



TRUNG TÂM TIN HỌC
Trường Đại học Khoa Học Tự Nhiên - Đại học quốc gia TP.HCM

LẬP TRÌNH VIÊN MÃ NGUỒN MỞ PHP

Module 1

Click vào phụ lục để chuyển tới bài cần đọc

Phụ lục

Bài 1 Tổng quan.....	2
Bài 2 Các kiểu dữ liệu cơ sở.....	12
Bài 3 Xây dựng phương thức(hàm) và xử lý lỗi	22
Bài 4 Sử dụng hàm của PHP.....	37
Bài 5 Xử lý tập tin	44
Bài 6 Mảng cơ bản.....	55
Bài 7 Lập trình hướng đối tượng cơ bản	65
Bài 8 Form và các điều khiển cơ sở	81
Bài 9 Sử dụng dịch vụ	94





Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC

LTV MÃ NGUỒN MỞ PHP

Module 1 – Bài 1: *Tổng quan*

Ngành LT & CSDL

www.t3h.vn



2014

2014



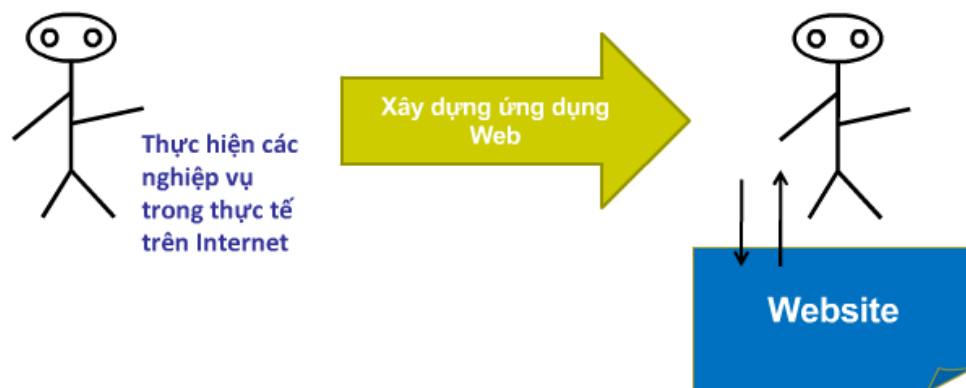
Nội dung

1. Giới thiệu ứng dụng Web
2. Giới thiệu ngôn ngữ lập trình PHP
3. Wamp Server
4. Xây dựng ứng dụng



Giới thiệu ứng dụng Web

☐ Ý nghĩa sử dụng



Ứng dụng Web là ứng dụng được xây dựng để hỗ trợ người dùng thực hiện các nghiệp vụ trên Internet



Giới thiệu ứng dụng Web



☐ Cơ chế tương tác từ người dùng đến Web Server



Giới thiệu ứng dụng Web



☐ Cơ chế tương tác từ người dùng đến Web Server



Nội dung

1. Giới thiệu ứng dụng Web
2. Giới thiệu ngôn ngữ lập trình PHP
3. Wamp Server
4. Xây dựng ứng dụng



Giới thiệu ngôn ngữ lập trình PHP

☐ Ngôn ngữ lập trình PHP (*Hypertext Preprocessor*)

- Là một ngôn ngữ lập trình phía server-side được thiết kế để xây dựng và phát triển ứng dụng Web.





Giới thiệu ngôn ngữ lập trình PHP

□ Đặc điểm

- Chạy trên hệ thống máy chủ (Server)
- Rất đơn giản
- Tốc độ xử lý nhanh, dễ sử dụng
- Luôn được cải tiến và cập nhật (mã nguồn mở)
- Có nhiều hướng dẫn sử dụng trên mạng
- Hoàn toàn miễn phí
- Có thể thực thi trên bất cứ hệ điều hành (Operator System) nào.
- Không chỉ làm việc với HTML mà còn có thể làm việc được với hình ảnh, PDF, Flash movie,...
- Có thể dễ dàng nối kết với các cơ sở dữ liệu như MySQL, mSQL, FrontBase, dBase, Solid, ODBC, Oracle, FilePro...
- Được sử dụng để xây dựng các framework lớn như: Joomla, Drupal, Zend...
- Có nhiều editor hỗ trợ: PHPDesigner, DreamWeaver...



LTV Mã nguồn mở PHP – Module 1

8

Nội dung



1. Giới thiệu ứng dụng Web
2. Giới thiệu ngôn ngữ lập trình PHP
3. Wamp Server
4. Xây dựng ứng dụng



LTV Mã nguồn mở PHP – Module 1

9

Wamp Server

❑ Wamp Server

- là một gói phần mềm **Web Server** tất cả trong một (All-in-One), gồm có: Apache, MySQL, PHP chạy trên nền Windows



Wamp Server

❑ Đặc điểm

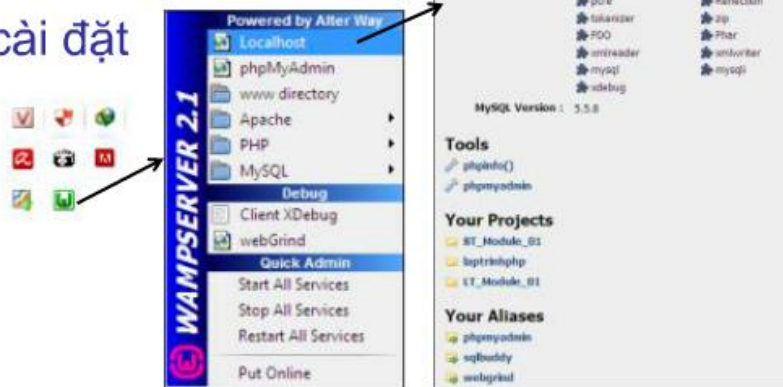
- Có thể cài đặt dễ dàng
- Được cập nhật đều đặn
- Rất thuận lợi cho việc tạo máy chủ Web để chạy thử, xây dựng Website bằng PHP
- Hỗ trợ PHP5
- Hoàn toàn miễn phí



Wamp Server

❑ Cài đặt

- Download Wamp tại địa chỉ <http://www.wampserver.com>
- Lần lượt thực hiện các bước để cài đặt



LTV Mã nguồn mở PHP – Module 1

12

Nội dung

1. Giới thiệu ứng dụng Web
2. Giới thiệu ngôn ngữ lập trình PHP
3. Wamp Server
4. Xây dựng ứng dụng

LTV Mã nguồn mở PHP – Module 1

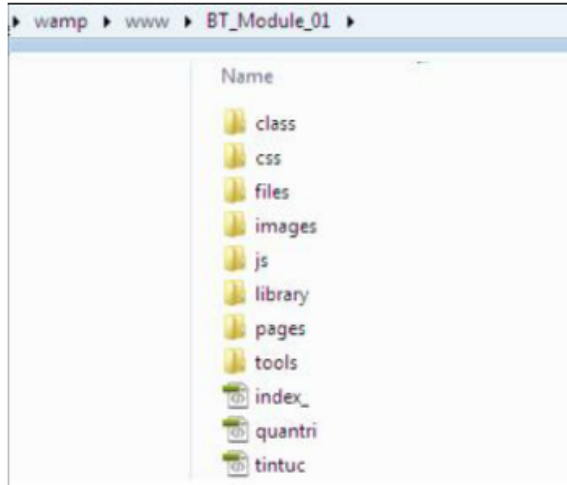
13



Xây dựng ứng dụng

❑ Tổ chức và lưu trữ ứng dụng

- Đặt trong thư mục wamp/www



Xây dựng ứng dụng

❑ Editor để xây dựng ứng dụng PHP

- Macro Media DreamWeaver
- PHP Designer
- ...





Xây dựng ứng dụng

❑ Thiết lập trang PHP

- Là một trang động (Dynamic Page)
- Có **charset = UTF-8** (để trang hiển thị tiếng Việt)
- Nhúng code PHP vào trang bằng thẻ php
- Hiển thị nội dung trên trang bằng lệnh **echo**



LTV Mã nguồn mở PHP – Module 1

16

Xây dựng ứng dụng



❑ Quy ước

- Code PHP được đặt trong các thẻ sau:

Thẻ mở	Thẻ đóng
<code><?php</code>	<code>?></code>
<code><?</code>	<code>?></code>
<code><script language="php"></code>	<code></script></code>

- Ví dụ:

```
<?php echo "<b>Chào các bạn.</b>" ?>
```

```
<? echo "<b>Chào các bạn.</b>" ?>
```

```
<script language="php">echo "<b>Chào các bạn.</b>"</script>
```



LTV Mã nguồn mở PHP – Module 1

17

Xây dựng ứng dụng

❑ Quy ước

- Các lệnh kết thúc bằng dấu ;
- Ghi chú trong PHP:

```

1 <?php
2 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
3 <title>Trang đầu tiên</title>
4 </head>
5
6 <body>
7 <?php
8 // in câu giới thiệu
9 echo "Đây là ngôn ngữ lập trình PHP";
10 // tính tổng hai số
11 sau đó in kết quả của tổng
12 */
13 $a = 5;
14 $b = 7;
15 $tong = $a+$b;
16 echo "Tổng của $a và $b là: " . $tong;
17
18 </body>
19 </html>
20

```

LTV Mã nguồn mở PHP – Module 1

18

18

Ví dụ minh họa

❑ Viết chương trình in câu chào

- Yêu cầu: Viết chương trình in câu chào như sau:

Chào mừng các bạn đến với
Ngôn ngữ lập trình PHP

- Thiết kế giao diện người dùng
 - Th_cau_chao: paragraph, align = "center"
- Tóm tắt yêu cầu:
 - Nhập: không có
 - Xuất: "Chào mừng các bạn đến với Ngôn ngữ lập trình PHP"
 - Quy tắc xử lý: không có
- Thuật giải: không có

LTV Mã nguồn mở PHP – Module 1

19



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC

LTV MÃ NGUỒN MỞ PHP

Module 1 – Bài 2: *Các kiểu dữ liệu cơ sở*

Ngành LT & CSDL

t3h.vn



2014

2014





Nội dung

1. Các kiểu dữ liệu cơ sở
2. Biến
3. Các phương thức kiểm tra giá trị của biến
4. Hằng



Các kiểu dữ liệu cơ sở



□ Đặc điểm

- Kiểu **Boolean**: chỉ có một trong hai giá trị là TRUE và FALSE
- Kiểu **Integer** (số nguyên): Giá trị có thể là số trong hệ thập phân, thập lục phân và bát phân.
 - Ví dụ:
 - 1234 // hệ thập phân
 - -123 // số âm hệ thập phân
 - 0123 // hệ bát phân (bắt đầu bằng 0)
 - 0x1A // hệ thập lục phân (bắt đầu bằng 0x)





Các kiểu dữ liệu cơ sở

□ Đặc điểm

- Kiểu **Float/ Double** (số thực)
 - Ví dụ:
 - 1.234
 - $1.2e3 \rightarrow 1.2 * 1000 = 1200$
- Kiểu **String** (chuỗi, ký tự)
 - Mỗi ký tự chiếm 1 byte
 - Mỗi chuỗi có thể chứa một hay nhiều ký tự thuộc 256 ký tự khác nhau
 - Chuỗi không có giới hạn về kích thước



Các kiểu dữ liệu cơ sở

□ Đặc điểm

- Kiểu **Array** (mảng các phần tử)
 - Ví dụ:
 - `array(1, 2, 3, 4, 5)`
- Kiểu **Object** (đối tượng)
 - Ví dụ
 - `$xe_hoi = new Xe(); // đối tượng xe hơi`





Các kiểu dữ liệu cơ sở

□ Chuyển đổi kiểu dữ liệu

- Trong quá trình tính toán, kiểu dữ liệu cũ có thể không còn phù hợp nữa (kết quả tính toán vượt khỏi phạm vi của kiểu dữ liệu cũ) => chuyển đổi kiểu dữ liệu
- Thực hiện: ghi tên kiểu dữ liệu mà biến muốn chuyển đổi vào phía trước biến
- Ví dụ:

```
$don_gia = 50000;  
$so_luong = 100;  
$thanh_tien = (double) ($so_luong*$don_gia);
```



Nội dung



1. Các kiểu dữ liệu cơ sở
2. Biến
3. Các phương thức kiểm tra giá trị của biến
4. Hằng



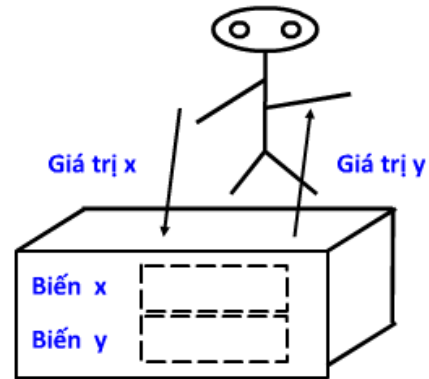
Biến

□ Khái niệm



Thực hiện nghiệp vụ f trong thực tế liên quan đến thông tin x, y

Xây dựng chương trình



Ví dụ

- Tính $y = \text{Căn bậc 2 của } x$
- Tính số ngày lưu trú tại khách sạn = Ngày đi – ngày đến
- Tính tiền điện căn cứ trên chỉ số cuối – chỉ số đầu

Biến là một vùng nhớ trong bộ nhớ chính giúp **biểu diễn thông tin** thực tế trong chương trình: chuỗi, số nguyên, số thực, boolean...



LTV Mã nguồn mở PHP – Module 1

8

Biến

□ Khai báo

- Cú pháp: `$ten_bien`

□ Quy tắc đặt tên biến

- Bắt đầu bằng ký tự \$, theo sau là 1 ký tự hoặc dấu _, tiếp đó là ký tự, ký số hoặc dấu _
- Nên khởi tạo giá trị ban đầu cho biến
- Không trùng với tên hàm
- Không nên bắt đầu bằng ký số
- Lưu ý: Tên biến có phân biệt chữ HOA – chữ thường



LTV Mã nguồn mở PHP – Module 1

9

Biến

□ Gán giá trị cho biến

- Trong PHP, khi khai báo Biến không cần ghi kiểu mà nó tự xác định kiểu dữ liệu cho biến khi biến được gán giá trị.

Khai báo chuỗi `Ho_ten` với giá trị Lê hùng

Khai báo số nguyên `m` với giá trị 4

Khai báo số thực `y` với giá trị là 3,4

Khai báo biến logic `Gioi_tinh` với giá trị Nam

```
$Ho_ten = "Lê Hùng"; // String
```

```
$m = 4; // Integer
```

```
$y = 3.4 ; // Double
```

```
$Gioi_tinh = true;
```



Biến

□ Kết xuất

Vấn đề:

Cần kết xuất thông tin của biến `x`

Ví dụ:

Kết xuất biến chuỗi `Ho_ten`

Kết xuất biến số thực `y`

Kết xuất biến logic `Gioi_tinh`

Giải quyết:

Khai báo chuỗi `$Chuoi` với giá trị ban đầu tương ứng ghi chú (hàng chuỗi) về nội dung của biến cần xuất

Bổ sung vào `$Chuoi` giá trị của biến cần xuất

Xuất `$Chuoi`

```
// Kết xuất Ho_ten
```

```
$Chuoi = "Ho ten:";
```

```
$Chuoi = $Chuoi . $Ho_ten;
```

```
// Kết xuất y
```

```
$Chuoi = "y = ";
```

```
$Chuoi = $Chuoi . $y;
```

```
// Kết xuất Gioi_tinh
```

```
$Chuoi = "Gioi tinh: ";
```

```
$Chuoi = $Chuoi + $Gioi_tinh;
```

Lệnh kết xuất chuỗi

Ý nghĩa: Cho phép xuất thông tin của biến hay hằng chuỗi

Cú pháp

echo Hằng;

Hay

echo Biến;

Phép toán bổ sung chuỗi

Ý nghĩa: Cho phép bổ sung thông tin của biến hay hằng chuỗi vào biến chuỗi `$S`

Cú pháp

`$S = $S . Hằng chuỗi`

Hay

`$S = $S + Biến chuỗi`



Biến

❑ Biến nhận giá trị từ điều khiển

Vấn đề:

Cần nhận giá trị từ điều khiển và gán cho biến x

Ví dụ:

Chuỗi Ho_ten nhận giá trị từ điều khiển textfield txtHo_ten

Biến y nhận giá trị từ điều khiển textfield txtY

Biến logic Gioi_tinh nhận giá trị từ điều khiển radio button rdoGioi_tinh

Giải quyết:

`x = $_POST["Tên điều khiển tương ứng"]`

```
// Nhận Ho_ten
$Ho_ten = $_POST["txtHo_ten"];

// Nhận y
$y = $_POST["y"];

// Nhận Gioi_tinh
$Gioi_tinh =
$_POST["rdoGioi_tinh"]
```



Nội dung

1. Các kiểu dữ liệu cơ sở
2. Biến
3. Các phương thức kiểm tra giá trị của biến
4. Hằng





Các phương thức kiểm tra giá trị của biến

Vấn đề:

Cần kiểm tra xem biến x có giá trị hay không, kiểu dữ liệu của biến x...

Ví dụ:

- Kiểm tra sự tồn tại của biến `Ho_ten`
- Kiểm tra biến y xem có rỗng hay không
- Kiểm tra biến m có phải là kiểu số hay không
- Kiểm tra biến y có phải là kiểu double hay không
- Xác định kiểu của biến `Ho_ten`

Giải quyết:

Sử dụng các phương thức kiểm tra giá trị của biến

```
// Kiểm tra tồn tại của biến
Ho_ten

$jq = isset($Ho_ten); // true
hoặc false

// Kiểm tra biến y có rỗng hay
không

$Is_empty = empty($y); // true
hoặc false

// Kiểm tra biến m có phải là
kiểu số hay không

$Is_number = is_numeric($m); //
true hoặc false

// Kiểm tra biến y có phải là kiểu
double hay không

$Is_y_double = is_double($y); //
true hoặc false

// Xác định kiểu của biến Ho_ten

$Type_Ho_ten = gettype($Ho_ten);
// string
```



Nội dung



1. Các kiểu dữ liệu cơ sở
2. Biến
3. Các phương thức kiểm tra giá trị của biến
4. Hằng



Hằng



□ Khái niệm

- Hằng là một giá trị không thể chỉnh sửa được trong quá trình thực hiện chương trình.
- Ví dụ: PI, USD (tỷ giá)...



Hằng



□ Quy tắc đặt tên

- Quy tắc đặt tên hằng cũng giống như quy tắc đặt tên biến.
- Tên hằng thường IN HOA





Hằng

□ Khai báo

- Dùng hàm `define()` để định nghĩa

```
define("TÊN_HÀNG", giá_trị);
```

- Ví dụ: Khai báo hằng số PI có giá trị là 3.14

```
define("PI", 3.14);
```

● Ghi chú

- Một khi hằng được định nghĩa, nó không bị thay đổi.
- Chỉ có các kiểu dữ liệu boolean, integer, float, string mới có thể chứa các hằng.





Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC

LTV MÃ NGUỒN MỞ PHP

Module 1 – Bài 3: *Xây dựng phương thức (hàm) và xử lý lỗi*

Ngành LT & CSDL

t3h.vn



2014

2014



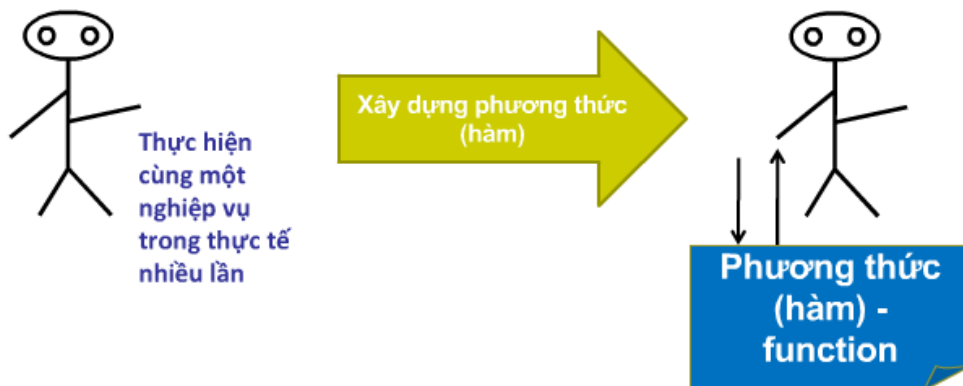
Nội dung

1. Phương thức (hàm)
2. Xây dựng phương thức (hàm)
3. Xử lý lỗi



Phương thức (hàm)

□ Khái niệm



Function là một đoạn code đặc biệt được viết để thực hiện một công việc nào đó thường được lặp đi lặp lại nhiều lần.





Phương thức (hàm)

□ Phân loại

- Built-in function: hàm do PHP cung cấp. Có trên 700 hàm chia thành nhiều nhóm: chuỗi, toán học, thời gian, lịch, mảng, tập tin, mail, xử lý CSDL... chỉ cần gọi hàm khi sử dụng và truyền vào các giá trị phù hợp dựa trên danh sách các tham số.

■ Ví dụ:

```
$so = 121.1542;
round($so, 2); → 121.15
round($so, -1); → 120
date("d/m/Y"); → 15/01/2014
```



Phương thức (hàm)



□ Phân loại

- User defined function: hàm do người dùng tự định nghĩa.

■ Ví dụ:

```
// ax + b = 0
$ngghiem = Giai_phuong_trinh_bac_I(2, 4);
```





Nội dung

1. Phương thức (hàm)
2. Xây dựng phương thức (hàm)
3. Xử lý lỗi



Xây dựng phương thức (hàm)



□ Khai báo

```
function
Tên_hàm(Danh_sách_các_tham_số)
{
    khối_lệnh_bên_trong_hàm
    return giá_trị;
}
```

• Trong đó:

- Tên hàm: được sử dụng khi gọi hàm, tên hàm nên có ý nghĩa gợi nhớ
- Danh sách các tham số: dùng để truyền dữ liệu bên ngoài vào, hàm có thể có hoặc không có tham số
- Giá trị: là kết quả trả về của hàm. Hàm có thể có hoặc không có giá trị trả về





Xây dựng phương thức (hàm)

- Ví dụ: Xây dựng hàm tính diện tích hình chữ nhật

```
function Tinh_S_hcn($a, $b)
{
    $S = $a * $b;
    return $S;
}
```



LTV Mã nguồn mở PHP – Module 1

8



Xây dựng phương thức (hàm)

□ Sử dụng

Tên_hàm(**Danh_sách_các_giá_trị**)

- Trong đó:
 - Tên hàm gọi sử dụng phải giống tên hàm đã xây dựng
 - Danh sách các giá trị: cung cấp các thông tin cho tham số của hàm. Với:
 - Số lượng các giá trị bằng số lượng các tham số của hàm
 - Thứ tự tương ứng theo thứ tự các tham số
 - Kiểu dữ liệu của giá trị phải tương ứng với kiểu của tham số
 - Nếu hàm không có giá trị truyền vào thì phía sau tên hàm cũng phải có cặp **()**



LTV Mã nguồn mở PHP – Module 1

9



Xây dựng phương thức (hàm)

- Ví dụ: gọi sử dụng hàm tính diện tích hình chữ nhật

```
$s = Tinh_S_hcn(4, 8);
```



Xây dựng phương thức (hàm)

□ Phân loại tham số của hàm

- Có 2 loại:
 - Tham trị: truyền tham số theo giá trị
 - Tham biến: truyền tham số theo địa chỉ





Xây dựng phương thức (hàm)

❑ Tham trị

- Tham số truyền giá trị từ bên ngoài vào cho hàm.
- Khi thay đổi giá trị của tham trị bên trong hàm thì giá trị của nó ở ngoài hàm vẫn không bị thay đổi.



LTV Mã nguồn mở PHP – Module 1

12



Xây dựng phương thức (hàm)

• Ví dụ

```
function them_vao_chuoi($chuoi)
{
    $chuoi .= "và chuỗi sau khi thêm.";
    return $chuoi;
}

$chuoi_goc= "Đây là chuỗi gốc, ";
echo them_vao_chuoi($chuoi_goc); // "Đây là
chuỗi gốc, và chuỗi sau khi thêm.«
echo $chuoi_goc ; // "Đây là chuỗi gốc, "
```



LTV Mã nguồn mở PHP – Module 1

13



Xây dựng phương thức (hàm)

❑ Tham biến

- Tham số truyền giá trị từ bên ngoài cho hàm và trả giá trị ở trong hàm ra bên ngoài.
- Khi thay đổi giá trị của tham biến bên trong hàm thì giá trị của nó ở ngoài hàm cũng sẽ thay đổi sau khi chúng ta gọi hàm đã xây dựng.
- Đối với tham biến chúng ta sẽ sử dụng cú pháp với ký tự **&** ở phía trước.



LTV Mã nguồn mở PHP – Module 1

14



Xây dựng phương thức (hàm)

- Cú pháp:

```
function Tên_hàm(&Tên_tham_biến, ...)
{
    ...
    Tên_tham_biến = giá trị;
    return ...;
}
```



LTV Mã nguồn mở PHP – Module 1

15



Xây dựng phương thức (hàm)

● Ví dụ:

```
function them_vao_chuoi(&$chuoi)
{
    $chuoi .= "và chuỗi sau khi thêm.";
    return $chuoi;
}
$chuoi_goc= "Đây là chuỗi gốc, ";
echo them_vao_chuoi($chuoi_goc); // "Đây là chuỗi
    gốc, và chuỗi sau khi thêm."
echo $chuoi_goc;
// "Đây là chuỗi gốc, và chuỗi sau khi thêm."
```



Xây dựng phương thức (hàm)

❑ Tham số tùy chọn

- Là những tham số có thể được truyền giá trị hoặc không
- Cho phép tạo sẵn các giá trị mặc định cho tham số
- Những tham số này chỉ xuất hiện ở cuối danh sách các tham số





Xây dựng phương thức (hàm)

●Cú pháp

```
function Tên_hàm(Danh sách các tham trị, tham
biến, $tham_số_tùy_chọn=giá_trị)
{
    khối lệnh bên trong hàm
    return giá_trị;
}
```



Xây dựng phương thức (hàm)

●Ví dụ

```
function so_thich($nuoc_uong = "cà phê")
{
    return "Tôi thích uống $nuoc_uong.";
}

echo so_thich(); // "Tôi thích uống cà phê."
echo so_thich("nước ép trái cây"); // "Tôi thích
uống nước ép trái cây."
```



Nội dung

1. Phương thức (hàm)
2. Xây dựng phương thức (hàm)
3. Xử lý lỗi



Xử lý lỗi

☐ Phân loại lỗi

- Lỗi cú pháp (syntax error)
- Lỗi thực thi
- Lỗi luận lý





Xử lý lỗi

● Lỗi cú pháp

- Xuất hiện khi ta viết code
- Được thông báo khi ta thực thi trang
- Nguyên nhân: Viết sai hoặc thiếu cú pháp
- Ví dụ: Một số lỗi sai thường gặp
 - Tên biến có chứa khoảng trắng
 - Thiếu ; cuối lệnh
 - Sai do PHP có phân biệt ký tự thường và hoa
 - Chuỗi không bắt đầu và kết thúc bởi cặp nháy đơn “ hoặc nháy đôi ”
 - ...



Xử lý lỗi

● Lỗi thực thi

- Xảy ra khi ta thực thi trang
- Khó xác định hơn lỗi cú pháp
- Nguyên nhân: Mở một tệp tin đang tồn tại, chia cho 0, truy xuất bảng không tồn tại trong CSDL ...



Xử lý lỗi



- Lỗi luận lý
 - Xảy ra khi ta thực thi trang
 - Được thể hiện dưới những hình thức hoặc những kết quả không mong đợi
 - Nguyên nhân: Sai lầm trong thuật giải



Xử lý lỗi



□ Sửa lỗi

- Sửa lỗi thủ công
- Dùng try ... catch để bắt lỗi





Xử lý lỗi

● Sửa lỗi thủ công

- Khi chạy chương trình, nếu có lỗi phát sinh thì trang thực thi sẽ tự động thông báo lỗi
- Cách sửa lỗi:
 - Xem thông báo lỗi và dòng xảy ra lỗi
 - Mở code và sửa lỗi tại dòng đó (hoặc những dòng lân cận)



Xử lý lỗi

● Dùng try ... catch để bắt lỗi

- Cho phép thử thực hiện một khối lệnh xem có bị lỗi hay không, nếu có sẽ bắt và xử lý lỗi.
- Cấu trúc có hai khối:
 - Khối **try**: các câu lệnh có khả năng gây ra lỗi
 - Khối **catch**: các câu lệnh để bắt và xử lý lỗi phát sinh trên khối **try**.
- Một lỗi xảy ra khi thực thi trang gọi là một Exception.
- Nếu dòng nào trong khối lệnh có khả năng tạo ra lỗi thì gọi trả về lỗi đó.





Xử lý lỗi

• Dùng try ... catch để bắt lỗi

- Trong PHP lỗi sẽ không tự động trả về => gọi lỗi đó bằng cú pháp sau:

```
throw new Exception("Câu thông báo lỗi",
code) ;
```

- Với câu thông báo lỗi và mã lỗi (code) đều là các tham số tùy chọn
- Lưu ý:
 - Một khối try có thể dùng một hay nhiều khối catch
 - Mỗi khối catch hiển thị một loại lỗi khác nhau



Xử lý lỗi

• Cú pháp

```
try
{
    - khối lệnh có khả năng phát sinh lỗi
    - các lỗi
    throw new Exception("Câu thông báo lỗi",
code) ;
}
catch (Exception $e)
{
    echo "<p>Lỗi: " . $e->getMessage() . "</p>";
    //xuất lỗi
}
```





Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC

LTV MÃ NGUỒN MỞ PHP

Module 1 – Bài 4: *Sử dụng hàm của PHP*

Ngành LT & CSDL

t3h.vn

2014



2014



Nội dung

1. Giới thiệu
2. Các hàm toán học
3. Các hàm chuỗi
4. Các hàm thời gian

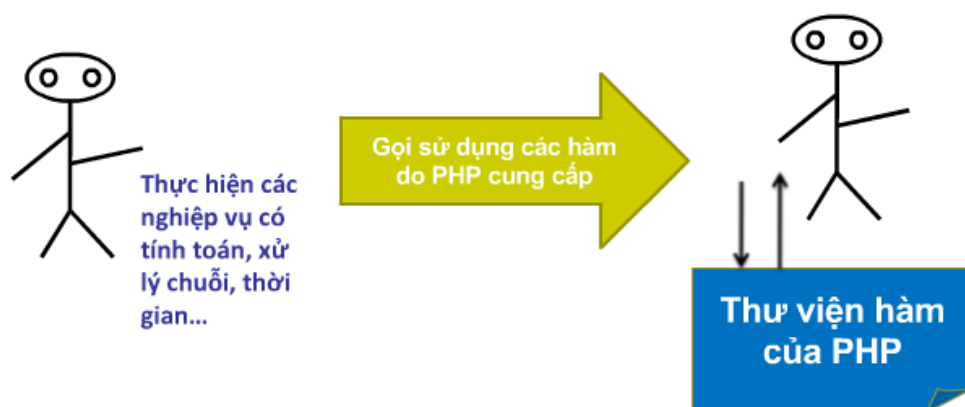


LTV Mã nguồn mở PHP – Module 1

2

Giới thiệu

□ Ý nghĩa sử dụng



Có những công việc được thực hiện đi thực hiện lại nhiều lần liên quan đến việc tính toán, xử lý chuỗi, thời gian... => Thư viện hàm PHP gồm trên 700 hàm được xây dựng và cung cấp cho developer để giải quyết những công việc đó nhằm tiết kiệm thời gian



LTV Mã nguồn mở PHP – Module 1

3

Nội dung



1. Giới thiệu
2. Các hàm toán học
3. Các hàm chuỗi
4. Các hàm thời gian



Các hàm toán học



□ Giới thiệu

- PHP cung cấp rất nhiều hàm phục vụ cho việc tính toán
- Các hàm toán học có thể xử lý các giá trị trong phạm vi của số nguyên và số thực





Các hàm toán học

□ Các hàm thông dụng

Vấn đề:

Cần có các giá trị để phục vụ cho việc tính toán

Ví dụ:

- Tính diện tích, chu vi hình tròn
- Tìm giá trị tuyệt đối của x
- Tính y^3
- Làm tròn Diem_trung_binh
- Tính căn bậc 2 của z
- Phát sinh một số ngẫu nhiên trong khoảng [1, 100]

Giải quyết:

Sử dụng hàm toán học trong thư viện hàm của PHP

```
// Tính S, P hình tròn bán kính r
$S = pi() * pow($r, 2);
$P = 2 * pi() * $r;

// Tìm giá trị tuyệt đối của x
$Gttd_x = abs($x);

// Tính y³
$Lap_phuong_y = pow($y, 3);

// Làm tròn điểm trung bình
$Diem_trung_binh =
round($Diem_trung_binh, 2);

// Tính căn bậc 2 của z
$Can_bac_2_z = sqrt($z);

// Phát sinh số ngẫu nhiên trong
khoảng [1, 100]
$So_ngau_nhien = rand(1, 100);
```



Nội dung

1. Giới thiệu
2. Các hàm toán học
3. Các hàm chuỗi
4. Các hàm thời gian





Các hàm chuỗi

□ Giới thiệu

- PHP cung cấp rất nhiều hàm phục vụ cho việc xử lý và hiển thị chuỗi



LTV Mã nguồn mở PHP – Module 1

8

Các hàm chuỗi



□ Các hàm thông dụng

Vấn đề:

Cần xử lý các chuỗi để có dữ liệu chuỗi phù hợp và hiển thị chuỗi có định dạng

Ví dụ:

- Loại bỏ các khoảng trắng thừa trong chuỗi `Ho_ten`
- Tính chiều dài của chuỗi `Ho_ten`
- So sánh hai chuỗi `str1` và `str2`
- Tìm vị trí của chuỗi `str1` trong chuỗi `str`

```
// Loại khoảng trắng thừa trong
Ho_ten
$Ho_ten = trim($Ho_ten);
// Tính chiều dài của chuỗi Ho_ten
$len_Ho_ten = strlen($Ho_ten);
// So sánh hai chuỗi str1 và str2
$Ket_qua = strcmp($str1, $str2);
// trả về 0, -1, 1
// Tìm vị trí của str1 trong str
$Vi_tri = strpos($str, $str1);
// trả về vị trí đầu tiên hoặc
false nếu không tìm thấy
```



LTV Mã nguồn mở PHP – Module 1

9



Các hàm chuỗi

□ Các hàm thông dụng

- Thay thế chuỗi str1 trong chuỗi str bằng chuỗi str2
- Lấy chuỗi con trong chuỗi cha str
- Kết hợp chuỗi str1 và str2 thành chuỗi str
- Tách chuỗi str thành các chuỗi con

Giải quyết:

Sử dụng hàm chuỗi trong thư viện hàm của PHP

```
// Thay thế str1 trong str bằng str2
$str_new = str_replace($str1,
    $str2, $str);

// lấy chuỗi con trong chuỗi cha str
$str1 = substr($str, $start,
    [$len]);

// Kết hợp chuỗi str1 & str2 thành
chuỗi str, str1 & str2 cách nhau bằng
" "

$str = implode(" ", array($str1,
    $str2));

// Tách chuỗi str thành các chuỗi
con, dấu phân cách là " "

$Mang_chuoi_con = explode("",
    $str);
```



Nội dung

1. Giới thiệu
2. Các hàm toán học
3. Các hàm chuỗi
4. Các hàm thời gian





Các hàm thời gian

□ Giới thiệu

- PHP cung cấp rất nhiều hàm phục vụ cho việc xử lý và hiển thị thời gian (date và time)
- Các hàm thời gian cho phép lấy được thời gian từ server nơi đặt code PHP.
- Có thể sử dụng hàm thời gian để định dạng thời gian theo nhiều cách khác nhau.
- *Lưu ý:* Các hàm thời gian phụ thuộc vào việc thiết lập vùng của server



LTV Mã nguồn mở PHP – Module 1

12

Các hàm thời gian

□ Các hàm thông dụng

Vấn đề:

Cần có các giá trị thời gian để phục vụ cho việc tính toán và hiển thị

Ví dụ:

- Kiểm tra ngày hợp lệ
- Định dạng ngày hiện tại
- Đổi thời gian sang đơn vị giây
- Lấy thời gian hiện tại

Giải quyết:

Sử dụng hàm thời gian trong thư viện hàm của PHP

```
// Kiểm tra ngày hợp lệ
$Hop_le = checkdate($tháng,
$ngày, $năm); // trả về true hoặc
false

// Định dạng ngày hiện tại
$Ngày_hien_tai = date("d/m/Y
h:i:s");

// Đổi thời gian sang đơn vị giây
$Tong_so_giay = mktime(0, 0, 0,
1, 1, 2014); // tính từ 01/01/1970

// Lấy thời gian hiện tại // tính
bằng giây
$Time = time();
```



LTV Mã nguồn mở PHP – Module 1

13



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC

LTV MÃ NGUỒN MỞ PHP

Module 1 – Bài 5: *Xử lý tập tin*

Ngành LT & CSDL

t3h.vn

2014



2014



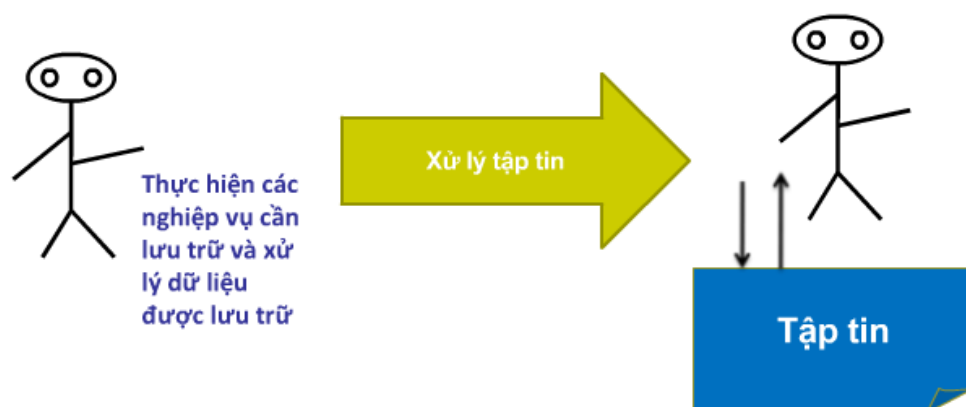
Nội dung

1. Giới thiệu
2. Các thao tác trên tập tin
3. Upload tập tin



Giới thiệu

□ Ý nghĩa sử dụng



Có những dữ liệu cần phải được lưu trữ và xử lý thường xuyên => Sử dụng tập tin



Nội dung

1. Giới thiệu
2. Các thao tác trên tập tin
3. Upload tập tin



Các thao tác trên tập tin

- ☐ Mở tập tin
- ☐ Đóng tập tin
- ☐ Đọc tập tin
- ☐ Ghi tập tin
- ☐ Các hàm xử lý tập tin





Các thao tác trên tập tin

❑ Mở tập tin

- Cú pháp: **fopen(filename, filemode)**
- Tham số đầu tiên chứa tên của file cần mở và tham số thứ hai chứa chế độ mở file (filemode)
- Chú ý: khi mở file bằng **fopen()** thì tất cả nội dung trong file sẽ được gán vào biến có tên là **\$f**
 - Ví dụ: mở file **LienLac.txt** trong thư mục **du_lieu** với chế độ chỉ đọc.

```
$f = fopen("du_lieu/LienLac.txt", "r");
```



LTV Mã nguồn mở PHP – Module 1

6

Các thao tác trên tập tin



❑ Mở tập tin

- Filemode:

Filemode	Mô tả
r	Chỉ đọc. Bắt đầu từ đầu của tập tin
r+	Đọc/ Ghi. Bắt đầu từ đầu tập tin
w	Chỉ ghi. Mở và xóa nội dung đang có của tập tin hoặc tạo tập tin mới nếu tập tin không tồn tại.
w+	Đọc/ Ghi. Mở và xóa nội dung đang có của tập tin hoặc tạo tập tin mới nếu tập tin không tồn tại.
a	Thêm vào. Mở và ghi nội dung mới vào cuối tập tin hoặc tạo tập tin mới nếu tập tin không tồn tại.
a+	Đọc và thêm. Bảo toàn nội dung tập tin và ghi vào cuối tập tin
x	Chỉ ghi. Tạo ra một tập tin mới. Trả về FALSE và thông báo lỗi nếu tập tin này đã tồn tại
x+	Đọc/ Ghi. Tạo một tập tin mới. Trả về FALSE và thông báo lỗi nếu tập tin này đã tồn tại



LTV Mã nguồn mở PHP – Module 1

7



Các thao tác trên tập tin

❑ Đóng tập tin

- Cú pháp: **fclose (\$f)**
 - Tham số \$f là biến đã khai báo ở hàm fopen()
- Kết quả trả về là TRUE nếu đóng được file, ngược lại kết quả trả về là FALSE
 - Ví dụ: đóng tập tin **LienLac.txt** trong thư mục **du_lieu**.
`fclose ($f) ;`



Các thao tác trên tập tin

❑ Đọc tập tin

- Kiểm tra trạng thái cuối tập tin
 - Cú pháp: **feof (\$f)**
 - Tham số \$f là biến khai báo ở hàm fopen().
 - Rất hữu hiệu cho việc đọc dữ liệu trong tập tin khi không biết chiều dài cụ thể của tập tin.
 - Kết quả trả về là TRUE nếu con trỏ ở cuối tập tin.
 - Chú ý: không thể đọc file khi mở file bằng chế độ w, a và x





Các thao tác trên tập tin

❑ Đọc tập tin

- Duyệt và đọc từng dòng nội dung trong tập tin bằng `fgets()`
 - **Cú pháp:** `fgets($f [, int độ_dài])`
 - Tham số `$f` là biến khai báo ở hàm `fopen()`.
 - Kết quả trả về là một chuỗi có chiều dài lên đến `<độ_dài> - 1` (độ dài có đơn vị tính là byte), nếu không quy định `<độ_dài>` thì hàm này sẽ tự thiết lập độ dài mặc định là 1024 byte.



Các thao tác trên tập tin



❑ Đọc tập tin

- Duyệt và đọc từng dòng nội dung trong tập tin bằng `fgets()`
 - Ví dụ: Đọc và in toàn bộ nội dung tập tin **LienLac.txt**

```
$f = fopen("hoa.txt", "r") or exit ("Không mở được tập tin này");
```

```
while(!feof($f))
```

```
{     $noi_dung = fgets($f);
```

```
     echo $noi_dung . "<br>";
```

```
}
```

```
fclose($f);
```





Các thao tác trên tập tin

❑ Đọc tập tin

- Đọc và duyệt toàn bộ nội dung của tập tin bằng `readfile()`
 - Cú pháp: `readfile("đường dẫn và tên của file")`
 - Kết quả trả về của hàm là số byte của file nếu đọc file thành công, ngược lại kết quả trả về là `FALSE`



Các thao tác trên tập tin

❑ Ghi tập tin

- Cú pháp: `fwrite($f, nội_dung [, độ_dài]);`
- Chú ý: độ dài là tham số tùy chọn: quy định số byte tối đa được ghi vào file của nội dung
- Trước khi ghi chuỗi vào file cần phải định dạng lại chuỗi đó theo nhu cầu xuất dữ liệu trở lại khi đọc file.
- Cách thức định dạng: tự thiết lập. Tuy nhiên, có một số định dạng được quy định sẵn như sau:
 - `\t`: nhảy tab
 - `\n`: xuống dòng





Các thao tác trên tập tin

❑ Ghi tập tin

▪ Ví dụ:

```
$noi_dung = "Ghi nội dung vào tập tin";

// thực hiện việc mở và ghi file

$f = fopen("$ten_file", "a");

fwrite($f, $noi_dung);

fclose($f);
```



LTV Mã nguồn mở PHP – Module 1

14

Các thao tác trên tập tin

❑ Các hàm xử lý tập tin

Vấn đề:

Cần xử lý nội dung tập tin phục vụ cho việc lưu trữ và hiển thị

Ví dụ:

- Kiểm tra tập tin LienLac.txt có tồn tại hay không?
- Lấy kích thước tập tin LienLac.txt
- Xóa tập tin LienLac.txt
- Kiểm tra LienLac.txt có phải là tập tin hay không?

Giải quyết:

Sử dụng hàm tập tin trong thư viện hàm của PHP

```
$file = "LienLac.txt";
// Kiểm tra tập tin tồn tại
$isExists = file_exists($file);
// trả về true hoặc false

// Lấy kích thước tập tin
$FileSize = filesize($file); // trả về kích thước hoặc false

// Xóa tập tin
$isDelete = unlink($file); // trả về true hoặc false

// Kiểm tra tập tin
$isFile = is_file($file); // trả về true hoặc false
```



LTV Mã nguồn mở PHP – Module 1

15

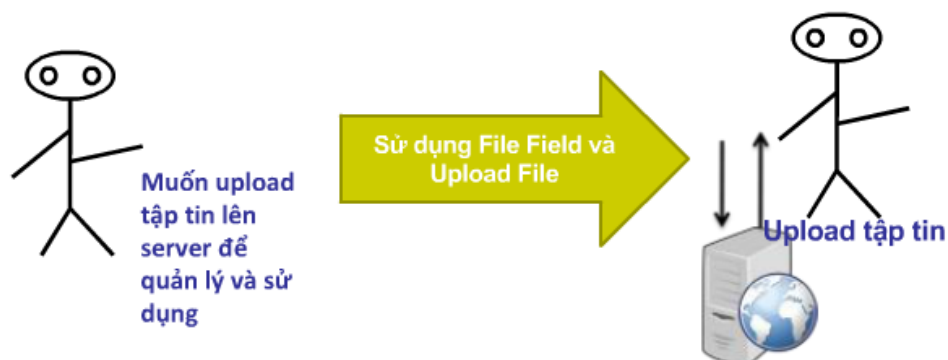
Nội dung

1. Giới thiệu
2. Các thao tác trên tập tin
3. Upload tập tin



Upload tập tin

□ Giới thiệu



File Field và việc Upload File rất cần thiết vì giúp người dùng có thể đưa tập tin lên sever để quản lý và sử dụng





Upload tập tin

❑Thực hiện

```
<form action="upload_file.php" method="post"
  enctype="multipart/form-data">
  <label for="file">Tên file:</label>
  <input type="file" name="file_upload" id="file" />
  <br />
  <input type="submit" name="submit" value="Upload
  File" />
</form>
```



LTV Mã nguồn mở PHP – Module 1

18



Upload tập tin

❑Thực hiện

```
<?php
// kiểm tra file được Upload có hợp lệ hay không
...
// nếu hợp lệ: dùng hàm move_uploaded_file để di
chuyển file upload vào thư mục định sẵn là thư mục
upload
move_uploaded_file($_FILES["file_upload"]["tmp_name"]
, "upload/" . $_FILES["file_upload"]["name"]);
?>
```



LTV Mã nguồn mở PHP – Module 1

19



Upload tập tin

□ Sử dụng thông tin của tin: \$_FILES

- \$_FILES["filefield"]["name"] – tên File
- \$_FILES["filefield"]["type"] – kiểu của File
- \$_FILES["filefield"]["size"] – kích cỡ của File
- \$_FILES["filefield"]["tmp_name"] – tên tạm của File
- \$_FILES["filefield"]["error"] – lỗi của File





Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC

LTV MÃ NGUỒN MỞ PHP

Module 1 – Bài 6: *Mảng cơ bản*

Ngành LT & CSDL

www.t3h.vn



2014

2014



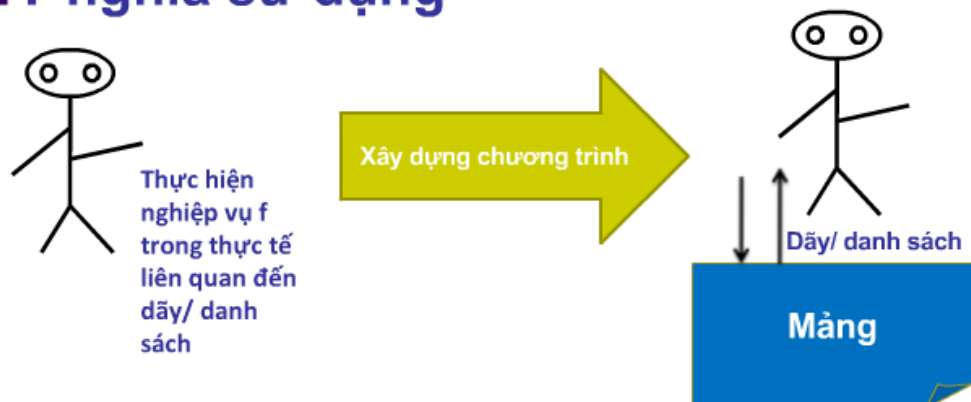
Nội dung

1. Giới thiệu
2. Khai báo và sử dụng
3. Thao tác trên mảng
4. Các hàm xử lý trên mảng



Giới thiệu

□ Ý nghĩa sử dụng



Mảng là một **loại biến đặc biệt**, bao gồm một **dãy các ô nhớ có nhiều ô nhớ con** cho phép biểu diễn thông tin dạng danh sách trong thực tế

Các phần tử trong mảng có thể có **kiểu dữ liệu khác nhau**





Giới thiệu

□ Ví dụ

- Cho dãy các số nguyên a bao gồm 1, 3, 4, -6, 7
 - Tính tổng các số nguyên của a
- Cho dãy các phân số Ps bao gồm $\frac{1}{3}$, $\frac{2}{5}$, $\frac{6}{7}$
 - Tính Tích các phân số của Ps



Nội dung

1. Giới thiệu
2. Khai báo và sử dụng
3. Thao tác trên mảng
4. Các hàm xử lý trên mảng





Khai báo và sử dụng

□ Khai báo mảng

- Cú pháp: `$tên_mảng = array();`
- Ví dụ: khai báo mảng a
 - `$a = array();`



Khai báo và sử dụng

□ Khởi tạo mảng

- Cú pháp: `$tên_mảng = array([khóa=>] giá trị, ...,);`
- Các phần tử trong mảng cách nhau bằng dấu “,”
- Với:
 - khóa: số nguyên dương/ chuỗi
 - Nếu khóa là chuỗi: dùng cặp nháy đôi "giá trị khóa" hoặc cặp nháy đơn 'giá trị khóa'
 - Mặc định, khóa tự động phát sinh, với phần tử đầu tiên của mảng khóa có giá trị là 0, phần tử thứ hai của mảng khóa có giá trị là 1, ...





Khai báo và sử dụng

□ Truy xuất phần tử trong mảng

- Cú pháp: `$tên_mảng[<khóa>]`
- Ví dụ:

```
$mang1 = array(1, 5, 7);
```

```
$gia_tri_1 = $mang1[0]; //→ 1
```

```
$mang2 = array(1=>"Một", 2=>"Hai")
```

```
$gia_tri_3 = $mang[1]; //→ "Một"
```

```
$mang2[3] = "Ba"; //Gán giá trị
```



LTV Mã nguồn mở PHP – Module 1

8



Nội dung

1. Giới thiệu
2. Khai báo và sử dụng
3. Thao tác trên mảng
4. Các hàm xử lý trên mảng



LTV Mã nguồn mở PHP – Module 1

9



Thao tác trên mảng

❑ Đếm số phần tử trong mảng

- Cú pháp: `count($tên_mảng)`

- Ví dụ:

```
$mang = array(1, 2, 3, 4, 5, 6);
```

```
$so_phan_tu = count($mang); → 6
```



Thao tác trên mảng

❑ Duyệt mảng có khóa tự động

- Cú pháp:

```
for($i = 0; $i < $số_phần_tử; $i++)
{
    Xử lý các phần tử trong mảng (đọc, tính toán, thay đổi...)
    với mỗi phần tử: $tên_mảng[$i]
}
```

- Ví dụ:

```
$mang = array(1, 2, 3, 4, 5, 6);
```

```
$n = count($mang);
```

```
for($i = 0; $i < $n; $i++)
```

```
    echo "\t" . $mang[$i];
```

```
→ 1 2 3 4 5 6
```





Thao tác trên mảng

❑ Duyệt mảng có khóa do người dùng tạo

- Duyệt để lấy và xử lý giá trị của các phần tử trong mảng:

```
foreach ($tên_mảng as $giá_trị)
{
    Xử lý các giá trị trong mảng (đọc, tính toán, thay đổi...)
    với mỗi phần tử: $giá_trị
}
```

- Ví dụ:

```
$mang = array(1=>"Một", 2=>"Hai", 3=>"Ba",
4=>"Bốn", 5=>"Năm");
foreach ($mang as $gia_tri)
    echo "\t $gia_tri";
→ Một Hai Ba Bốn Năm
```



LTV Mã nguồn mở PHP – Module 1

12



Thao tác trên mảng

❑ Duyệt mảng có khóa do người dùng tạo

- Duyệt để lấy cả giá trị của khóa và giá trị của phần tử

```
foreach($tên_mang as $khóa=>$giá_trị)
{
    Xử lý
    với mỗi phần tử: $khóa=>$giá_trị
}
```

- Ví dụ:

```
$mang = array("mot"=>1, "hai"=>2, "ba"=>3,
"bon"=>4, "nam"=>5);
foreach ($mang as $khoa=>$gia_tri)
    echo "[ $khoa ] => $gia_tri, ";
→ [mot] => 1, [hai] => 2, [ba] => 3, [bon] =>
4 [nam] => 5
```



LTV Mã nguồn mở PHP – Module 1

13



Thao tác trên mảng

❑ Tạo mảng từ chuỗi: Dùng explode()

- Chú ý: cần có quy ước cấu trúc cho chuỗi mảng
- Ví dụ: Quy ước khi nhập (tạo chuỗi mảng là các phần tử cách nhau bằng dấu “,”)

```
$chuoi_mang = '1, 6, 3, 12, 8, 2';
//tạo mảng
$mang = explode(',', $chuoi_mang);
```



Thao tác trên mảng

❑ Xuất mảng ra chuỗi: Dùng implode()

- Chú ý: cần có quy ước cấu trúc cho chuỗi mảng
- Ví dụ: Quy ước khi xuất (xuất chuỗi mảng gồm các phần tử cách nhau bằng dấu “,”)

```
$mang = array(1, 6, 3, 12, 8, 2);
//tạo chuỗi kết quả
$mang_chuoi = implode(',', $mang);
echo $mang_chuoi;
```





Nội dung

1. Giới thiệu
2. Khai báo và sử dụng
3. Thao tác trên mảng
4. Các hàm xử lý trên mảng



Các hàm xử lý trên mảng



Vấn đề:

Cần xử lý mảng một cách nhanh chóng

Ví dụ:

- Tìm kiếm một giá trị trên mảng \$mang
- Ghép mảng 1 với mảng 2 thành một mảng

Giải quyết:

Sử dụng hàm mảng trong thư viện hàm của PHP

```
$mang = array(0 => 'xanh', 1 => 'đỏ', 2
=> 'tím', 3 => 'vàng');
// Tìm kiếm phần tử có giá trị "đỏ"
$khoa_do = array_search('đỏ', $mang); →
1, nếu không thấy trả về NULL
// Ghép mảng 1 với mảng 2 thành một mảng
$mang1 = array("màu" => "đỏ", 2, 4);
$mang2 = array("a", "b", "màu" => "xanh",
"hình" => "tròn", 4);
$mang_chung = array_merge($mang1,
$mang2); // khi các mảng dùng để ghép có
khóa trùng nhau thì mảng ghép sẽ chỉ lấy
phần tử có khóa trùng của mảng cuối
cùng:
Array ( [màu] => xanh, [0] => 2, [1] =>
4, [2] => a, [3] => b, [hình] => tròn
[4] => 4 )
```





Các hàm xử lý trên mảng

....

- Đếm số lần xuất hiện của các phần tử trong mảng \$mang
- Tạo mảng duy nhất từ mảng hiện có
- Tìm các giá trị khác nhau của mảng 1 so với mảng 2

Giải quyết:

Sử dụng hàm mảng trong thư viện hàm của PHP

```
$mang = array(1, "hello", 1, "world",
"hello", 2, "Chào", 1);

// Đếm số lần xuất hiện của các phần tử
trong mảng

$mang_slxh = array_count_values($mang); //
Array ([1] => 3; [hello] => 2; [world] => 1
; [2] => 1 ; [Chào] => 1)

// Tạo mảng duy nhất từ mảng 1
$mang1 = array(1, 3, 1, 2, 5, 1, 3, 4);
$mang_duy_nhat= array_unique($mang1); -> 1,
3, 2, 5, 4

// Tìm giá trị khác nhau của mảng 1 so với
mảng 2
$mang_1 = array("a"=>"xanh", "đỏ", "tím",
"vàng");
$mang_2 = array("b"=>"xanh", "vàng", "đỏ");
$mang_con_1 = array_diff($mang_1, $mang_2);
// Array ( [1] => tím )
```





Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC

LTV MÃ NGUỒN MỞ PHP

Module 1 – Bài 7: *LT hướng đối tượng cơ bản*

Ngành LT & CSDL

www.t3h.vn



2014

2014



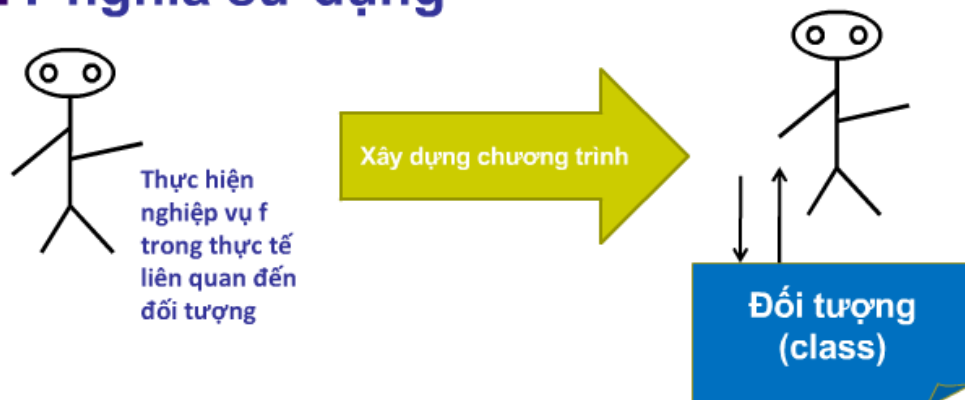
Nội dung

1. Tổng quan đối tượng
2. Xây dựng lớp đối tượng (class)
3. Sử dụng lớp đối tượng



Tổng quan đối tượng

□ Ý nghĩa sử dụng



Đối tượng (Object) là những thực thể tồn tại trong thế giới thực.





Tổng quan đối tượng

□ Ví dụ

- Con người
 - Sinh viên Nguyễn Văn An
 - Nhân viên Trần Thị Thảo
- Đồ vật
 - Phòng học C41
 - Máy in Laser Jet 4300
- Chứng từ
 - Hóa đơn HD01
 - Đơn đặt hàng DDH_14022008_01
- Một chiếc xe hơi cụ thể với các thông tin về chiếc xe: Biển số xe, Hiệu xe, Màu sơn, Hãng sản xuất, Năm sản xuất



Tổng quan đối tượng

□ Tiếp cận hướng đối tượng

- Là kỹ thuật cho phép biểu diễn tự nhiên các đối tượng trong thực tế với các đối tượng bên trong chương trình



Tổng quan đối tượng



☐ Nhân cách hóa đối tượng



LTV Mã nguồn mở PHP – Module 1

6

Tổng quan đối tượng



☐ Ví dụ: bài toán Giải PT bậc 2



LTV Mã nguồn mở PHP – Module 1

7



Tổng quan đối tượng

❑ Ví dụ: Tính tiền lương cho nhân viên



LTV Mã nguồn mở PHP – Module 1

8



Tổng quan đối tượng

❑ Lớp đối tượng

- Là một khái niệm trong Lập trình hướng đối tượng, mô tả những **thực thể có chung tính chất và hành vi**. Class định nghĩa những thuộc tính và hành vi được dùng cho những đối tượng của lớp đó

❑ Kết quả của sự **TRỪU TƯỢNG HOÁ (Abstraction)** các đối tượng:

- Cùng loại
- Cùng các thông tin mô tả về đối tượng



LTV Mã nguồn mở PHP – Module 1

9

Tổng quan đối tượng



☐ Lớp đối tượng

- Ví dụ: Lớp XE_OTO



Tổng quan đối tượng



☐ Lớp đối tượng

- Ví dụ: lớp NHAN_VIEN





Nội dung

1. Tổng quan đối tượng
2. Xây dựng lớp đối tượng (class)
3. Sử dụng lớp đối tượng



Xây dựng lớp đối tượng

- ☐ Tạo lớp
- ☐ Khai báo thuộc tính
- ☐ Xây dựng phương thức
- ☐ Minh họa





Xây dựng lớp đối tượng

❑ Tạo lớp

- Cú pháp:

```
class <tên lớp>
{ // khai báo các thuộc tính của lớp
  // gán và lấy giá trị của thuộc tính
  // các phương thức của lớp
}
```



Xây dựng lớp đối tượng

❑ Khai báo thuộc tính

- Thuộc tính: thành phần lưu trữ các tính chất, đặc điểm của đối tượng.

- Cú pháp

```
var <tên thuộc tính 1>;
```

...

- Chú ý: Có thể thiết lập những giá trị mặc định ban đầu cho tất cả các thuộc tính được tạo ra từ lớp đó





Xây dựng lớp đối tượng

❑ Ví dụ

```
class PHAN_SO
{
    var $tu_so = 1;

    var $mau_so = 1;

    ...

}
```



LTV Mã nguồn mở PHP – Module 1

16



Xây dựng lớp đối tượng

❑ Xây dựng phương thức

- Phương thức: Là chức năng mà đối tượng có thể thực hiện; có thể có hoặc không có giá trị trả về

- Cú pháp:

```
function <tên phương thức> (các tham số truyền
    vào nếu có)
{
    // khối lệnh
}
```



LTV Mã nguồn mở PHP – Module 1

17



Xây dựng lớp đối tượng

❑ Ghi chú

- Trong các phương thức của lớp, có thể truy cập các thuộc tính hay phương thức thông qua con trỏ **\$this**
- Con trỏ **\$this** được dùng để chỉ lớp hiện tại đang làm việc.
- Ví dụ: nếu một thuộc tính của lớp phân số có tên là \$tu_so thì có thể tham chiếu đến nó như sau: **\$this->tu_so**



Xây dựng lớp đối tượng

❑ Minh họa: Xây dựng lớp Phân số





Xây dựng lớp đối tượng

❑ Minh họa: Xây dựng lớp Phân số

```
class PHAN_SO
{
    // khai báo các thuộc tính
    var $tu_so;
    var $mau_so;
    // gán và lấy giá trị của các thuộc tính
    // xây dựng các phương thức cần thiết cho lớp
    phân số
}
```



LTV Mã nguồn mở PHP – Module 1

20



Xây dựng lớp đối tượng

❑ Minh họa: Xây dựng lớp Phân số

```
class PHAN_SO
{ ...
    // xây dựng các phương thức cần thiết cho lớp
    phân số
    function khai_tao_phan_so($ptu_so,$pmau_so)
    {
        $this->tu_so = $ptu_so;
        $this->mau_so = $pmau_so;
    }
    ...
}
```



LTV Mã nguồn mở PHP – Module 1

21



Xây dựng lớp đối tượng

❑ Minh họa: Xây dựng lớp Phân số

```
class PHAN_SO
{
    ...

    function USCLN($a, $b)
    {
        $so_nho = ($a < $b) ? $a : $b;
        for($i = $so_nho; $i > 0; $i--)
            if( ($a % $i) == 0 && ($b % $i) == 0 )
            {
                return $i;
                break;
            }
    }
    ...
}
```



LTV Mã nguồn mở PHP – Module 1

22



Xây dựng lớp đối tượng

❑ Minh họa: Xây dựng lớp Phân số

```
class PHAN_SO
{
    ...

    function Toi_gian_phan_so()
    {
        $uscln = USCLN($this->tu_so, $this->mau_so);
        $this->tu_so = $this->tu_so / $uscln;
        $this->mau_so = $this->mau_so / $uscln;
        return PHAN_SO;
    }
    ...
}
```



LTV Mã nguồn mở PHP – Module 1

23



Xây dựng lớp đối tượng

❑ Minh họa: Xây dựng lớp Phân số

● Tính tổng hai phân số

```
class PHAN_SO
{
    ...
    function tong($ptu_so, $pmau_so)
    {
        $phan_so = new PHAN_SO();
        $phan_so->khoi_tao($ptu_so, $pmau_so);
        $phan_so->tu_so = ($this->tu_so * $phan_so->mau_so)
            + ($phan_so->tu_so * $this->mau_so);
        $phan_so->mau_so = $this->mau_so * $phan_so->mau_so;
        $phan_so->Toi_gian_phan_so();
        return $phan_so;
    }
    ... (tiếp tục thực hiện các phương thức khác của phân số)
}
```



Nội dung



1. Tổng quan đối tượng
2. Xây dựng lớp đối tượng (class)
3. Sử dụng lớp đối tượng





Sử dụng lớp đối tượng

- ☐ Khởi tạo đối tượng
- ☐ Gán giá trị cho các thuộc tính của lớp
- ☐ Gọi sử dụng các phương thức của lớp



Sử dụng lớp đối tượng

☐ Khởi tạo đối tượng

- Cú pháp: `$<tên biến đối tượng> = new <tên lớp>();`

- Ví dụ:

```
$phan_so = new PHAN_SO();
```





Sử dụng lớp đối tượng

□ Gán giá trị cho các thuộc tính của lớp

- Cú pháp: `$<tên biến đối tượng>-><tên thuộc tính> = <giá trị>;`

- Ví dụ:

```
$phan_so->tu_so = 2;
```

```
$phan_so->mau_so = 4;
```



Sử dụng lớp đối tượng

□ Gọi sử dụng phương thức của lớp

- Phương thức không có giá trị trả về

- Cú pháp: `$<tên biến đối tượng>-><tên phương thức>(<các giá trị truyền vào nếu có>;`

- Ví dụ:

```
$phan_so->khoi_tao_phan_so($ptu_so,$pmau_so);
```





Sử dụng lớp đối tượng

□ Gọi sử dụng phương thức của lớp

● Phương thức có giá trị trả về

- **Cú pháp:** `$<tên biến nhận giá trị> = $<tên biến đối tượng>-> <tên phương thức>(các giá trị truyền vào nếu có);`
- **Ví dụ:**

```
$phan_so_2 = New PHAN_SO();
$phan_so_2 = $phan_so->tong($a,$b);
```



Sử dụng lớp đối tượng

□ Gọi sử dụng phương thức của lớp

● Phương thức có giá trị trả về

- **Cú pháp:** `$<tên biến nhận giá trị> = $<tên biến đối tượng>-> <tên phương thức>(các giá trị truyền vào nếu có);`
- **Ví dụ:**

```
$phan_so_2 = New PHAN_SO();
$phan_so_2 = $phan_so->tong($a,$b);
```





Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC

LTV MÃ NGUỒN MỞ PHP

Module 1 – Bài 8: *Form và các điều khiển cơ sở*

Ngành LT & CSDL

www.t3h.vn



2014

2014



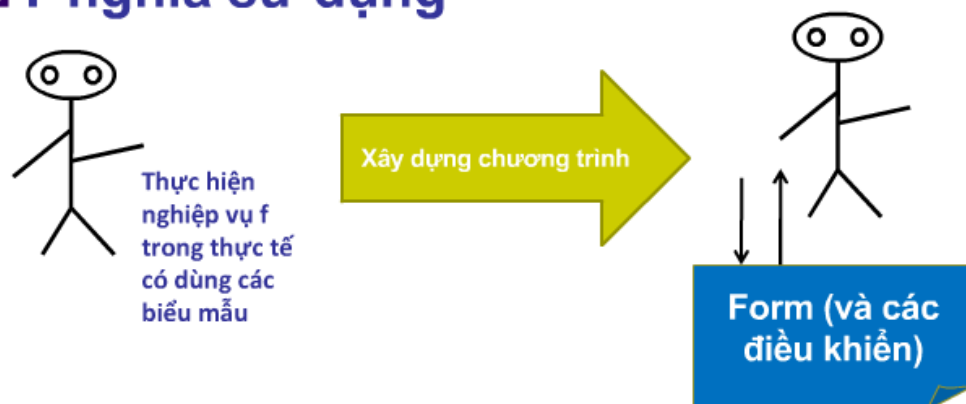
Nội dung

1. Form
2. Các điều khiển cơ sở
3. Đọc giá trị từ điều khiển form



Form

□ Ý nghĩa sử dụng



Form là một đối tượng dành cho người dùng tương tác với chương trình để thực hiện một công việc nào đó



Form



□ Đặc điểm

- Form được dùng để gửi yêu cầu về server.
- Các thuộc tính cơ bản của Form



Form



□ Thuộc tính action

- Quy định trang xử lý yêu cầu khi người dùng submit form.

- Code HTML

```
<form action="vi_du_c2.php"  
method="post" name="frm1">
```



Nội dung

1. Form
2. Các điều khiển cơ sở
3. Đọc giá trị từ điều khiển form



Các điều khiển cơ sở

- ☐ Đặc điểm chung
- ☐ TextFiled/TextArea
- ☐ Button
- ☐ Checkbox
- ☐ RadioButton/RadioGroup





Các điều khiển cơ sở

❑ Đặc điểm chung

- Nằm trong thẻ Form
- Tên được thiết lập trong thuộc tính **name** của đối tượng.
- Có **giá trị ban đầu** (initial value) và **giá trị hiện tại** (current value), kiểu chuỗi.



Các điều khiển cơ sở



❑ Đặc điểm chung

- Giá trị ban đầu không thay đổi, vì vậy khi reset Form, các giá trị hiện tại sẽ được đặt lại thành giá trị ban đầu.
- Có giá trị hiện tại (current value) khi Form submit
- Tạo một đối tượng: vào **menu Insert => Form => chọn đối tượng muốn tạo**. Hoặc cũng có thể tạo ra đối tượng bằng cách viết thẻ lệnh.





Các điều khiển cơ sở

□ TextField

- Là ô nhập liệu và hiển thị dữ liệu. Có 3 loại:
 - Single line: hiển thị và nhập liệu trên một dòng văn bản.
 - Multi line: hiển thị và nhập liệu nhiều dòng văn bản
 - Password: hiển thị dấu * thay cho các ký tự trong TextField
- Dữ liệu nhập vào cho TextField là giá trị hiện tại



LTV Mã nguồn mở PHP – Module 1

10

Các điều khiển cơ sở



□ TextField

- Vào Menu Insert => Form => chọn TextFiled
- Hoặc dùng thẻ input để tạo


```
<input type="text" name="textfield" value="" />
```
- Thiết lập các thuộc tính cơ bản như tên (name), độ rộng (size), số ký tự tối đa (maxlength)...



LTV Mã nguồn mở PHP – Module 1

11



Các điều khiển cơ sở

□ TextArea

- Là TextField dạng multi line, dùng để nhập liệu trên nhiều dòng (multi line). Với:
 - Num lines: số dòng văn bản được hiển thị trên Textarea
 - Wrap: quy định việc hiển thị của văn bản có/không được phép tự động xuống dòng khi kích thước ngang của điều khiển không đủ để hiển thị nội dung văn bản. Mặc định là tự động xuống dòng.



Các điều khiển cơ sở

□ TextArea

- Vào Menu Insert => Form => chọn TextArea
- Hoặc dùng thẻ textarea để tạo:


```
<textarea name="textarea" cols=""
rows="3"> </textarea>
```
- Thiết lập các thuộc tính cơ bản như (name), độ rộng (cols), số dòng hiển thị (rows)...





Các điều khiển cơ sở

□ Button

- Có:
 - Submit button: khi nhấn button này thì thông tin sẽ postback về server. Trong một Form có thể có một hay nhiều Submit button.
 - Reset button: khi nhấn button này thì tất cả các đối tượng trên Form sẽ được reset trở lại giá trị ban đầu.



Các điều khiển cơ sở

□ Button

- Vào Menu Insert => Form => chọn Button
- Hoặc dùng thẻ input để tạo:


```
<input type="submit" name="Submit" value="Submit" />
```
- Thiết lập các thuộc tính cơ bản như tên (name), và giá trị (value), kiểu (type)





Các điều khiển cơ sở

☐ Checkbox

- Là đối tượng có hai trạng thái on/off (chọn/ không chọn). Nếu trạng thái checked được chọn thì Checkbox sẽ có giá trị là “checked”.
- Khi có nhiều Checkbox trong Form thì tại một thời điểm chúng ta có thể chọn một hay nhiều Checkbox (cũng có thể không chọn một Checkbox nào)



LTV Mã nguồn mở PHP – Module 1

16



Các điều khiển cơ sở

☐ Checkbox

- Vào Menu Insert => Form => chọn Checkbox
- Hoặc dùng thẻ input để tạo:

```
<input name="checkbox" type="checkbox" value="" checked />
```
- Thiết lập các thuộc tính cơ bản như tên (name), giá trị (value), trạng thái ban đầu.



LTV Mã nguồn mở PHP – Module 1

17



Các điều khiển cơ sở

☐ RadioButton

- Tương tự như Checkbox
- Nếu trên Form có nhiều RadioButton mà chúng có tên khác nhau thì mỗi RadioButton là một đối tượng độc lập có hai trạng thái on/off.



Các điều khiển cơ sở

☐ RadioButton

- Vào Menu Insert => Form => chọn RadioButton

- Hoặc dùng thẻ input để tạo:

```
<input name="radio" type="radio"
value="1" />
```

- Thiết lập các thuộc tính cơ bản: tên (name), loại (type), giá trị (value)





Các điều khiển cơ sở

☐ RadioGroup

- Là một nhóm các RadioButton có cùng tên
- Khi một RadioButton đã được chọn (ở trạng thái on) thì tất cả các RadioButton cùng tên khác sẽ không được chọn (ở trạng thái off).



Các điều khiển cơ sở

☐ RadioGroup

- Vào Menu Insert => Form => chọn RadioGroup
 - Thêm các RadioButton vào RadioGroup, đặt tên cho RadioGroup, thiết lập nội dung và giá trị cho các Radiobutton





Các điều khiển cơ sở

☐ RadioGroup

- Hoặc dùng thẻ input để tạo

```
<label>
  <input type="radio" name="RadioGroup1" value="1"
  />
  Radio1
</label>
<label>
  <input type="radio" name="RadioGroup1" value="0"
  />
  Radio2
</label>
```



Nội dung

1. Form
2. Các điều khiển cơ sở
3. Đọc giá trị từ điều khiển form





Đọc giá trị từ điều khiển form

☐\$_POST

- Được dùng để lấy giá trị của các điều khiển trên Form thông qua phương thức POST.
- Thông tin được gửi từ Form với phương thức POST không giới hạn lượng thông tin gửi đi và sẽ không được hiển thị trên địa chỉ URL nên người dùng không thể thấy được



Đọc giá trị từ điều khiển form

☐\$_POST

- Cú pháp: lấy giá trị của một đối tượng trên Form sau khi Form submit:

`$_POST["tên điều khiển"]`

- Ví dụ: lấy giá trị TextField tên là txtTen

`$ten = $_POST["txtTen"] ;`





Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC

LTV MÃ NGUỒN MỞ PHP

Module 1 – Bài 9: *Sử dụng dịch vụ*

Ngành LT & CSDL

www.t3h.vn



2014

2014



Nội dung

1. Giới thiệu
2. Sử dụng dịch vụ - Web Service



Dịch vụ - Web Service

☐ Giới thiệu: Web service (www.phpvn.org)

- Dịch vụ Web (Web Service) được coi là một công nghệ mang đến cuộc cách mạng trong cách thức hoạt động của các dịch vụ B2B (Business to Business) và B2C (Business to Customer).
- Giá trị cơ bản của dịch vụ Web dựa trên việc cung cấp các phương thức theo chuẩn trong việc truy nhập đối với hệ thống đóng gói và hệ thống kế thừa.
- Các phần mềm được viết bởi những ngôn ngữ lập trình khác nhau và chạy trên những nền tảng khác nhau có thể sử dụng dịch vụ Web để chuyển đổi dữ liệu thông qua mạng Internet theo cách giao tiếp tương tự bên trong một máy tính.





Dịch vụ - Web Service

❑ Giới thiệu: Web service (www.phpvn.org)

- Tuy nhiên, công nghệ xây dựng dịch vụ Web không nhất thiết phải là các công nghệ mới, nó có thể kết hợp với các công nghệ đã có như XML, SOAP, WSDL, UDDI...
- Với sự phát triển và lớn mạnh của Internet, dịch vụ Web thật sự là một công nghệ đáng được quan tâm để giảm chi phí và độ phức tạp trong tích hợp và phát triển hệ thống.



Dịch vụ - Web Service

❑ Giới thiệu: Web service (www.w3schools.com)

- Là các thành phần của ứng dụng có thể được xuất bản, tìm thấy và sử dụng trên trang web
- Giao tiếp bằng cách sử dụng các giao thức mở
- Khép kín và tự mô tả
- Có thể được tìm thấy bằng cách sử dụng UDDI
- Có thể được sử dụng bởi nhiều ứng dụng khác nhau
- HTTP và XML là nền tảng cho các dịch vụ Web



Dịch vụ - Web Service



□ Ví dụ

- Dịch vụ của Google: Google Weather, Google Map...
- Dịch vụ của Yahoo: YM!, Sports...
- Dịch vụ của các website khác:
<http://quotes.vinatrading.com/gia-vang.html>,
<http://dongabank.com.vn/exchange/export...>



LTV Mã nguồn mở PHP – Module 1

6

Giới thiệu



□ Hoạt động



LTV Mã nguồn mở PHP – Module 1

7

Nội dung

1. Giới thiệu
2. Sử dụng dịch vụ - Web Service



Sử dụng dịch vụ

- ☐ Để sử dụng một Web Service, Client cần phải biết Web Service đó hỗ trợ những phương thức nào, phương thức cần có những tham số nào, kết quả trả về ra sao...
- ☐ Những thông tin này của một Web Service được mô tả bởi tài liệu WSDL (Web Service Description Language).





Sử dụng dịch vụ

❑ Sử dụng Web Service miễn phí

- Ví dụ: Sử dụng dịch vụ YM! để làm phần tư vấn trực tuyến cho website như sau:



Sử dụng dịch vụ

❑ Sử dụng Web Service miễn phí

- Ví dụ: Sử dụng dịch vụ YM! để làm phần tư vấn trực tuyến cho website như sau:

▪ Đường dẫn

```
<a href="ymsgr:sendim?cong.bknp">

</a>
```





Sử dụng dịch vụ

❑ Sử dụng Web Service miễn phí

- Ví dụ: Sử dụng dịch vụ YM! để làm phần tư vấn trực tuyến cho website như sau:

■ Tham số

- Nick name
- Info (thông tin hiển thị)
- Icon (chọn icon hiển bằng cách nhập các số 1, 2, 3,...)



Sử dụng dịch vụ

❑ Sử dụng Web Service miễn phí

- Ví dụ: Sử dụng dịch vụ YM! để làm phần tư vấn trực tuyến cho website như sau:

■ Phương thức hỗ trợ

```
<a rel='nofollow' href='ymsgr:sendim<?php echo $nick ?>'>
<script type="text/javascript"
  src="http://download.skype.com/share/skypebuttons/js/skypeCh
  eck.js"></script>
<img alt='Administrator'
  src='http://opi.yahoo.com/online?u=<?php echo $nick
  ?>&m=g&t=<?php echo $ico ?>' border=0 align=middle
  width="80px" />
<span><?php echo $info ?></span>
</a>
```

