



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 2

---

*Click vào phụ lục để chuyển tới bài cần đọc*

### Phụ lục

Bài 1 Form và các điều khiển cơ sở .....	2
Bài 2 Javascript căn bản.....	13
Bài 3 Kết hợp PHP và JavaScript .....	29
Bài 4 Lập trình hướng đối tượng nâng cao.....	37
Bài 5 XML.....	50
Bài 6 XML - DOM .....	65
Bài 7 Tích hợp mã PHP .....	74
Bài 8 SESSION - COOKIE .....	85
Bài 9 Hoàn chỉnh website .....	95





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

## LTV MÃ NGUỒN MỞ PHP

### Module 2 – Bài 1: *Form và các điều khiển cơ sở*

Ngành LT & CSDL

[www.t3h.vn](http://www.t3h.vn)

2014



2014





## Nội dung

1. Các điều khiển cơ sở (tt)
2. Đọc giá trị từ điều khiển form



## Các điều khiển cơ sở

### ☐ List/Menu

#### ● List:

- Là một danh sách nhiều mục chọn cho phép người dùng chọn lựa. Có thể chọn một mục chọn hay nhiều mục chọn.
- Nếu muốn chọn được nhiều mục chọn trên List thì thuộc tính “Selections Allow Multiple” phải được chọn.





## Các điều khiển cơ sở

### ☐ List/Menu

- List:

- Cách tạo: vào Menu Insert => Form => List/Menu
- Lần lượt thêm các mục chọn, thiết lập nội dung và giá trị cho mỗi mục trong List.



## Các điều khiển cơ sở

### ☐ List/Menu

- Menu:

- Là một danh sách nhiều mục chọn cho phép người dùng chọn lựa một mục chọn.



## Các điều khiển cơ sở



### ☐ List/Menu

- Menu:

- Cách tạo: vào Menu Insert => Form => List/Menu
- Lần lượt thêm các mục chọn, thiết lập nội dung và giá trị cho mỗi mục trong Menu.



## Các điều khiển cơ sở



### ☐ List/Menu

- Menu:

- Cách tạo: Sử dụng thẻ select





## Các điều khiển cơ sở

### □ Field Field

- Cho phép người dùng chọn một file để xem xét các thuộc tính hoặc Upload file,...



## Các điều khiển cơ sở

### □ Field Field

- Cách tạo: vào Menu Insert => Form => chọn FileField
- Sử dụng thẻ input để tạo: `<input type="file" name="file" />`
  - Thiết lập các thuộc tính cơ bản như tên (name), và kiểu (type)





## Nội dung

1. Các điều khiển cơ sở
2. 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"];
```



## Đọc giá trị từ điều khiển form

### ☐\$\_GET

- Được dùng để lấy các giá trị trên Form thông qua phương thức GET.
- Thông tin được gửi từ Form với phương thức GET có giới hạn lượng thông tin gửi đi (tối đa 100 ký tự), và sẽ hiển thị trên địa chỉ URL nên người dùng có thể thấy được các thông tin này.
- Chú ý:** vì người dùng có thể thấy được thông tin gửi đi thông qua địa chỉ URL do đó không nên dùng phương thức GET nếu các thông tin gửi đi có tính chất bảo mật như mật khẩu, số tài khoản...







## Đọc giá trị từ điều khiển form

### ☐\$\_GET

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

```
$_GET["tên điều khiển"]
```

- Ví dụ: lấy giá trị TextField tên là txtTen

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



## Đọc giá trị từ điều khiển form

### ☐\$\_REQUEST

- Chứa nội dung của cả biến \$\_POST, \$\_GET, \$\_COOKIE
  - Có thể được sử dụng để lấy thông tin trên Form được gửi đi bằng cả phương thức POST và GET
- Mức độ bảo mật thông tin khi sử dụng biến \$\_REQUEST thấp → chỉ nên sử dụng biến \$\_REQUEST khi trang Web không có yêu cầu về bảo mật





## Đọc giá trị từ điều khiển form

### ☐\$\_REQUEST

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

```
$_REQUEST["tên điều khiển"]
```

- Ví dụ: lấy giá trị TextField tên là txtTen

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



## Đọc giá trị từ điều khiển form

### ☐\$\_FILES

- Là một biến mảng chứa các thông tin của file được chọn từ điều khiển FileField. Chỉ có thể lấy được giá trị của các thuộc tính của File thông qua biến này.





## Đọc giá trị từ điều khiển form

### ☐\$\_FILES

- Một số thuộc tính của file:
  - \$\_FILES["tên\_đk\_filefield"]["name"] – tên File
  - \$\_FILES["tên\_đk\_filefield"]["type"] – kiểu của File
  - \$\_FILES["tên\_đk\_filefield"]["size"] – kích cỡ của File
  - \$\_FILES["tên\_đk\_filefield"]["tmp\_name"] – tên tạm của File
  - \$\_FILES["tên\_đk\_filefield"]["error"] – lỗi của File



## Đọc giá trị từ điều khiển form

### ☐\$\_FILES

- Cách sử dụng
    - Cú pháp để lấy tên, kiểu, ... của một đối tượng trên Form sau khi Form submit:
- `$_FILES["tên_đk_filefield"]["name"]`**
- Ví dụ: lấy giá trị kích cỡ của điều khiển có tên là file\_1







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

# LTV MÃ NGUỒN MỞ PHP

## Module 2 – Bài 2: *JavaScript cơ bản*

Ngành LT & CSDL

[www.t3h.vn](http://www.t3h.vn)

2014



2014





## Nội dung

1. Tổng quan về JavaScript
2. Một số hộp thoại cơ bản – Popup Boxes
3. Sử dụng JavaScript trong trang HTML
4. Ngôn ngữ lập trình JavaScript
5. Sử dụng Javascript xử lý yêu cầu



## Tổng quan về JavaScript

- ☐ Của hãng Netscape
- ☐ Là một ngôn ngữ kịch bản (scripting language) dùng để tương tác với các trang HTML dựa trên đối tượng (object-based scripting language )
- ☐ Chủ yếu dùng cho kỹ thuật lập trình ở phía client
- ☐ Code của JavaScript thường được nhúng (embedded) trực tiếp hoặc tích hợp (integrated) vào trang web





## Tổng quan về JavaScript

### ❑ Đặc điểm

- Là một ngôn ngữ thông dịch (interpreted language), nghĩa là các script thi hành không cần biên dịch trước (precompile). Trình duyệt dịch script, phân tích và thi hành ngay tức thời
- Lập trình theo cấu trúc (Structured programming)
- Có phân biệt chữ HOA và thường



## Tổng quan về JavaScript

### ❑ Trình duyệt hỗ trợ:

- Netscape Navigator (bắt đầu từ phiên bản 2.0)
- Microsoft Internet Explorer (bắt đầu từ phiên bản 3.0)
- Những trình duyệt khác có hỗ trợ JavaScript (như Firefox, Chrome, Opera, ...)



## Nội dung

1. Tổng quan về JavaScript
2. Một số hộp thoại cơ bản – Popup Boxes
3. Sử dụng JavaScript trong trang HTML
4. Ngôn ngữ lập trình JavaScript
5. Sử dụng Javascript xử lý yêu cầu



## Một số hộp thoại cơ bản

### □ Hộp thông báo – Alert Box:

- Xuất câu thông báo trong một cửa sổ







## Một số hộp thoại cơ bản

### ☐ Hộp xác nhận – Confirm box

- Có dạng hàm (true/false), giúp người dùng quyết định đồng ý hoặc từ chối một yêu cầu



## Nội dung

1. Tổng quan về JavaScript
2. Một số hộp thoại cơ bản – Popup Boxes
3. Sử dụng JavaScript trong trang HTML
4. Ngôn ngữ lập trình JavaScript
5. Sử dụng Javascript xử lý yêu cầu



## Sử dụng JavaScript trong trang HTML



- ☐ Chèn JavaScript vào Head section
- ☐ Chèn JavaScript vào Body section
- ☐ Sử dụng tập tin thư viện – External script
- ☐ Chèn JavaScript vào sự kiện



## Sử dụng JavaScript trong trang HTML



- ☐ Chèn JavaScript vào Head section
  - Đặt đoạn lệnh JavaScript trong cặp tag **<script>**





## Sử dụng JavaScript trong trang HTML

### ❑ Chèn JavaScript vào Body section

- Đặt cặp tag <script> trong cặp tag <body>



## Sử dụng JavaScript trong trang HTML

### ❑ Sử dụng tập tin thư viện – External script

- Các đoạn code dùng chung sẽ được viết trong một tập tin và lưu với phần mở rộng là .js
- Sử dụng thuộc tính **src** (source) của tag <script> để khai báo tập tin \*.js





## Sử dụng JavaScript trong trang HTML

### ❑ Chèn JavaScript vào sự kiện

- HTML cho phép chèn JavaScript vào các sự kiện của tag
- Code JavaScript sẽ được thi hành khi phát sinh sự kiện



## Nội dung

1. Tổng quan về JavaScript
2. Một số hộp thoại cơ bản – Popup Boxes
3. Sử dụng JavaScript trong trang HTML
4. Ngôn ngữ lập trình JavaScript
5. Sử dụng Javascript xử lý yêu cầu





## Ngôn ngữ lập trình JavaScript

- ❑ Chú thích
- ❑ Biến – Variable
- ❑ Hàm - Function
- ❑ Một số hàm cơ bản trong JavaScript



## Ngôn ngữ lập trình JavaScript

- ❑ Chú thích
  - Chú thích cho một dòng, ta dùng ký hiệu //
  - Chú thích từ 2 dòng trở lên, bắt đầu bằng ký hiệu /\* và kết thúc bằng \*/





## Ngôn ngữ lập trình JavaScript

### ❑ Biến – Variable

#### ● Khai báo:

```
var <tên biến>;
```

```
hoặc var <tên biến> = <giá trị>;
```

```
hoặc <tên biến> = <giá trị>;
```

- <tên biến> phải bắt đầu bằng ký tự (a -> z) hoặc dấu gạch dưới (\_)
- <tên biến> có phân biệt chữ HOA/thường
- Khai báo bên ngoài hàm gọi là biến toàn cục (global variable)
- Khai báo trong hàm gọi là biến cục bộ (local variable) và chỉ được phép sử dụng trong hàm khai báo nó



## Ngôn ngữ lập trình JavaScript

### ❑ Biến – Variable

#### ● Giá trị của biến:

- Số (Number): một con số (như 23 hoặc 3.1415)
- Luận lý (Boolean) : true hoặc false
- Chuỗi (String) : một chuỗi các ký tự ( như “abc”)
- null : khi không xác định giá trị
- undefined : khi không xác định giá trị





## Ngôn ngữ lập trình JavaScript

### ❑ Biến – Variable

#### ● Phân biệt giá trị null và undefined:

- undefined: biến được khai báo nhưng chưa được gán giá trị

```
var TestVar;  
alert(TestVar); ➔ undefined
```

- null: biến được khai báo và được gán = null

```
var TestVar = null;  
alert(TestVar); ➔ null
```



## Ngôn ngữ lập trình JavaScript

### ❑ Biến – Variable

#### ● Phân biệt giá trị null và undefined:

- undefined: biến được khai báo nhưng chưa được gán giá trị

```
var TestVar;  
alert(TestVar); ➔ undefined
```

- null: biến được khai báo và được gán = null

```
var TestVar = null;  
alert(TestVar); ➔ null
```





## Ngôn ngữ lập trình JavaScript

### ❑ Hàm - Function

- **Khai báo:** Hàm có thể được khai báo trong tag <head> hoặc tag <body> hoặc trong tập tin \*.js

```
function <tên hàm>( [ tham số 1, tham số 2, ... ] )
{
    /* khối lệnh xử lý */
    [ return <giá trị trả về > ]
}
```

- Ví dụ

```
function dt_hinhvuong(canh)
{ return canh * canh; }
```



## Ngôn ngữ lập trình JavaScript



### ❑ Một số hàm cơ bản trong JavaScript

Vấn đề:

Cần kiểm tra, đánh giá, chuyển đổi dữ liệu hiển thị...

Ví dụ:

- Đánh giá kết quả của chuỗi str
- Kiểm tra xem biến giaTri có phải là số hay không
- Phân tích chuỗi thành số nguyên/ số thực

Giải quyết:

Sử dụng hàm thời gian trong thư viện hàm của Java

```
var str="if (3>2)
document.write('đúng'); else
document.write('sai');"

var giaTri = 5;
// Đánh giá kết quả của chuỗi str
var ktra = eval(str);
// Kiểm tra biến giá trị
var result = isNaN(giaTri);
// Phân tích chuỗi thành số nguyên/ số thực
var so = parseInt("10.33"); // 10
var so = parseFloat("10.33"); // 10.33
```







## Nội dung

1. Tổng quan về JavaScript
2. Một số hộp thoại cơ bản – Popup Boxes
3. Sử dụng JavaScript trong trang HTML
4. Ngôn ngữ lập trình JavaScript
5. Sử dụng JavaScript xử lý yêu cầu



## Sử dụng JavaScript xử lý yêu cầu



### □ Các sự kiện của thẻ

- Các sự kiện của thẻ xảy ra khi người dùng tác động đến thẻ.
  - Ví dụ click vào thẻ thì xảy ra biến cố onclick
- Ta thường sử dụng hàm để xử lý biến cố cho thẻ
  - Ví dụ: Hàm javascript lưu\_click để xử lý khi click vào nút Lưu trữ

```
<input type = "submit" value = "Lưu trữ" onclick =  
"luu_click()" />
```





## Sử dụng JavaScript xử lý yêu cầu

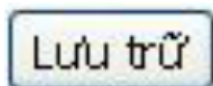
### □ Các sự kiện của thẻ



## Sử dụng JavaScript xử lý yêu cầu

### □ Các sự kiện của thẻ

- Ví dụ: xử lý kiểm tra dữ liệu nhập khi click vào nút lưu.





## Thao tác với đối tượng thẻ HTML

### ❑ Truy xuất đối tượng thẻ:

```
the = document.getElementById('the_id');
```

- Ví dụ: Truy xuất ô nhập liệu Họ tên có id="ho\_ten"

```
var ho_ten =  
    document.getElementById('ho_ten');
```



## Thao tác với đối tượng thẻ HTML

### ❑ Lấy giá trị của thẻ

- Thẻ nhập liệu (input, select, textarea)

```
gia_tri = the.value;
```

- Thẻ có khả năng chứa thẻ khác (div, span, h1->h6, p, td, ...)

```
gia_tri = the.innerHTML;
```





## Thao tác với đối tượng thẻ HTML

### ❑ Lấy giá trị của thẻ

- Ví dụ: Tính chu vi và diện tích hình tròn khi biết bán kính.

Tính chu vi - diện tích hình tròn	
Bán kính	<input type="text" value="6"/>
Chu vi	37.68
Diện tích	113.03999999999999
<input type="button" value="Tính"/>	





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

## LTV MÃ NGUỒN MỞ PHP

### Module 2 – Bài 3: *Kết hợp PHP & JavaScript*

Ngành LT & CSDL

[www.t3h.vn](http://www.t3h.vn)

2014



2014





## Nội dung

1. Kiểm tra dữ liệu nhập
2. Di chuyển qua trang khác
3. Xác nhận xóa
4. Sử dụng thư viện JavaScript có sẵn



## Kiểm tra dữ liệu nhập

### □ Giới thiệu:

- Để kiểm tra dữ liệu nhập của form, ta thường dùng cả JavaScript và PHP (phòng trường hợp trình duyệt không hỗ trợ hay tắt JavaScript) để kiểm tra.





## Kiểm tra dữ liệu nhập

### ❑ Xử lý kiểm tra:

- Thường dùng JavaScript kiểm tra ở biến cố onclick của button submit.
- Dùng PHP kiểm tra sau khi lấy dữ liệu nhập



## Kiểm tra dữ liệu nhập



### ❑ Ví dụ: Xử lý kiểm tra dữ liệu nhập của form thêm mới khách hàng

Thêm mới khách hàng

Mã khách hàng	<input type="text"/>
Họ tên	<input type="text"/>
Phái	<input checked="" type="radio"/> Nam <input type="radio"/> Nữ
Địa chỉ	<input type="text"/>
Điện thoại	<input type="text"/>
Email	<input type="text"/>
<input type="button" value="Lưu trữ"/> <input type="button" value="Nhập lại"/> <input type="button" value="Quay về"/>	



## Kiểm tra dữ liệu nhập

Lưu trữ



LTV Mã nguồn mở PHP – Module 2

6

## Kiểm tra dữ liệu nhập

### Code PHP kiểm tra dữ liệu nhập

```
//Nếu người dùng click vào nút Lưu trữ
if(isset($_POST['Ten_khach_hang'])) {
    //lấy dữ liệu nhập
    $Ma_khach_hang = $_POST['Ma_khach_hang'];
    $Ten_khach_hang = $_POST['Ten_khach_hang'];
    $Phai = $_POST['Phai'];
    $Dia_chi = $_POST['Dia_chi'];
    $Dien_thoai = $_POST['Dien_thoai'];
    $Email = $_POST['Email'];
    //kiểm tra dữ liệu nhập
    $hop_le = true;
    if(trim($Ma_khach_hang)=='') {
        $thong_bao = "'Mã khách hàng' không được phép rỗng!";
        $hop_le = false;
    }
    elseif(trim($Ten_khach_hang)=='') {
        $thong_bao = "'Tên khách hàng' không được phép rỗng!";
        $hop_le = false;
    }
    elseif(!preg_match('/^[^@\s]+@([-a-z0-9]+\.)+[a-z]{2,}$/i', $Email)) {
        $thong_bao = "'Email' không hợp lệ!";
        $hop_le = false;
    }
}

if($hop_le) {
    //xử lý thêm
    //viết truy vấn thêm
    $lenhsql = "INSERT INTO KHACH_HANG(Ma_khach_hang, Ten_khach_hang, Phai, Dia_chi,
    Dien_thoai, Email)
    VALUES('$Ma_khach_hang', '$Ten_khach_hang', '$Phai', '$Dia_chi', '$Dien_thoai',
    '$Email')";
    //thực hiện lệnh
    $kq = Thuc_hien_lenh($lenhsql);
}
```



LTV Mã nguồn mở PHP – Module 2

7





## Nội dung

1. Kiểm tra dữ liệu nhập
2. Di chuyển qua trang khác
3. Xác nhận xóa
4. Sử dụng thư viện JavaScript có sẵn



## Di chuyển qua trang khác

### □Cú pháp:

```
window.location = url;
```

- Ví dụ: Khi click vào button thêm thì di chuyển sang trang thêm khách hàng





## Nội dung

1. Kiểm tra dữ liệu nhập
2. Di chuyển qua trang khác
3. Xác nhận xóa
4. Sử dụng thư viện JavaScript có sẵn



LTV Mã nguồn mở PHP – Module 2

10

## Xác nhận xóa



- ☐ Chức năng xóa thường có ở lưới liệt kê dữ liệu

Danh sách khách hàng

Mã khách hàng	Tên khách hàng	Giới tính	Địa chỉ	Số điện thoại	
kh001	Khuất Thủy Phương	Nữ	A21 Nguyễn Oanh quận Gò Vấp	9874125	<a href="#">Xóa</a> <a href="#">Sửa</a>
kh002	Đỗ Lâm Thiên	Nam	357 Lê Hồng Phong Q. 10	8351056	<a href="#">Xóa</a> <a href="#">Sửa</a>
kh003	Phạm Thị Nhung	Nữ	56 Đinh Tiên Hoàng quận 1	9745698	<a href="#">Xóa</a> <a href="#">Sửa</a>
kh004	Nguyễn Khắc Thiện	Nam	12bis Đường 3-2 quận 10	8769128	<a href="#">Xóa</a> <a href="#">Sửa</a>
kh005	Tô Trần Hồ Giảng	Nam	75 Nguyễn Kiệm quận Gò Vấp	5792564	<a href="#">Xóa</a> <a href="#">Sửa</a>
kh006	Nguyễn Kiến Thi	Nữ	357 Lê Hồng Phong Q. 10	9874125	<a href="#">Xóa</a> <a href="#">Sửa</a>
kh007	Trần Quốc Thông	Nam	123 Trần Hưng Đạo	8754123	<a href="#">Xóa</a> <a href="#">Sửa</a>
kh008	Nguyễn Anh Tuấn	Nam	1/2bis Nơ Trang Long Q. BT TP.HCM	8753159	<a href="#">Xóa</a> <a href="#">Sửa</a>
<input type="button" value="Thêm mới"/>					



LTV Mã nguồn mở PHP – Module 2

11



## Nội dung

1. Kiểm tra dữ liệu nhập
2. Di chuyển qua trang khác
3. Xác nhận xóa
4. Sử dụng thư viện JavaScript có sẵn



## Sử dụng thư viện JavaScript có sẵn



### ☐ Tìm kiếm thư viện JavaScript

- Có rất nhiều website cung cấp mã nguồn JavaScript, ta chỉ việc download về sử dụng (có hướng dẫn kèm theo).
- Địa chỉ một số website:
  - <http://jquery.com>
  - <http://jqueryui.com/>
  - <http://javascriptlibraries.com/>
  - [http://www.w3schools.com/js/js\\_libraries.asp](http://www.w3schools.com/js/js_libraries.asp)





## Sử dụng thư viện JavaScript có sẵn

### □ Sử dụng thư viện JavaScript

- Các bước thực hiện:
  - Chép thư viện JavaScript vào ứng dụng web
  - Trong trang web, link tới các file css, JavaScript cần dùng.
  - Sử dụng thư viện JavaScript theo hướng dẫn hoặc ví dụ đi kèm với thư viện JavaScript





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

# LTV MÃ NGUỒN MỞ PHP

## Module 2 – Bài 4: *LTHĐT nâng cao*

Ngành LT & CSDL


[www.t3h.vn](http://www.t3h.vn)

2014



2014






---




LTV Mã nguồn mở PHP – Module 2

2




---




LTV Mã nguồn mở PHP – Module 2

3




---




LTV Mã nguồn mở PHP – Module 2

4



---




LTV Mã nguồn mở PHP – Module 2

5




---



LTV Mã nguồn mở PHP – Module 2

6




---




LTV Mã nguồn mở PHP – Module 2

7






---




LTV Mã nguồn mở PHP – Module 2

8



---



LTV Mã nguồn mở PHP – Module 2

9




---




LTV Mã nguồn mở PHP – Module 2

10




---

<b>nv_vp</b> số ngày vắng định mức vắng đơn giá phạt	<b>nv_sx</b> số sản phẩm định mức sp đơn giá sp
---	--



LTV Mã nguồn mở PHP – Module 2

11



---



LTV Mã nguồn mở PHP – Module 2

12




---




LTV Mã nguồn mở PHP – Module 2

13



---



LTV Mã nguồn mở PHP – Module 2

14



---




LTV Mã nguồn mở PHP – Module 2

15




---



LTV Mã nguồn mở PHP – Module 2

16




---




LTV Mã nguồn mở PHP – Module 2

17




---



LTV Mã nguồn mở PHP – Module 2

18



---




LTV Mã nguồn mở PHP – Module 2

19



---




LTV Mã nguồn mở PHP – Module 2

20




---




LTV Mã nguồn mở PHP – Module 2

21



---




LTV Mã nguồn mở PHP – Module 2

22



---



LTV Mã nguồn mở PHP – Module 2

23





QUẢN LÝ NHÂN VIÊN			
Họ và tên:	<input type="text" value="Nguyễn Thiên Trang"/>	Số con:	<input type="text" value="2"/>
Ngày sinh:	<input type="text" value="10/10/1978"/>	Ngày vào làm:	<input type="text" value="20/11/2005"/>
Giới tính:	<input type="radio"/> Nam <input checked="" type="radio"/> Nữ	Hệ số lương:	<input type="text" value="2.0"/>
Loại nhân viên:	<input checked="" type="radio"/> Văn phòng <input type="radio"/> Sản xuất		
	Số ngày vắng : <input type="text" value="3"/>	Số sản phẩm:	<input type="text"/>
		Tăng ca:	<input type="radio"/> Có <input checked="" type="radio"/> Không
Tiền lương:	<input type="text" value="875.000 VNĐ"/>	Trợ cấp:	<input type="text" value="240.000 VNĐ"/>
Thực lĩnh:		<input type="text" value="1.115.000 VNĐ"/>	
<input type="button" value="Tính lương"/>			





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

## LTV MÃ NGUỒN MỞ PHP

### Module 2 – Bài 5: **XML**

Ngành LT & CSDL

[www.t3h.vn](http://www.t3h.vn)

2014



2014





## Giới thiệu





## Giới thiệu

### ❑ XML (eXtensible Markup Language) – Ngôn ngữ đánh dấu mở rộng

- Được thiết kế để chuyển và lưu trữ dữ liệu
- Do W3C đề nghị sử dụng
- Mục đích chính của XML là đơn giản hoá việc chia sẻ dữ liệu giữa các hệ thống khác nhau, đặc biệt là các hệ thống được kết nối với Internet



## Giới thiệu

### ❑ Đặc điểm

- Dễ dàng viết được những chương trình xử lý tài liệu XML
- Dễ đọc và có tính hợp lý cao
- Giảm thiểu những thuộc tính tùy chọn
- Dễ dàng được sử dụng trên Internet
- Hỗ trợ nhiều ứng dụng





## Cấu trúc và cú pháp



### □ Cấu trúc

- Tài liệu XML có cấu trúc cây (tree) bắt đầu bằng một nút gốc (root) và các nhánh lá (leave)
- Tài liệu XML là tài liệu tự mô tả và có cấu trúc đơn giản





## Cấu trúc và cú pháp

### □ Cấu trúc

```
<?xml version="1.0" encoding="UTF-8"?>
<note>
  <to>Students</to>
  <from>Teacher</from>
  <heading>Reminder</heading>
  <body>Please come to class this Monday evening
on time</body>
</note>
```

Khai báo phiên bản và loại mã hóa

Nút gốc (root element)

Nút con (child element)

Kết thúc nút gốc



## Cấu trúc và cú pháp



### □ Cú pháp

- Có một root Element duy nhất
- Mỗi tag được mở ra cần phải đóng lại
- Tag có phân biệt chữ hoa và chữ thường
- Nút con (Child Element) phải nằm trong nút cha (Parent Element)





## Cấu trúc và cú pháp

### □ Cú pháp

- Ghi chú trong XML: `<!-- Nội dung ghi chú-->`
- Lưu trữ dấu xuống dòng: LF
- Chú ý: Trong file .xml cần phải có dòng khai báo XML ở đầu file



## Cấu trúc và cú pháp



### □ Ví dụ

```
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<Don_hang>
  <Ngay_dat_hang>2010-01-27</Ngay_dat_hang>
  <Ten_khach_hang>Khuất Thủy Phương</Ten_khach_hang>
  <Noi_dung>
    <Ma_san_pham>1</Ma_san_pham>
    <So_luong>2</So_luong>
  </Noi_dung>
  <Noi_dung>
    <Ma_san_pham>4</Ma_san_pham>
    <So_luong>1</So_luong>
  </Noi_dung>
</Don_hang>
```





## Phần tử (element)



- ☐ **Phần tử XML (XML element) là các nội dung được bắt đầu bằng tag mở và kết thúc bằng tag đóng**
- ☐ **Một phần tử có thể chứa:**
  - Các phần tử khác
  - Văn bản
  - Thuộc tính
  - Và dữ liệu tổng hợp
- ☐ **Cũng có những phần tử XML không có nội dung được viết như sau: `<book></book>` hoặc `<book />`**







## Phần tử (element)

### ❑ Quy tắc đặt tên phần tử XML

- Tên có thể chứa chữ, số và các ký tự khác
- Tên không bắt đầu bằng số hoặc dấu chấm câu (.)
- Tên không bắt đầu là xml (XML, hoặc Xml, etc)
- Tên không chứa khoảng trắng
- Tất cả tên đều có thể sử dụng, không có từ dành riêng



## Phần tử (element)



### ❑ Ví dụ:

```
<bookstore>
  <book category="CHILDREN">
    <title>Harry Potter</title>
    <author>J K. Rowling</author>
    <year>2005</year>
    <price>29.99</price>
  </book>
</bookstore>
```

- <bookstore> và <book> là các phần tử có nội dung, nó chứa các phần tử khác.
- <book> có một thuộc tính (attribute) là (category="CHILDREN").
- <title>, <author>, <year>, and <price> chứa nội dung văn bản





## Phần tử (element)

- ❑ Các phần tử XML có thể mở rộng thêm thông tin mà không làm cho ứng dụng bị ảnh hưởng

● Ví dụ:

Mở rộng

```
<note>
<to>Students</to>
<from>Teacher</from>
<body>Come to class on
time</body>
</note>
```

```
<note>
<date>2014-03-01</date>
<to>Students</to>
<from>Teacher</from>
<heading>Reminder</heading>
<body>Come to class on
time</body>
</note>
```





## Thuộc tính (attribute)

- ❑ Thuộc tính cung cấp thêm các thông tin về một phần tử
  - Ví dụ: `<file type="gif">computer.gif</file>`
- ❑ Thông tin được cung cấp thêm này thường không phải là một phần của dữ liệu
  - Ví dụ: `type="gif"` không liên quan đến dữ liệu, nhưng có thể quan trọng với phần mềm khi muốn thao tác với các phần tử
- ❑ Giá trị của thuộc tính phải đặt trong ngoặc đơn “” hoặc ngoặc kép “”
  - Ví dụ: `<person sex="female">` hoặc `<person sex='female'>`



LTV Mã nguồn mở PHP – Module 2

18

## Thuộc tính (attribute)



### ❑ So sánh

- Không có quy định khi nào thì sử dụng thuộc tính khi nào thì sử dụng phần tử. Tuy nhiên nên sử dụng phần tử thay cho thuộc tính
- Ví dụ: cả thuộc tính và phần tử `sex` dưới đây cùng có chung thông tin là `female`

<pre>&lt;person sex="female"&gt;   &lt;firstname&gt;Anna&lt;/firstn ame&gt;   &lt;lastname&gt;Smith&lt;/lastna me&gt; &lt;/person&gt;</pre>	<pre>&lt;person&gt;   &lt;sex&gt;female&lt;/sex&gt;   &lt;firstname&gt;Anna&lt;/firstn ame&gt;   &lt;lastname&gt;Smith&lt;/lastna me&gt; &lt;/person&gt;</pre>
---	--



LTV Mã nguồn mở PHP – Module 2

19



## Thuộc tính (attribute)

### ❑ So sánh

- Tuy nhiên nên sử dụng phần tử thay cho thuộc tính, vì:
  - Thuộc tính không thể chứa nhiều giá trị (multiple values)
  - Thuộc tính không thể chứa cấu trúc cây (tree structure)
  - Thuộc tính không dễ mở rộng (khi có yêu cầu thay đổi trong tương lai)





## Không gian tên (Namespace)

- ❑ XML Namespace cung cấp một phương pháp để tránh xung đột tên các phần tử: sử dụng tiền tố

```
<table>
  <tr>
    <td>Apples</td>
    <td>Bananas</td>
  </tr>
</table>
```

**Xung đột**

```
<table>
  <name>African Coffee
  Table</name>
  <width>80</width>
  <length>120</length>
</table>
```



```
<h:table>
  <h:tr>
    <h:td>Apples</h:td>
    <h:td>Bananas</h:td>
  </h:tr>
</h:table>
```

```
<f:table>
  <f:name>African Coffee
  Table</f:name>
  <f:width>80</f:width>
  <f:length>120</f:length>
</f:table>
```



## Không gian tên (Namespace)



### ❑ Thuộc tính xmlns

- Khi sử dụng tiền tố trong xml, một không gian tên cho tiền tố phải được xác định.
- Không gian tên được xác định bởi thuộc tính xmlns trong thẻ bắt đầu của một phần tử
- Cú pháp khai báo không gian tên:

**xmlns:prefix="URI"**





## Không gian tên (Namespace)

### ❑ Ví dụ:

```
<root>

  <h:table xmlns:h="http://www.w3.org/TR/html4/">
    <h:tr>
      <h:td>Apples</h:td>
      <h:td>Bananas</h:td>
    </h:tr>
  </h:table>

  <f:table xmlns:f="http://www.w3schools.com/furniture">
    <f:name>African Coffee Table</f:name>
    <f:width>80</f:width>
    <f:length>120</f:length>
  </f:table>

</root>
```



## Không gian tên (Namespace)

### ❑ Không gian tên mặc định (default namespace)

- Xác định một không gian tên mặc định cho một phần tử giúp chúng ta không phải sử dụng tiền tố trong các phần tử con

- Cú pháp: **xmlns="namespaceURI"**

- Ví dụ:
 

```
<table xmlns="http://www.w3.org/TR/html4/">
  <tr>
    <td>Apples</td>
    <td>Bananas</td>
  </tr>
</table>
```





## Mã hóa (encoding)



- ☐ Tài liệu XML có thể chứa các ký tự quốc tế như æøå, hay èèé. Để tránh sai sót, chúng ta cần phải xác định mã hóa được sử dụng hoặc lưu tập tin XML với kiểu mã hóa UTF-8
- ☐ UTF-8 là bảng mã ký tự chuẩn trên web
- ☐ UTF-8 là bảng mã ký tự mặc định cho HTML5, CSS, JavaScript, PHP, SQL, và XML.
- ☐ Mã hóa trong XML: `<?xml version="1.0" encoding="UTF-8"?>`





## Mã hóa (encoding)

### ❑ Chú ý:

- Nên sử dụng một trình soạn thảo XML hỗ trợ mã hóa
- Biết được cái gì sẽ mã hóa khi sử dụng trình soạn thảo
- Mô tả mã hóa trong thuộc tính mã hóa
- UTF-8 là bảng mã hóa an toàn nhất để sử dụng
- UTF-8 là tiêu chuẩn Web







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

## LTV MÃ NGUỒN MỞ PHP

Module 2 – Bài 6: ***XML - DOM***

Ngành LT & CSDL

[www.t3h.vn](http://www.t3h.vn)

2014



2014



## Nội dung

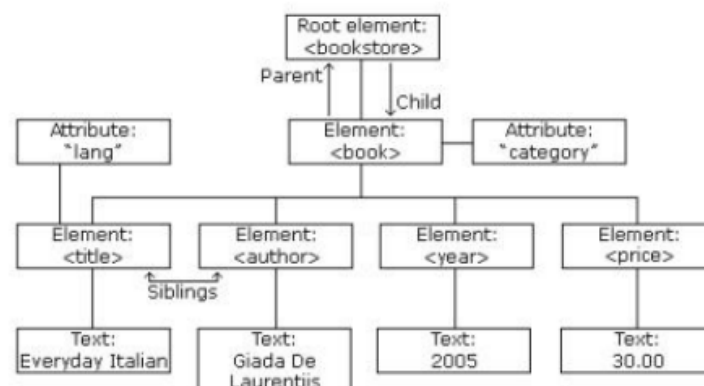
1. Mô hình DOM
2. Truy xuất tài liệu XML với DOM



## Mô hình DOM

### □ Khái niệm

- Mô hình đối tượng cho phép xử lý trên tài liệu XML từ các ngôn ngữ lập trình
- Cấu trúc dữ liệu động biểu diễn thông tin của văn bản có cấu trúc nói chung và tài liệu XML nói riêng





## Mô hình DOM

### ❑ Đặc điểm

- XML DOM xác định các đối tượng và thuộc tính của tất cả các phần tử XML cũng như các phương thức để truy cập chúng (cách lấy, thay đổi, thêm hoặc xóa các phần tử XML...)
- Đọc ghi dễ dàng
- Khả năng truy vấn cao với ngôn ngữ Xpath
- Theo DOM, mọi thứ trong tài liệu XML đều là node
  - Toàn bộ tài liệu là một node tài liệu
  - Mỗi phần tử XML là một node phần tử
  - Văn bản trong phần tử XML là node văn bản
  - Mỗi thuộc tính là một node thuộc tính
  - Ghi chú là node ghi chú



## Nội dung

1. Mô hình DOM
2. Truy xuất tài liệu XML với DOM





## Mở tài liệu XML

### □ Các bước thực hiện:

- Tạo đối tượng DOMDocument

```
$tai_lieu = new DOMDocument();
```

- Gọi phương thức load để mở tài liệu XML

```
$tai_lieu->load('file.xml');
```



## Đọc tài liệu XML



### □ Các bước thực hiện:

- Mở tài liệu XML

- Truy xuất node gốc

```
$node_goc = $tai_lieu->documentElement;
```

- Truy xuất các node con:

- Dùng thuộc tính childNodes để truy xuất các node con
- Mỗi node là một đối tượng thuộc class XMLElement hoặc là text





## Đọc tài liệu XML

### □XMLElement

- Đọc nội dung node
  - Thuộc tính nodeValue trả về nội dung của node
  - Dùng thuộc tính nodeType để kiểm tra 1 node là text hay Element



## Đọc tài liệu XML

### □XMLElement

```
<?xml version="1.0"
encoding="utf-8"?>
<PHAN_SO>
    <TU_SO>2</TU_SO>
    <MAU_SO>3</MAU_SO>
</PHAN_SO>
```

```
//tạo đối tượng
$tai_lieu = new DOMDocument();
//mở tài liệu XML
$tai_lieu->load("PHAN_SO.xml");
//truy xuất node gốc
$node_goc = $tai_lieu->documentElement;
$mang = $node_goc->childNodes;

foreach($mang as $node)
{
    if($node->nodeType==XML_ELEMENT_NODE)
        echo $node->nodeValue;
}
```





## Đọc tài liệu XML

### ❑XMLElement

- Đọc thuộc tính của 1 node: sử dụng phương thức **getAttribute** của node.
- Ví dụ: đọc thuộc tính `tu_so` và `mau_so` của node `PHAN_SO`



## Đọc tài liệu XML



```
<?xml version="1.0" encoding="utf-8"?>
<PHAN_SO tu_so="2" mau_so="3" />
```

```
//tạo đối tượng
$tai_lieu = new DOMDocument();
//mở tài liệu XML
$tai_lieu->load("PHAN_SO.xml");
//truy xuất node gốc
$node_goc = $tai_lieu->documentElement;
$tu_so = $node_goc->getAttribute('tu_so');
$mau_so = $node_goc->getAttribute('mau_so');
```





## Ghi tài liệu xml

### ❑ Tạo mới 1 node

```
$node = $tai_lieu->createElement($name)
```

Hoặc

```
$node = $tai_lieu->createTextNode($content)
```

### ❑ Tạo nội dung cho node

```
$node->nodeValue = $noi_dung;
```

### ❑ Thêm node mới vào node cha

```
$node_cha->appendChild($node_moi);
```



LTV Mã nguồn mở PHP – Module 2

12



## Ghi tài liệu xml

### ❑ Tạo thuộc tính cho node

```
$node->setAttribute('ten_thuoc_tinh', 'giá trị')
```

### ❑ Lưu tài liệu xml

```
$tai_lieu->save($duong_dan);
```



LTV Mã nguồn mở PHP – Module 2

13



## Ghi tài liệu xml

### ❑ Ví dụ: tạo tài liệu XML biểu diễn phân số.

```
<?xml version="1.0"
encoding="utf-8"?>
<PHAN_SO>
    <TU_SO>2</TU_SO>
    <MAU_SO>3</MAU_SO>
</PHAN_SO>
```

```
//tạo đối tượng Dom
$tai_lieu = new DOMDocument('1.0', 'utf-8');
//tạo node PHAN_SO
$node_phan_so = $tai_lieu->createElement('PHAN_SO');
//gắn node PHAN_SO vào tài liệu xml
$tai_lieu->appendChild($node_phan_so);
//tạo node TU_SO và MAU_SO
$node_tu_so = $tai_lieu->createElement('TU_SO');
$node_mau_so = $tai_lieu->createElement('MAU_SO');
//Gán giá trị
$node_tu_so->nodeValue = 2;
$node_mau_so->nodeValue = 3;
//gắn node TU_SO và node MAU_SO vào node PHAN_SO
$node_phan_so->appendChild($node_tu_so);
$node_phan_so->appendChild($node_mau_so);
//lưu tài liệu
$tai_lieu->save('PHAN_SO.XML');
```



LTV Mã nguồn mở PHP – Module 2

14

## Ghi tài liệu xml

### ❑ Ví dụ: tạo tài liệu XML biểu diễn phân số.

```
<?xml version="1.0"
encoding="utf-8"?>
<PHAN_SO tu_so="2"
    mau_so="3" />
```

```
//tạo đối tượng Dom
$tai_lieu = new DOMDocument('1.0', 'utf-8');
//tạo node PHAN_SO
$node_ps =
    $tai_lieu->createElement('PHAN_SO');
//gắn node PHAN_SO vào tài liệu xml
$tai_lieu->appendChild($node_ps);
//tạo các thuộc tính tu_so và mau_so
$node_ps->setAttribute('tu_so', 2);
$node_ps->setAttribute('mau_so', 3);
//lưu tài liệu
$tai_lieu->save('PHAN_SO.XML');
```



LTV Mã nguồn mở PHP – Module 2

15







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

## LTV MÃ NGUỒN MỞ PHP

### Module 2 – Bài 7: *Tích hợp mã PHP*

Ngành LT & CSDL

[www.t3h.vn](http://www.t3h.vn)

2014



2014





## Giới thiệu



- ❑ **Khi xây dựng các ứng dụng, việc kế thừa và sử dụng lại các đoạn chương trình đã viết là một công việc rất cần thiết vì:**
  - Giảm được thời gian và chi phí
  - Giúp tăng độ tin cậy của ứng dụng vì các function và module đã được xây dựng, kiểm tra và sử dụng tốt cho các ứng dụng trước cũng có thể được sử dụng lại cho các ứng dụng khác với mục đích tương tự.
  - Tạo sự nhất quán trong thiết kế và xây dựng ứng dụng





## Giới thiệu

---

- ❑ PHP cung cấp các phương thức rất đơn giản và hữu dụng cho phép sử dụng lại các đoạn chương trình đã viết là `require()` và `include()`.
- ❑ Hai phương thức này dùng để chèn file vào trang PHP.
- ❑ Các file được chèn có thể chứa các đoạn chương trình PHP, thẻ HTML, văn bản, hàm và class.





## require

- ❑ Cách sử dụng
- ❑ Các loại tập tin được dùng trong require()
- ❑ Thẻ PHP và require()
- ❑ Dùng require() cho các template



LTV Mã nguồn mở PHP – Module 2

6

## require



### ❑ Cách sử dụng

#### ● Đặc điểm

- Tất cả nội dung bên trong file được chèn vào trang sẽ được biên dịch
- Muốn sử dụng đoạn chương trình đã được viết sẵn ở vị trí nào trong trang thì dùng require() để chèn file chứa đoạn chương trình này ở vị trí đó

#### ● Cú pháp:

```
require ("tên_và_đường_dẫn_của_file");
```



LTV Mã nguồn mở PHP – Module 2

7



## require

### □ Cách sử dụng

- Ví dụ



## require



### □ Các loại tập tin được dùng trong require()

- Các câu lệnh PHP được viết trong các file .html sẽ không thực hiện được. → Các câu lệnh PHP trong các file được dùng để chèn phải có kiểu file thích hợp như .inc hay .php để các câu lệnh này có thể biên dịch được khi thực thi.





## require

### ❑ Các loại tập tin được dùng trong require()

- Chú ý:
  - PHP không quy định cách đặt tên file đối với file được chèn vào bằng require() → có thể đặt tên file tùy ý.
  - Dùng require() để chèn file → nội dung của file sẽ trở thành một phần trong trang PHP.



## require



### ❑ Thẻ PHP và require()

- Các lệnh PHP cần phải được bao bởi cặp thẻ PHP `<?php` và `?>`.
- Nếu không có thẻ PHP `<?php` và `?>` khi viết thì các lệnh PHP này sẽ trở thành dạng văn bản hoặc HTML và không thể thực thi được.





## require

### ❑ Dùng require() cho các template

- Nếu ứng dụng Web có cùng một mẫu thiết kế và chỉ khác về nhau nội dung bên trong → tạo ra một template và khai báo các biến trình bày dữ liệu bên trong template.
- Khi template này được thực thi, tất cả các biến này sẽ có giá trị và trình bày như một định dạng template được sử dụng nhiều lần.

→ Dùng require() để chèn template vào các trang cần thiết



LTV Mã nguồn mở PHP – Module 2

12

## require



### ❑ Ví dụ

- Trong ứng dụng web phần header và footer thường được hiển thị ở hầu hết các trang → tạo ra trang header.inc để chứa định dạng và nội dung phần đầu trang và trang footer.inc để chứa định dạng và nội dung phần cuối trang.
- Trang nào của ứng dụng có sử dụng header và footer thì dùng require() để chèn hai trang này vào.



LTV Mã nguồn mở PHP – Module 2

13





## Dùng require() cho các template

- ❑ Điểm đặc biệt và quan trọng nhất của việc kết hợp này là sau khi đã thực hiện nhiều trang có chèn các trang header.inc và footer.inc, ta có thể dễ dàng thay đổi trang header.inc và trang footer.inc này.
- ❑ Chỉ cần lưu lại các thay đổi thì tất cả các trang đã chèn các trang này sẽ tự động thay đổi.





## include

### □ Cách sử dụng

#### ● Đặc điểm

- include() cũng có cách sử dụng tương tự như require().
- Điểm khác nhau của include() và require(): khi nội dung bị lỗi thì dùng require() sẽ xuất hiện thông báo lỗi trong khi đó dùng include() sẽ chỉ xuất hiện cảnh báo.
- Trong những file có dùng require() thì không nên sử dụng các cấu trúc điều khiển vì sẽ không hiệu quả hoặc sử dụng include() thay cho require()

#### ● Cú pháp:

```
include("tên_và_đường_dẫn_của_file");
```



## include



### □ Ví dụ





## include

### □ require\_once() và include\_once()

- require\_once() và include\_once() là hai dạng biến đổi của require() và include()
- Mục đích sử dụng: đảm bảo cho việc chèn file chỉ được thực hiện duy nhất một lần.
- Chức năng: các hàm này trở nên hữu ích khi bắt đầu sử dụng chúng để chèn các thư viện và các hàm vào, sử dụng các cấu trúc này giúp tránh được việc chèn cùng một hàm hay thư viện lần thứ hai bởi vì khi khai báo lại một hàm đã được xây dựng sẽ phát sinh lỗi.



LTV Mã nguồn mở PHP – Module 2

18

## include

### □ require\_once() và include\_once()

- Hoạt động:
  - Việc chèn vào hai lần cùng một file thường xảy ra khi xây dựng các ứng dụng lớn, khi nhiều file thư viện khác nhau được chèn vào trong cùng một file chung. Trong trường hợp file được chèn vào bởi require\_once() hay include\_once() thì trong lần đầu tiên cách thức hoạt động của nó cũng giống như require() và include().
  - Tuy nhiên require\_once() và include\_once() nếu được gọi để chèn vào file đã được chèn thì file này sẽ không được chèn vào nữa.
- Hàm này là một công cụ thông minh cho việc tạo ra các thư viện dùng lại.



LTV Mã nguồn mở PHP – Module 2

19





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

## LTV MÃ NGUỒN MỞ PHP

### Module 2 – Bài 8: **SESSION - COOKIE**

Ngành LT & CSDL

[www.t3h.vn](http://www.t3h.vn)

2014



2014





## SESSION



- ☐ Khái niệm
- ☐ Cách thức hoạt động
- ☐ Khởi động session
- ☐ Đăng ký session
- ☐ Sử dụng session
- ☐ Hủy biến session



## SESSION



### □ Khái niệm

- Session PHP cho phép lưu trữ thông tin người dùng trên trình duyệt (tên người dùng, danh mục hàng hóa, ...) trong suốt quá trình làm việc của họ.
- Thông tin của session chỉ tạm thời và thông tin này sẽ bị xóa sau khi người dùng rời khỏi ứng dụng Web.
- Nếu cần, phải lưu trữ thông tin trong CSDL.



LTV Mã nguồn mở PHP – Module 2

4

## SESSION



### □ Cách thức hoạt động

- Session làm việc bằng cách tạo ra một địa chỉ duy nhất (UID) cho mỗi người sử dụng.
- UID có giá trị là một dãy số ngẫu nhiên.
- UID có thể được lưu trong COOKIE hoặc được truyền lên URL.
- Ngoài UID, có thể khai báo, khởi tạo và sử dụng một số biến session khác, tất cả các session này có giá trị cho mỗi người sử dụng khi họ truy cập đến ứng dụng Web.



LTV Mã nguồn mở PHP – Module 2

5

## SESSION



### ❑ Khởi động session

- Trước khi lưu trữ thông tin người dùng vào session, cần khởi động session.
- Chú ý: hàm khởi động session phải đặt phía trên thẻ HTML
- Cú pháp: `session_start();`



## SESSION



### ❑ Đăng ký session

- Sử dụng biến `$_SESSION` nhận và lưu trữ giá trị của biến session
- Cú pháp:  
`$_SESSION["tên biến session"] = "giá trị";`
- Ví dụ: tạo ra một biến session lưu tên đăng nhập của người dùng







## SESSION

### ❑ Sử dụng session

- Khi muốn sử dụng các biến session hoặc giá trị lưu trong biến session đã đăng ký => dùng biến `$_SESSION`.
- Cú pháp:  
`$gia_tri = $_SESSION["tên biến session"];`
- Ví dụ: đọc giá trị biến session tên đăng nhập



## SESSION

### ❑ Hủy toàn bộ các biến session

- Khi không cần dùng đến các biến session nữa thì có thể hủy bỏ toàn bộ các biến session đã đăng ký bằng hàm `session_destroy()`
- Cú pháp: `session_destroy();`



## SESSION



### ❑ Hủy một biến session

- Khi không cần dùng đến biến session nào thì có thể dùng hàm `unset()` để hủy bỏ biến session đó.

- Cú pháp:

```
unset($_SESSION["tên biến session"]);
```

- Ví dụ: hủy bỏ biến session tên đăng nhập





## COOKIE

- ☐ Khái niệm
- ☐ Khai báo cookie
- ☐ Sử dụng cookie
- ☐ Hủy cookie



## COOKIE

### ☐ Khái niệm cookie

- Được sử dụng để xác định thông tin của người dùng.
- Là một file nhỏ được server lưu trữ xuống từng máy tính của người dùng.
- Mỗi khi máy tính này yêu cầu một trang tới trình duyệt, nó cũng sẽ gửi theo cookie.
- Với PHP ta có thể tạo ra và sử dụng giá trị của biến cookie.





## COOKIE

### ❑ Khai báo cookie

- Chú ý: Khai báo cookie ở phía trên thẻ HTML
- Cú pháp: `setcookie(name, value, expire[, path, domain]);`
  - name: tên biến cookie
  - value: giá trị
  - expire: thời gian giới hạn dành cho cookie – đơn vị tính là giây. Nếu thời gian này không được thiết lập trong hàm `setcookie()`, biến cookie này sẽ còn hiệu lực cho đến khi người dùng xóa tập tin cookie.
  - path: đường dẫn
  - domain: tên miền của website



LTV Mã nguồn mở PHP – Module 2

14

## COOKIE



### ❑ Khai báo cookie

- Ví dụ: tạo ra một biến cookie người dùng có giá trị là “phuong”, thời gian giới hạn là một giờ
- Chú ý: giá trị của biến cookie sẽ tự động được URL mã hóa khi gửi cookie đi, và tự động giải mã khi nhận cookie về. (Nếu không muốn URL mã hóa thì dùng hàm `setrawcookie()`)



LTV Mã nguồn mở PHP – Module 2

15



## COOKIE

### ☐ Sử dụng cookie

- Dùng biến `$_COOKIE` để đọc giá trị biến cookie
- Cú pháp: `$gia_tri = $_COOKIE["tên biến cookie"];`
- Ví dụ: đọc giá trị của biến cookie người dùng



## COOKIE

### ☐ Hủy cookie

- Khi muốn hủy một biến cookie thì cần kiểm tra lại thời gian giới hạn dành cho biến cookie này
- Sử dụng hàm `setcookie()` để hủy bằng cách đặt giá trị của biến cookie bằng "" và thời gian = - thời gian giới hạn
- Cú pháp: `setcookie(name, "", time() - thời gian giới hạn)`



## COOKIE



### ❑ Hủy cookie

- Ví dụ: hủy biến cookie người dùng với thời gian giới hạn của biến này là 3600s





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

## LTV MÃ NGUỒN MỞ PHP

### Module 2 – Bài 9: *Hoàn chỉnh website*

Ngành LT & CSDL

[www.t3h.vn](http://www.t3h.vn)

2014



2014





## Các công việc cần thực hiện



### ☐ Để một website hoàn chỉnh, cần:

- Kiểm tra, rà soát lại danh mục các chức năng cần phải thực hiện cho website
  - Đối với các chức năng chưa thực hiện:
    - Thực hiện hết các chức năng nếu còn đủ thời gian
    - Ưu tiên chọn các chức năng cần thiết, quan trọng hơn để thực hiện nếu không đủ thời gian
  - Đối với các chức năng đã thực hiện: kiểm tra xem các chức năng này đã hoạt động tốt và ổn định chưa?







## Các công việc cần thực hiện

---

### ☐ Để một website hoàn chỉnh, cần:

- Ráp các chức năng đã xây dựng xong vào website
- Kiểm tra lại tính ổn định của toàn bộ website
- Đóng gói website

### ☐ Đưa website lên host

### ☐ Chạy thử website và kiểm tra sau khi đã đưa lên host





## Đưa website lên host

- ❑ Sau khi đã hoàn chỉnh website, công việc cần phải làm tiếp theo là xuất bản website lên Internet. Trước tiên cần phải đăng ký một host (có đóng phí hoặc miễn phí), chọn một tên miền thích hợp, upload Website lên host đã đăng ký.



## Đưa website lên host

- ❑ **Đối với host miễn phí**
  - Tìm một website cung cấp host miễn phí dành cho website viết bằng ngôn ngữ PHP & MySQL
  - Vào mục đăng ký, và điền đầy đủ các thông tin cần thiết cho việc xin một host miễn phí (trong đó có cả tên miền của website)
  - Đọc các thông tin hướng dẫn và các điều khoản đi kèm
  - Xác nhận lại để hoàn thành
- > Sau khi đăng ký thành công, các thông tin về host miễn phí đã xin sẽ được gửi theo địa chỉ mail đã đăng ký.





## Đưa website lên host

---

### ☐ Upload website

- Đăng nhập vào website cung cấp host đã mua hoặc xin bằng tài khoản đã đăng ký được gửi về qua email
- Chọn thư mục sẽ lưu trữ website khi upload
- Upload website lên host vào thư mục này





## Chạy thử website và kiểm tra

- ❑ Sau khi đã đưa website lên host, công việc cần phải làm tiếp theo là chạy thử và kiểm tra website trên Internet .



## Chạy thử website và kiểm tra

- ❑ Thực hiện
  - Nhập đường dẫn website đã đăng ký vào URL > nhấn Go
  - Kiểm tra lại việc hiển thị của trang web: có đầy đủ cá thành phần? Bố cục có đầy đủ?
  - Kiểm tra xem các chức năng vẫn hoạt động chính xác? Các hiệu ứng có hiển thị trên các trình duyệt khác nhau?
  - Tiến hành chỉnh sửa khi có vấn đề



