đã minh họa
- Thao tác CSDL bằng công cụ phpMyAdmin



## Lập Trình Web

### PHP Nâng Cao

GV: Trần Đình Nghĩa  
tdnghia1977@gmail.com



1. Upload File
2. Gửi Email
3. Cookies
4. Sessions



# Upload File

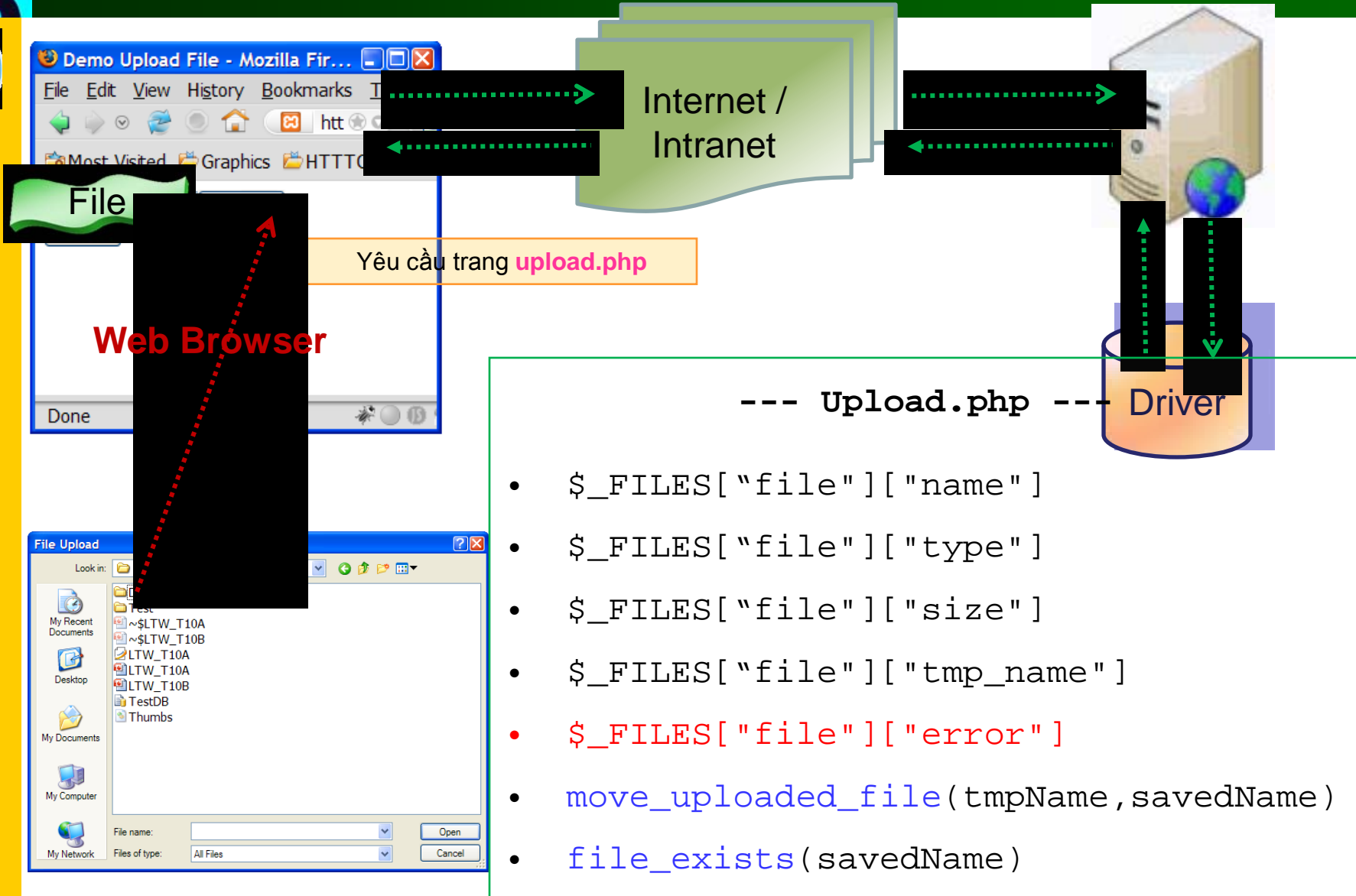


- Cơ chế Upload File
- Thiết kế Form cho phép Upload File
- Xử lý Lưu File upload lên server
- Kiểm tra lỗi, định dạng, kích thước File upload





# Cơ chế Upload File



# Thiết kế Form cho phép Upload File

## ➤ Thiết lập thuộc tính Form

↪ Method : **POST**

↪ Enctype: **multipart/form-data**

```
--- UploadForm.htm ---
```

```
<FORM method="POST" action="upload.php"
      enctype="multipart/form-data">
  <input type="FILE" name = "myfile" ><br>
  <input type="submit" value="Submit">
</FORM>
```



# Xử lý lưu File lên File System trên Server

➤ `move_uploaded_file(tmp_name, saved_name)`

--- Upload.php ---

```
<?php
    move_uploaded_file( $_FILES["file"]["tmp_name"],
                       "upload/" . $_FILES["file"]["name"] );

    echo "Upload: " . $_FILES["file"]["name"] . "<br />";
    echo "Type: " . $_FILES["file"]["type"] . "<br />";
    echo "Size: " . ($_FILES["file"]["size"] / 1024) .
    "Kb<br />";

    echo "Temp. Stored in: " . $_FILES["file"]["tmp_name"];
?>
```



# Kiểm tra lỗi, định dạng và kích thước file



--- Upload.php ---

```
<?php
if (($_FILES["file"]["type"] == "image/gif")
    || ($_FILES["file"]["type"] == "image/jpeg")
    && ($_FILES["file"]["size"] < 20000))
{
    if ($_FILES["file"]["error"] > 0)
    {
        echo "Error Code: " . $_FILES["file"]["error"] .
" <br /> ";
    }
    else{ // Xu ly luu file }
}
?>
```



# Cấu hình phân quyền WRITE cho thư mục Upload



PHP Warning:  
 move\_uploaded\_file(upload/14.jpg)  
 [function.move-uploaded-file]:  
 failed to open stream: Permission  
 denied in .

The screenshot shows the 'upload Properties' dialog box with the 'Security' tab selected. The 'Group or user names' list includes 'CREATOR OWNER', 'Internet Guest Account (HANH-TRAN\USR\_HANH-TRA)', 'Launch IIS Process Account (HANH-TRAN\WAM\_HANH-TRA)', 'SYSTEM', and 'ttbhanh (HANH-TRAN\ttbhanh)'. The 'Internet Guest Account' is selected, and its permissions are shown in the table below.

Permissions for Internet Guest Account	Allow	Deny
Full Control	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Modify	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Read & Execute	<input checked="" type="checkbox"/>	<input type="checkbox"/>
List Folder Contents	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Read	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Write	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Special Permissions	<input type="checkbox"/>	<input type="checkbox"/>

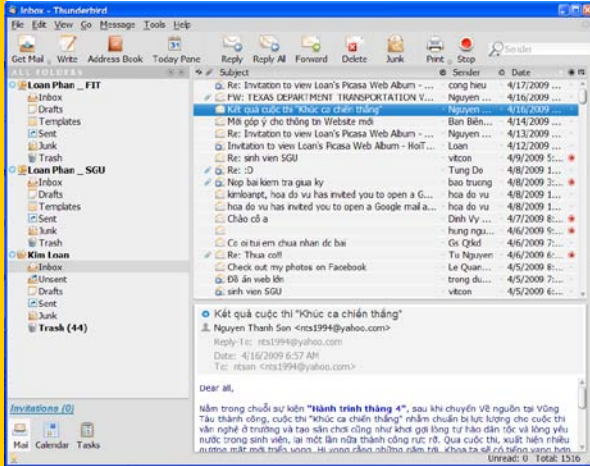
Buttons: Add..., Remove, Advanced, OK, Cancel, Apply.



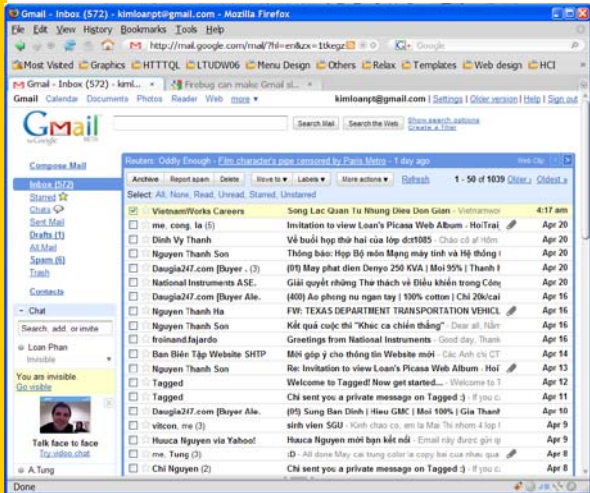
- Cơ chế gửi Email
- Gửi mail sử dụng Outlook
- Gửi mail sử dụng Web Mail
  - ↪ PHP mail() – Non Authentication
  - ↪ PHP PEAR package – SMTP Authentication



# Cơ chế gửi email



Mail Client



THIẾT KẾ VÀ LẬP TRÌNH WEB

Internet / Intranet



Account Name:

**Default Identity**

Each account has an identity, which is the information that other people see your messages.

Your Name:

Email Address:

Reply-to Address:

Organization:

Attach this signature:  
E:\LoanPhan\GMail\ThunderbirdPortable\Sign.txt

Attach my vCard to messages

Outgoing Server (SMTP):

mail(to,subject,message...)

Internet / Intranet



Web Server



# Ví dụ cấu hình gửi email sử dụng GMAIL



THIẾT KẾ VÀ LẬP TRÌNH WEB

Gmail - Settings - kimloanpt@gmail.com - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://mail.google.com/mail/?hl=en&zx=1tkegz

Most Visited Graphics HTTTQL LTUDW06 Menu Design Others Relax Templates Web design HCI

Gmail - Settings - kimloanp... x Firebug can make Gmail sl... x

Gmail Calendar Documents Photos Reader Web more v kimloanpt@gmail.com | Settings | Older version | Help | Sign out

**Gmail** by Google BETA

Compose Mail

Inbox (572)

Starred ☆

Chats

Sent Mail

Drafts (1)

All Mail

Spam (6)

Trash

Contacts

- Chat

Search, add, or invite

Loan Phan Invisible

You are invisible. Go visible

Talk face to face Try video chat

Thuy Nguyen

Done

### Settings

General Accounts Labels Filters **Forwarding and POP/IMAP** Chat Web Clips Labs Themes

**Forwarding:**

- Disable forwarding
- Forward a copy of incoming mail to  and

Tip: You can also forward only some of your mail by [creating a filter!](#)

**POP Download:** [Learn more](#)

- Status: POP is enabled** for all mail that has arrived since 7/30/08
  - Enable POP for **all mail** (even mail that's already been downloaded)
  - Enable POP for **mail that arrives from now on**
  - Disable POP
- When messages are accessed with POP**
- Configure your email client** (e.g. Outlook, Eudora, Netscape Mail) [Configuration instructions](#)

**IMAP Access:** [Learn more](#)

- Status: IMAP is enabled**
  - Enable IMAP
  - Disable IMAP
- Configure your email client** (e.g. Outlook, Thunderbird, iPhone) [Configuration instructions](#)

Save Changes Cancel





# Gửi mail sử dụng cấu hình trong Mail-Client

## ➤ Sử dụng mailto

```
<A href="mailto:abc@yahoo.com?SUBJECT=Question&BODY=Plz answer">
```

## ➤ Format: `mailto:address[?key=setting][&key=setting] ...`

Key	Ý nghĩa
SJT	Tiêu đề Email
CC	Danh sách địa chỉ Email người nhận, mỗi địa chỉ cách nhau bằng dấu chấm phẩy (;).
BCC	Danh sách địa chỉ Email người nhận được ẩn mỗi địa chỉ cách nhau bằng dấu chấm phẩy (;).
BODY	Nội dung Email.



# Sử dụng PHP mail() – Non Authentication

➤ `mail(to,subject,message,headers,parameters)`

Key	Specifies
TO	Địa chỉ email người nhận
SUBJECT	Tiêu đề Email (KHÔNG được chứa ký tự xuống dòng)
MESSAGE	Nội dung Email
HEADERS	Thông tin thêm (vd: FROM, BCC, CC,...) Các thông tin này nên phân cách nhau bằng ký tự xuống dòng (\r\n)
PARAMETERS	Tham số cấu hình cho ứng dụng gửi mail



# Cấu hình chuẩn của PHP mail() – Non Authentication

- Cấu hình mặc định được định nghĩa trong file **php.ini**

	Default	Ý nghĩa
SMTP	localhost	DSN hoặc IP Address của SMTP Server
smtp_port	25	Port của SMTP
sendmail_from	NULL	Địa chỉ người gửi
sendmail_path	NULL	Đường dẫn đến ứng dụng gửi mail (UNIX)



# Ví dụ



```
<?php
    // định nghĩa email người nhận
    $to = 'noname@yahoo.com';
    // tiêu đề
    $subject = 'Test email';
    // nội dung email
    $message = "Hello World!\n\nThis is my first mail.";
    // định nghĩa email người gửi và email trả lời
    $headers = "From: mymail@yahoo.com\r\n" .
        "Reply-To: replymail@yahoo.com";
    // gửi email
    $mail_sent = @mail( $to, $subject, $message, $headers );
    // kiểm tra gửi thành công và thông báo
    echo $mail_sent ? "Mail sent" : "Mail failed";
?>
```



# Gửi mail sử dụng PHP Pear package – SMTP Authentication



## 1. Sử dụng PEAR package Mail.php

```
↳ require_once('Mail.php');
```

## 2. Khởi tạo thể hiện để gửi mail sử dụng Mail::factory()

```
↳ $smtp = Mail::factory('smtp', $params);
```

- \$params["host"] = "hostname";
- \$params["auth"] = true;
- \$params["username"] = "smtp\_username";
- \$params["password"] = "smtp\_password";

## 3. Gửi mail sử dụng hàm send()

```
↳ $mail = $smtp->send($to, $headers, $message)
```

- \$headers['From'] = 'from@example.com';
- \$headers['To'] = \$to;
- \$headers['Subject'] = 'Test message';
- \$headers['Cc'] = 'cc@example.com';
- \$headers['Reply-To'] = 'from@example.com';

## 4. Kiểm tra lỗi gửi mail

```
↳ if (PEAR::isError($mail))  
    echo($mail->getMessage());
```



# Ví dụ



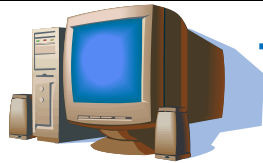
```
<?php
    require_once "Mail.php";

    $host = "smtp.gmail.com";
    $username = "ttbhanh@gmail.com";
    $password = "smtp_password";
    $to = "noname@yahoo.com";
    $headers = array ('From' => 'ttbhanh@gmail.com',
                     'Subject' => 'Test send mail');
    $smtp = Mail::factory('smtp', array ('host' => $host,
                                        'auth' => true,
                                        'username' => $username,
                                        'password' => $password));
    $mail = $smtp->send($to, $headers, 'this is my message');

    if (PEAR::isError($mail))
        echo($mail->getMessage());
    else
        echo("<p>Message successfully sent!</p>");
?>
```



# Cookies



*Client*



**setcookie**



*Webservice*

**\$\_COOKIE**

- Là 1 tập tin được server lưu xuống máy của client
- Mỗi lần client gửi request 1 trang web, đồng thời sẽ gửi file cookie này lên server
- Các điều khiển xử lý thông tin (lưu, lấy) do server thực hiện
- Thường được sử dụng để lưu thông tin cá nhân của client



## ➤ Lệnh ghi cookie

↪ `setcookie(name, value, expire, path, domain);`

↪ `setrawcookie(name, value, expire, path, domain);`

- Name: tên cookie
- Value: giá trị cookie
- Expire: ngày/khoảng thời gian hết hạn hết hạn
- Path : đường dẫn mà cookie được gửi đi
- Domain : xác định tên miền mà cookie được gửi đi
- *Bắt buộc phải xuất hiện trước thẻ <html>*

## ➤ Lấy giá trị cookie

↪ `echo $_COOKIE["cookieName"];`

## ➤ Xóa cookie

↪ `setcookie("cookieName", "", time()-3600);`





# Ví dụ cookies

- Lưu thông tin đăng nhập



Username:

Password:

Saved Username & Password

Login

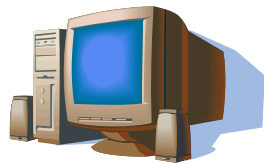
`$_COOKIE[fieldName]`

Username:

Password:

Saved Username & Password

Login



**Client**

Lưu username & password



setcookie



**Webservice**



# Session



Cookie: PHPSESSID

`$_SESSION[sessname]`



- Là thông tin về client được server lưu trên máy của server
- Sử dụng cookie lưu định danh duy nhất cho từng client
- Mục đích lưu biến dữ liệu dùng chung cho nhiều trang trong 1 phiên làm việc của client



# Sử dụng session



## ➤ Khởi động Session

↪ `session_start();`

- *Bắt buộc phải xuất hiện trước thẻ `<html>`*

## ➤ Ghi & Đọc giá trị Session

↪ `$_SESSION["sessionName"] = value`

↪ `$_SESSION["sessionName"][ ] = array(...)`

↪ `if (isset($_SESSION["sessionName"]))  
    echo $_SESSION["sessionName"];`

## ➤ Hủy Session

↪ `session_destroy();`



# Cấu hình session

- Cấu hình trong php.ini
- `ini_set('parameter', value);`
  - ↪ Gọi trước hàm `session_start();`

Parameter	Default	
<code>session.auto_start</code>	0	Không tự động
<code>session.cookie_lifetime</code>	0	Tồn tại cho đến khi đóng browser
<code>session.name</code>	PHPSESSID	Tên biến cookie
...		



# Ví dụ session

- Đếm số lần duyệt 1 trang web của 1 user

```
<?php  
  
session_start( );  
  
if (isset($_SESSION["count"]))  
    $_SESSION["count"] = $_SESSION["count"] + 1;  
else  
    $_SESSION["count"] = 1;  
  
print "You've looked at this page " .  
    $_SESSION['count'] . ' times.';  
  
?>
```



# Ứng dụng session



- Làm thế nào để ngăn không cho người dùng truy cập vào các trang web nếu chưa đăng nhập?
- Ý tưởng: Dùng các biến Session để lưu trạng thái đăng nhập của người dùng:
  - ↪ `$_SESSION["IsLogin"] = 0/1` : Lưu trạng thái đăng nhập
  - ↪ `$_SESSION["Username"]` : Lưu Tên đăng nhập
  - ↪ `$_SESSION["Authentication"]` : Lưu Loại quyền đăng nhập
  - ↪ ...



# Bắt người dùng truy cập web sau đăng nhập



1. Tạo trang **login.htm** yêu cầu người dùng đăng nhập.
2. Tạo trang **validateuser.php** là trang xử lí thông tin đăng nhập từ trang login
  - ↪ Kết nối với CSDL, kiểm tra xem thông tin đăng nhập có đúng hay không.
  - ↪ Nếu không đúng thì cho redirect đến trang login.htm.
  - ↪ Còn nếu ĐÚNG thì dùng một biến Session để lưu trạng thái login thành công lại.
    - Ví dụ `$_SESSION["IsLogin"] = 1`.
    - Lưu ý là phải đặt giá trị mặc định cho biến này là 0 khi khởi tạo một Session.
3. Tạo trang **logout.php** là trang xử lý khi người dùng logout
  - Reset trạng thái login là chưa đăng nhập



# Bắt người dùng truy cập web sau đăng nhập



- Trong tất cả các trang muốn bảo mật, phải thêm đoạn mã sau để kiểm tra người dùng đã login hay chưa, nếu chưa thì redirect lại trang login.htm

```
<?php
    session_start();

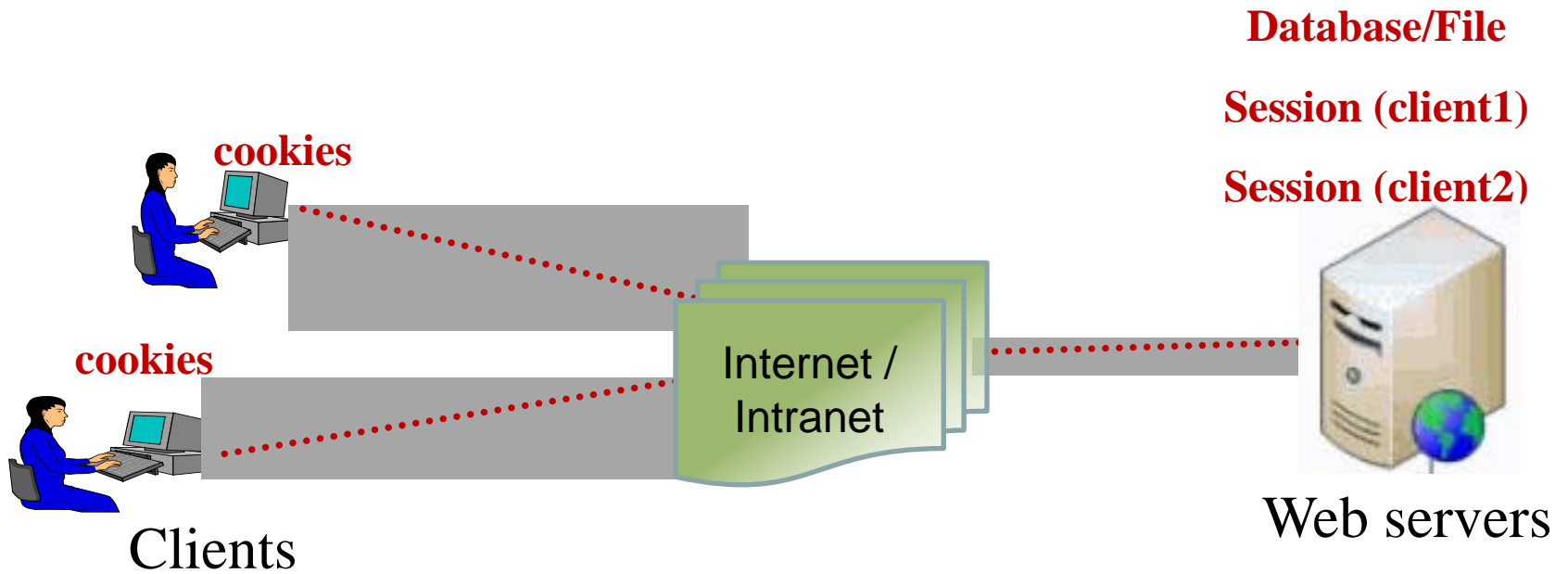
    if ( $_SESSION["IsLogin"] != 1 )
        header( "Location: login.htm" );

?>
```





# So sánh



Thông tin chia sẻ	Vị trí lưu trữ	Thời gian tồn tại	Số client	Số trang web sử dụng
Database/File	Server	Dài	N	N
Session	Server	Ngắn	1	N
Cookies	Client	Dài	1	N



# Tham khảo

- PHP Cookbook, 2nd Edition, David Sklar & Adam Trachtenberg, O'Reilly, 2006
- Learning PHP and MySQL, Michele Davis & Jon Phillips, O'Reilly, June 2006
- Learning PHP 5, David Sklar, O'Reilly, June 2004
- [http://www.w3schools.com/JS/js\\_cookies.asp](http://www.w3schools.com/JS/js_cookies.asp)
- <http://www.quirksmode.org/js/cookies.html>
- Google ....



## Lập Trình Web

### THỰC HÀNH PHP Nâng Cao

GV: Trần Đình Nghĩa  
tdnghia1977@gmail.com

# Bài thực hành PHP – Lập trình với CSDL

- Xem demo một số ví dụ
- Code lại các bài đã minh họa



## Lập Trình Web

### Kỹ thuật giỏi hàng trong PHP

GV: Trần Đình Nghĩa  
tdnghia1977@gmail.com



# Nội dung

- Cấu trúc dữ liệu & mô hình hoạt động giỏ hàng
- Thêm sản phẩm vào giỏ hàng
- Hiện thị giỏ hàng
- Sửa số lượng sản phẩm
- Xóa một sản phẩm hay Xóa toàn bộ giỏ hàng
- Ví dụ
- Lưu giỏ hàng vào GSDL



# Cấu trúc dữ liệu cho giỏ hàng

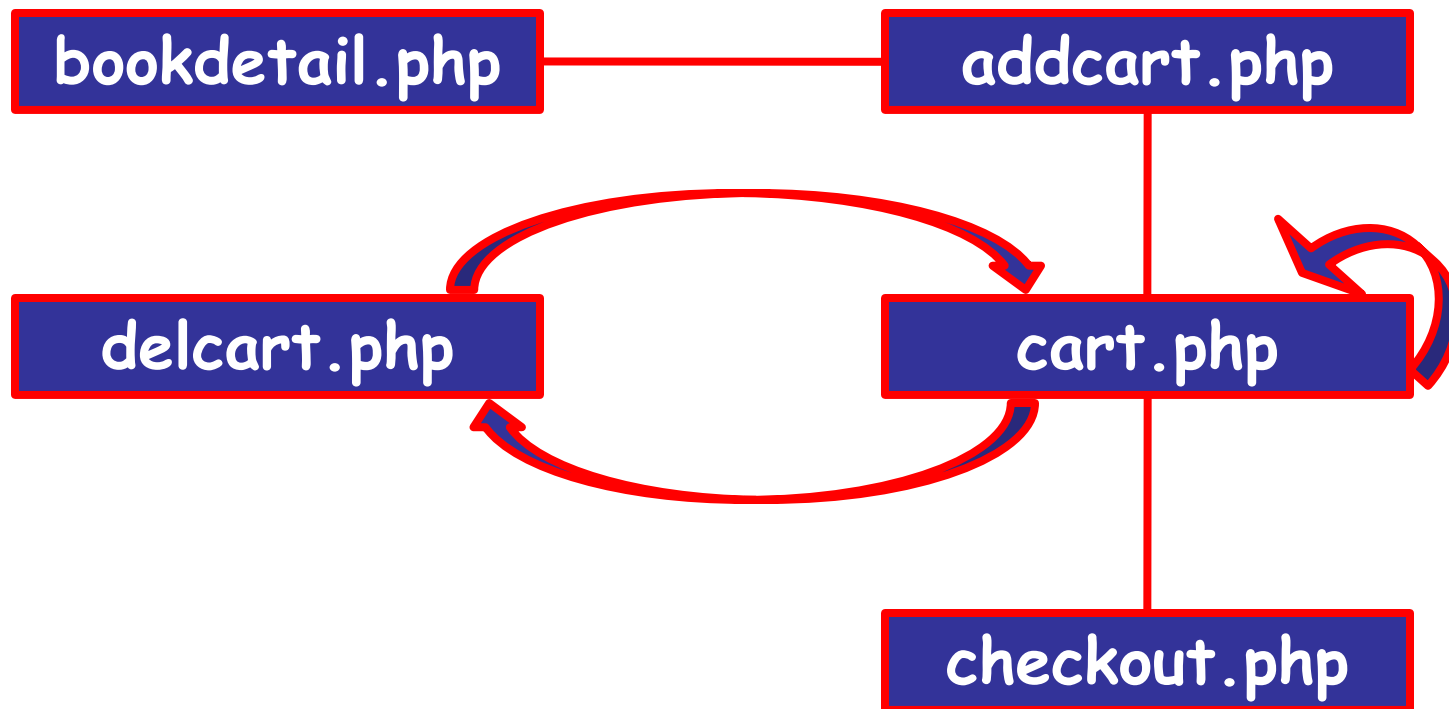


- Mỗi khách truy cập website sẽ có 1 session riêng biệt
- Lưu toàn bộ giỏ hàng vào session thông qua biến mảng cart
  - ↪ Mỗi phần tử của cart sẽ được truy xuất theo dạng id ==> value.  
Với: id = mã sách, value = số lượng

```
<?php  
session_start();  
session_register("cart");  
?>
```



# Mô hình hoạt động của giỏ hàng







# Thêm sản phẩm vào giỏ hàng



- Khi khách chọn đưa 1 sản phẩm vào giỏ hàng
  - ↪ `$_SESSION["cart"]` sẽ được kiểm tra và tạo nếu chưa có.
  - ↪ Nếu `$_SESSION["cart"]["id"]` chưa có:  
`$_SESSION["cart"]["id"] = 1`
  - ↪ Nếu `$_SESSION["cart"]["id"]` đã có:  
`$_SESSION["cart"]["id"] ++`



# Thêm sản phẩm vào giỏ hàng



```
bookdetail.php x addcart.php x
Code Split Design Live View Live
1 <?
2 session_start();
3 session_register("cart");
4 $id=$_REQUEST['item_id'];
5 if(isset($_SESSION['cart'][$id]))
6 {
7     $qty = $_SESSION['cart'][$id] + 1;
8 }
9 else
10 {
11     $qty=1;
12 }
13 $_SESSION['cart'][$id]=$qty;
14 header("location:cart.php");
15 exit();
16 ?>
```



# Hiển thị giỏ hàng (cart.php)

- Trước tiên, ta cần kiểm tra giỏ hàng có hàng?

```
<?php
$ok=0;
if(isset($_SESSION['cart']))
{
    foreach($_SESSION['cart'] as $k => $v)
    {
        if(isset($k))
        {
            $ok=1;
        }
    }
}
if($ok == 0)
{
    echo "<div class=pro>";
    echo '<p align=center>Ban khong co mon hang nao trong gio hang<br />
|<a href=index.php>Buy Ebook</a></p>';
    echo "</div>";
}
else
{
    ?>
```



# Hiển thị giỏ hàng (cart.php)

- Nếu có, ta xây dựng câu lệnh SQL để lấy ra các sản phẩm có trong giỏ hàng từ CSDL

```
<?php
foreach($_SESSION['cart'] as $key=>$value)
{
    $item[]=$key;
}
$str=implode(",",$item);
$conn=mysql_connect("localhost","root","") or
    die("Can not connect database");
mysql_select_db("bookstore",$conn);
session_start();
$sql="SELECT SACH.*, TENTL FROM SACH, THELOAI WHERE
    |SACH.MATL=THELOAI.MATL AND MASACH IN (" . $str . ")";
$query=mysql_query($sql);
$tong = 0;
while($row=mysql_fetch_array($query))
{
```



# Hiển thị giỏ hàng (cart.php)

- Hiển thị từng sản phẩm trong giỏ hàng cùng với số lượng đặt mua của chúng thông qua vòng lặp while

```
while($row=mysql_fetch_array($query))
{
    echo('<tr>
    <td><a href=bookdetail.php?masach=' . $row[MASACH] . '>' . $row[TENSACH] . '</td>
    <td>' . $row[GIA] . '</td>
    <td><input type=text name=qty[' . $row[MASACH] . '] value="" .
        $_SESSION["cart"][$row[MASACH]] . '></td>
    <td>' . $_SESSION["cart"][$row[MASACH]]*$row[GIA] . '</td>
    <td><a href=delcart.php?masach=' . $row[MASACH] . '>Xoa</a></td>
    </tr> ');
    $tong += $_SESSION['cart'][$row[MASACH]]*$row[GIA];
}
?>
```



# Hiển thị giỏ hàng (cart.php)



- Sau vòng lặp, hiển thị tổng tiền giỏ hàng và các liên kết cần thiết

```
echo('<tr colspan=5 style="text-align:right">
    <td>
    <b>To`ng tie`n gio hang:  ' . $tong . '</b>
    </td>
</tr></table>');
echo "<input type='submit' name=submit value='Cap Nhat Gio Hang'>";
echo "<b><a href=index.php>Mua Sach Tiep</a> -
    <a href=delcart.php?masach=0>Xoa Bo Gio Hang</a> -
    <a href=checkout.php>Check out</a></b>";
echo "</div></form>";
```



# Sửa số lượng sản phẩm



- Số lượng mỗi sản phẩm trong giỏ hàng được hiển thị bằng mảng TextField có tên qty[]
- Khi form trong cart.php được submit (Button submit có tên submit), các số lượng mới sẽ được chuyển cho chính trang này
- Như vậy, vào **đầu trang cart.php**, cần kiểm tra nếu đang được submit, ta sẽ cập nhật số lượng mới
- Việc cập nhật số lượng mới chỉ đơn giản là ghi đè số lượng mới vào số lượng cũ. Tuy nhiên, nếu số lượng mới = 0, ta sẽ xóa sản phẩm này khỏi giỏ hàng



# Sửa số lượng sản phẩm

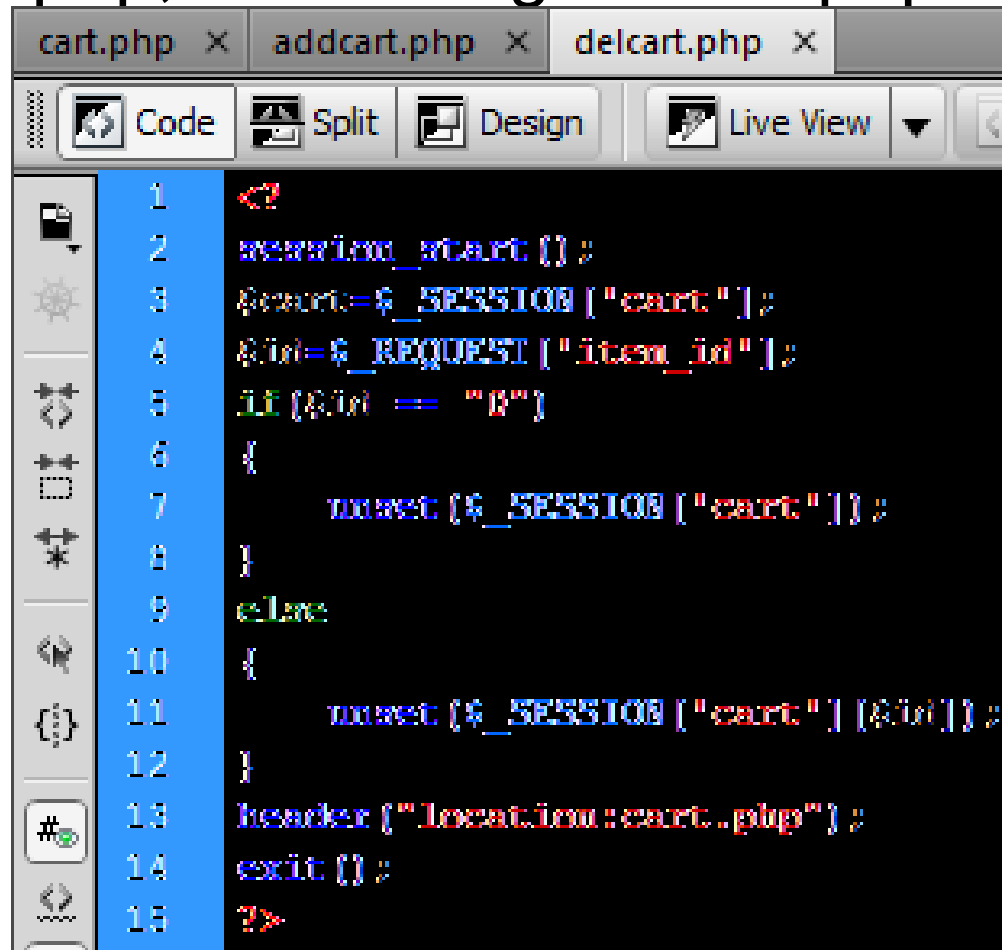


```
<?php
session_start();
if(isset($_POST['submit']))
{
    foreach($_POST['qty'] as $key=>$value)
    {
        if( ($value == 0) and (is_numeric($value)))
        {
            unset ($_SESSION['cart'][$key]);
        }
        elseif(($value > 0) and (is_numeric($value)))
        {
            $_SESSION['cart'][$key]=$value;
        }
    }
    header("location:cart.php");
}
?>
```



# Xóa một sản phẩm hay Xóa toàn bộ giỏ hàng

- Xem xét link *Xóa* tương ứng với từng sản phẩm hay link *Xóa Bỏ Giỏ Hàng* trong `cart.php`, ta có trang `delcart.php` như sau:



```
1 <?
2 session_start();
3 $sort=$_SESSION["cart"];
4 $id=$_REQUEST["item_id"];
5 if($id == "0")
6 {
7     unset($_SESSION["cart"]);
8 }
9 else
10 {
11     unset($_SESSION["cart"][$id]);
12 }
13 header("location:cart.php");
14 exit();
15 ?>
```

# Ví dụ



## Online BookStore



Tên sản phẩm

Giá Số lượng Tiền Xóa

[MySQL and mSQL](#)

28  28 [Xóa](#)

[1001 Web Site Construction Tips and Tricks](#) 40  40 [Xóa](#)

**Tổng tiền giỏ hàng: 68**

Cap Nhat Gio Hang

masach	[1]	[5]
\$_SESSION[cart]	1	1

[Mua Sach Tiep](#) - [Xoa Bo Gio Hang](#) - [Check out](#)



## Online BookStore



Tên sản phẩm

Giá Số lượng Tiền Xóa

[MySQL and mSQL](#)

28  140 [Xóa](#)

[1001 Web Site Construction Tips and Tricks](#) 40  400 [Xóa](#)

**Tổng tiền giỏ hàng: 540**

Cap Nhat Gio Hang

masach	[1]	[5]
\$_SESSION[cart]	5	10

[Mua Sach Tiep](#) - [Xoa Bo Gio Hang](#) - [Check out](#)



# Lưu giỏ hàng vào CSDL (checkout.php)

- Nếu khách chưa đăng nhập, yêu cầu đăng nhập bằng cách thông báo và chuyển đến trang login.php

Nếu khách đã đăng nhập, lần lượt thực hiện:

- Thêm record mới vào GioHang với các giá trị:  
\$\_SESSION[matv], ngày hiện hành,  
DATHANHTOAN=0
- Lấy MaGH mới thêm
- Lặp qua từng sản phẩm trong \$\_SESSION[cart],
  - Lấy giá của sản phẩm \$id
  - Thêm record mới vào CTGH với các giá trị:  
\$magh, \$id, \$value, \$gia, \$gia\*\$value



## Lập Trình Web

### Kỹ thuật phân trang trong PHP

GV: Trần Đình Nghĩa  
tdnghia1977@gmail.com



# Các bước tiến hành



- Tính tổng số record -  $\$totalrecords$
- Xác định số lượng record tối đa được trình bày trên trang -  $\$recs\_per\_page$
- Lấy biến trang hiện hành -  $\$page$ 
  - ↪ Nếu chưa có,  $\$page = 1$
- Sử dụng phát biểu LIMIT start, num\_record để lấy chính xác dữ liệu cần
- Xác định liên kết đến Trang trước, Trang kế

# Ví dụ mẫu

Online Book Store - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/BookStore/bookscategory.php?category\_id=1

Most Visited Getting Started Latest Headlines Slides Download Thất Giới Truyền Thuyết TKLTWeb Android TinhTe

Bai 20: Viet ung dung tao gio hang ... Gmail - Inbox - tdnghia1977@gmail... Online Book Store



Trang chủ Đăng ký Giỏ hàng Đăng nhập Quản trị

## Sản phẩm

Tên sản phẩm	Tác giả	Giá	Thẻ loại
<a href="#">Perl and CGI for the World Wide Web: Visual QuickS</a>	Elizabeth Castrow	15	Programming
<a href="#">PHP and MySQL Web Development</a>	Luke Welling, Laura Thomson	40	Programming
<a href="#">MySQL &amp; PHP From Scratch</a>	Wade Maxfield	24	Programming
<a href="#">Web Application Development with PHP 4.0 (with CD-</a>	Tobias Ratschiller, Till Gerken	36	Programming
<a href="#">Black Belt Web Programming Methods, Servers, Secur</a>	Không biết	28	Programming
<a href="#">Web Development with Java Server Pages</a>	Duane K. Fields, Mark A. Kolb	36	Programming
<a href="#">Professional Java Server Programming J2EE Edition</a>	Wrox Multi Team	48	Programming
<a href="#">Mastering ColdFusion 4.5</a>	Arman Danesh, Kristin Aileen Motlagh, Kristin Motlagh	40	Programming
<a href="#">Teach Yourself ColdFusion in 21 Days (Teach Yourse</a>	Charles Mohnike	32	Programming
<a href="#">ColdFusion Fast &amp; Easy Web Development</a>	T. C., III Bradley	20	Programming
<a href="#">Beginning Active Server Pages 3.0</a>	David Buser, Chris Ullman, Jon Duckett...	32	Programming
<a href="#">C# - Programming with the Public Beta</a>	Burton Harvey, Simon Robinson, Julian Templeman, Burt Harvey, Karli Watson	35	Programming
<a href="#">Sách mới thêm</a>	Nghĩa Trần	80	Programming

Done

1 0 tdnghia1977@gmail.com



# Ví dụ mẫu



- `$fldcategory_id = $_REQUEST["category_id"];`
- `$conn = mysql_connect("localhost", "root", "") or die ("Không thể kết nối csdl");`
- `mysql_select_db("bookstore", $conn);`
- `mysql_set_charset("utf8", $conn);`
- `$sql = "SELECT MASACH, TENSACH, TENTG, GIA, GHICHU, TENTL FROM SACH, THELOAI WHERE SACH.MATL=" . $fldcategory_id . " and THELOAI.MATL=SACH.MATL "`
- `$result = mysql_query($result);`
- `while ($row = mysql_fetch_array($result))`
  - `{`
  - `// các lệnh trình bày record $row`
  - `}`



# \$totalrecords và \$recs\_per\_page



- Tính tổng số record – \$totalrecord: Bổ sung lệnh sau vào trước câu lệnh gán \$sql trong ví dụ mẫu:
  - \$sql = "SELECT COUNT(\*) FROM SACH WHERE MATL=" . \$fldcategory\_id;
  - \$result = mysql\_query(\$sql);
  - \$row = mysql\_fetch\_array(\$result);
  - **\$totalrecords = \$row[0];**
- Xác định lượng record tối đa được trình bày trên trang - *\$recs\_per\_page*:
  - \$recs\_per\_page = 5;





# &page & LIMIT

➤ Lấy biến trang hiện hành – \$page

↳ \$page = \$\_REQUEST["page"];

↳ if (!isset(\$page)) \$page = 1;

➤ Sử dụng phát biểu LIMIT start, num\_record:  
sửa lệnh gán \$sql trong ví dụ mẫu thành:

↳ \$sql = "SELECT MASACH, TENSACH, TENTG,  
GIA, GHICHU, TENTL  
FROM SACH, THELOAI  
WHERE SACH.MATL=" . \$fldcategory\_id . " and  
THELOAI.MATL=SACH.MATL **LIMIT** " . (\$page-  
1)\*\$recs\_per\_page . ", " . \$recs\_per\_page;



# Xác định liên kết đến Trang trước, Trang kế

- Bổ sung các lệnh sau vào vị trí đặt liên kết

```
<?php
$sFileName="bookscategory.php";

if (( $page == 1 ) && ($page*$recs_per_page<$totalrecords))
    echo("<a href=" . $strFileName . "?page=" . ($page + 1) .
"&category_id=" . $fldcategory_id . ">Trang kế</a>");
else {
    if ($page > 1)
        echo("<a href=" . $strFileName . "?page=" . ($page - 1) .
"&category_id=" . $fldcategory_id . ">Trang trước</a>");
    if ($page*$recs_per_page<$totalrecords)
        echo("<a href=" . $strFileName . "?page=" . ($page + 1) .
"&category_id=" . $fldcategory_id . "> Trang kế</a>");
}
?>
```



## Lập Trình Web

### Các trang quản trị trong PHP

GV: Trần Đình Nghĩa  
tdnghia1977@gmail.com



# Nội dung



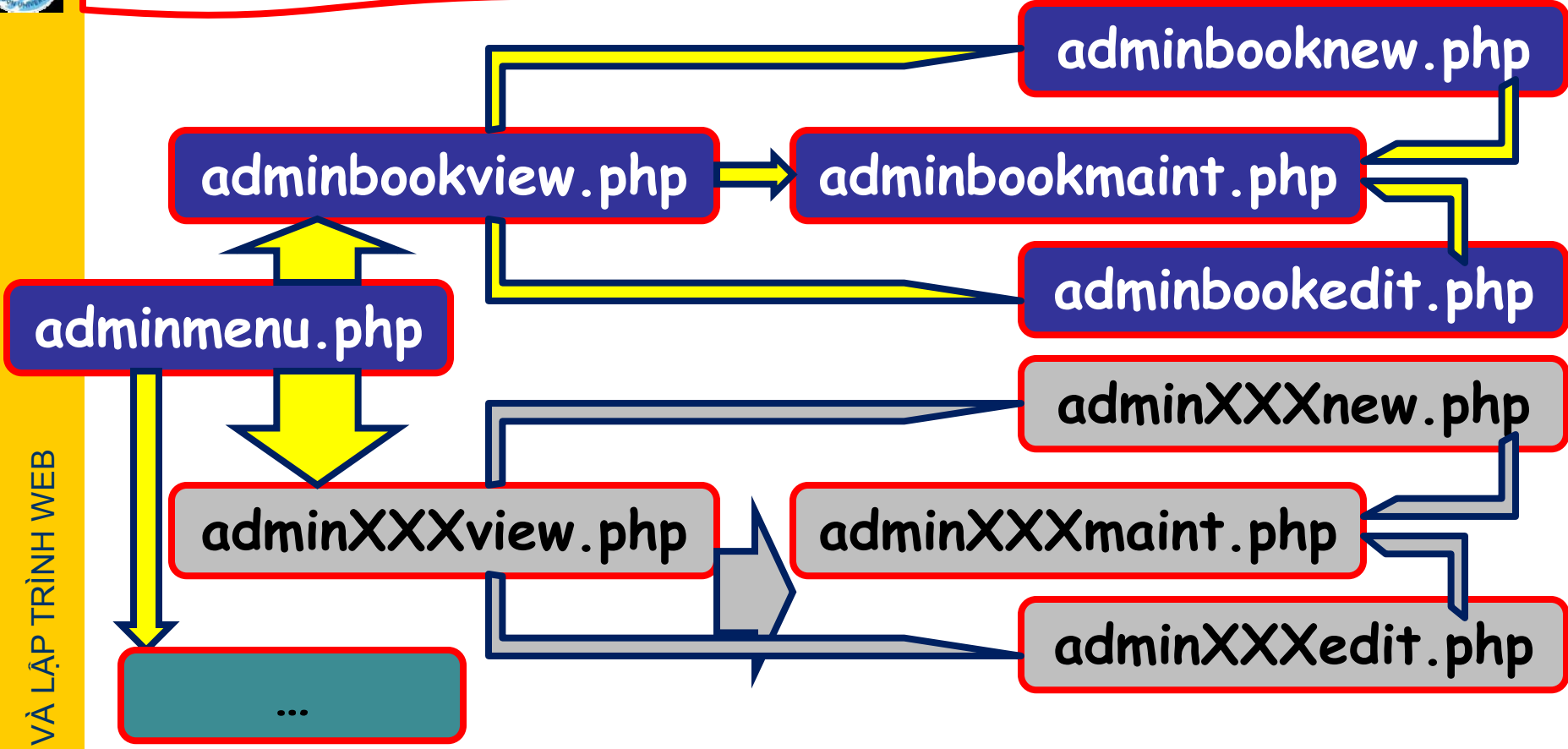
- Mô hình hoạt động các trang quản trị
- Chức năng từng trang
- `checkadmin.php`
- `adminbookmaint.php`



# Mô hình hoạt động



☠️ **checkadmin** ☠️



THIẾT KẾ VÀ LẬP TRÌNH WEB

# Chức năng từng trang

- adminXXXview.php: trình bày tất cả các mục thuộc danh mục XXX (ví dụ: book) cần quản trị
  - ↪ Nên áp dụng phân trang, hay chia theo thể loại, hoặc cả hai
  - ↪ Với mỗi mục trình bày, nên có liên kết đến adminXXXmaint.php cho phép Xóa, Sửa
  - ↪ Chứa liên kết cho phép thêm mục mới



**Online BookStore**



Trang chủ



Đăng ký



Giỏ hàng



Đăng nhập



Quản trị

[Thêm sản phẩm mới](#)

## sản phẩm

Tên sản phẩm	Tác giả	Giá	Thể loại	Trình bày?	Cập nhật	Xóa
Programming Perl (3rd Edition)	Larry Wall, Tom Christiansen, Jon Orwant	40	Programming	No	<a href="#">Sửa</a>	<a href="#">Xóa</a>
Perl and CGI for the World Wide Web: Visual QuickS	Elizabeth Castrow	15	Programming	No	<a href="#">Sửa</a>	<a href="#">Xóa</a>
PHP and MySQL Web Development	Luke Welling, Laura Thomson	40	Programming	No	<a href="#">Sửa</a>	<a href="#">Xóa</a>
MySQL & PHP From Scratch	Wade Maxfield	24	Programming	No	<a href="#">Sửa</a>	<a href="#">Xóa</a>
Web Application Development with PHP 4.0 (with CD-	Tobias Ratschiller, Till Gerken	36	Programming	No	<a href="#">Sửa</a>	<a href="#">Xóa</a>
Black Belt Web Programming Methods, Servers, Secur	Không biết	28	Programming	No	<a href="#">Sửa</a>	<a href="#">Xóa</a>
Web Development with Java Server Pages	Duane K. Fields, Mark A. Kolb	36	Programming	No	<a href="#">Sửa</a>	<a href="#">Xóa</a>
Professional Java Server Programming J2EE Edition	Wrox Multi Team	48	Programming	No	<a href="#">Sửa</a>	<a href="#">Xóa</a>
Mastering ColdFusion 4.5	Arman Danesh, Kristin Aileen Motlagh, Kristin Motlagh	40	Programming	No	<a href="#">Sửa</a>	<a href="#">Xóa</a>
Teach Yourself ColdFusion in 21 Days (Teach Yourse	Charles Mohnike	32	Programming	No	<a href="#">Sửa</a>	<a href="#">Xóa</a>

[Trang kế](#)

[Trang chủ](#) [Đăng ký](#) [Giỏ hàng](#) [Đăng nhập](#) [Quản trị](#)

Coded by: • Trần Đình Nghĩa





# Chức năng từng trang

- adminXXXedit.php, adminXXXnew.php: dùng các FormField tương ứng với các thuộc tính của mục XXX cho phép sửa thông tin / thêm mới
  - ↪ Trình bày các thuộc tính nên rõ ràng, tạo sự thuận tiện cho thao tác nhập liệu
  - ↪ Sử dụng Javascript kiểm tra dữ liệu nhập trước khi submit đến adminXXmaint.php
  - ↪ Nếu có upload file, phải khai báo form là dạng `enctype="multipart/form-data"`

Chỉnh sửa sản phẩm	
Tên sản phẩm	<input type="text" value="Programming Perl (3rd Edition)"/>
Tên tác giả	<input type="text" value="Larry Wall, Tom Christiansen, Jon"/>
Thể loại	<input type="text" value="Programming"/>
Giá	<input type="text" value="40"/>
Đường dẫn hình	<input type="text" value="images/books/0596000278.jpg"/> <input type="button" value="Browse..."/>
Ghi chú	<pre>Perl is a powerful programming language that has grown in popularity since it first appeared in 1988. The first edition of this book, Programming Perl, hit the shelves in 1990, and was quickly adopted</pre>
Trình bày	<input type="checkbox"/>
<input type="button" value="Lưu"/>	

Thêm mới sản phẩm	
Tên sản phẩm	<input type="text"/>
Tên tác giả	<input type="text"/>
Thể loại	<input type="text" value="Programming"/>
Giá	<input type="text"/>
Đường dẫn hình	<input type="text"/> <input type="button" value="Browse..."/>
Ghi chú	<input type="text"/>
Trình bày	<input type="checkbox"/>
<input type="button" value="Lưu"/>	



# Chức năng từng trang



- adminXXXmaint.php: thông thường chỉ chứa các đoạn lệnh php thực hiện các thao tác thêm, sửa, xóa các mục XXX trên CSDL.
- Sau khi xử lý các thao tác, thường có thông báo cụ thể tình trạng lỗi và chuyển hướng trang
- Có thể trình bày hoàn chỉnh như 1 trang bình thường và chứa các liên kết chuyển hướng trang





# checkadmin.php



```

<?php
session_start();
$admin = 0;
if (isset($_SESSION["tv"]))
{
    $conn = mysql_connect("localhost", "root", "") or die("Can not connect to database");
    mysql_select_db("bookstore", $conn);
    mysql_set_charset("utf8", $conn);
    $sql = "SELECT * FROM THANHVIEN WHERE MATV=" . $_SESSION["tv"] . " AND ADMIN=1";
    $result = mysql_query($sql);
    $row = mysql_fetch_array($result);
    if ($row["MATV"] == $_SESSION["tv"])
        $admin = 1;
    mysql_close($conn);
}
if (!isset($_SESSION["tv"]) || $admin==0)
{
    echo('<html><head><META HTTP-EQUIV="refresh" CONTENT="3;
URL=http://localhost/BookStore/index.php"></head><body>');
    echo("<h3>Ban khong phai la administrator cua website.<br>");
    echo("Vui long click vao <a href='index.php'>ĐÃ Y</a> để đến trang chủ hoặc cho' trong
gia'y để trình duyệt của bạn tự chuyển.</h3></body></html>");
    exit();
}
?>

```



- Hoạt động dựa trên biến option được truyền theo địa chỉ:
  - ↪ addnew: Lấy các thông tin được POST từ **adminbooknew** về sách được thêm mới, xây dựng và thi hành câu lệnh SQL insert.
  - ↪ edit: Lấy các thông tin được POST từ **adminbookedit** về sách được sửa, xây dựng và thi hành câu lệnh SQL update.
  - ↪ delete: Lấy các thông tin được truyền theo địa chỉ từ **adminbookview** về sách muốn xóa, xây dựng và thi hành câu lệnh SQL delete.