

# Tài liệu thực hành lập trình WEB

---

*Giảng viên: Lâm Chí Nguyễn*

## Chương I Đặt vấn đề

### I.1. Yêu cầu đề tài:

Thiết kế WebSite trao đổi thông tin và tìm kiếm việc làm cho sinh viên và nhà tuyển dụng. Website cần có các chức năng chính :

- Trang chủ liệt kê các chủ đề mới nhất của Website : thông tin tìm nhân viên, thông tin tìm việc làm
- Trang cá nhân : liệt kê thông tin cá nhân của người dùng, liệt kê các tin tức do người dùng tạo ra, liệt kê các thông tin mới của website.
- Trang soạn thảo tin tức : cho người dùng tạo tin tức và đăng
- Trang quản lý : cho phép quản lý tin tức của mỗi người dùng
  - Thêm tin tức
  - Xóa tin tức
  - Vô hiệu hoá tin tức

### I.2. Người dùng trong Website:

Người sử dụng trong website bao gồm các đối tượng sau :

- Khách vãng lai
  - Được phép vào trang tin tức của Website
  - Được phép xem tin tức
  - Được phép tìm kiếm tin tức theo chủ đề
- Người dùng : **Người tìm việc/ Người cung cấp việc làm**
  - Phải thực hiện đăng ký
  - Có các chức năng của **khách vãng lai**.
  - Được phép tạo tin tức
  - Được phép quản lý các tin tức của mình
  - Không được phép xóa tin tức
- Quản trị viên hệ thống
  - Tài khoản được tạo ra bởi nhà quản trị mạng cao nhất(cài đặt hệ thống).
  - Có các chức năng của **người dùng**
  - Được phép xem và xóa tin tức của tất cả các người dùng
  - Được phép thực hiện các chức năng đối với tài khoản người dùng :
    - Thêm tài khoản
    - Khoá tài khoản

- Xoá tài khoản

### ***I.3. Các đối tượng thông tin khác :***

#### *a. Tin tức :*

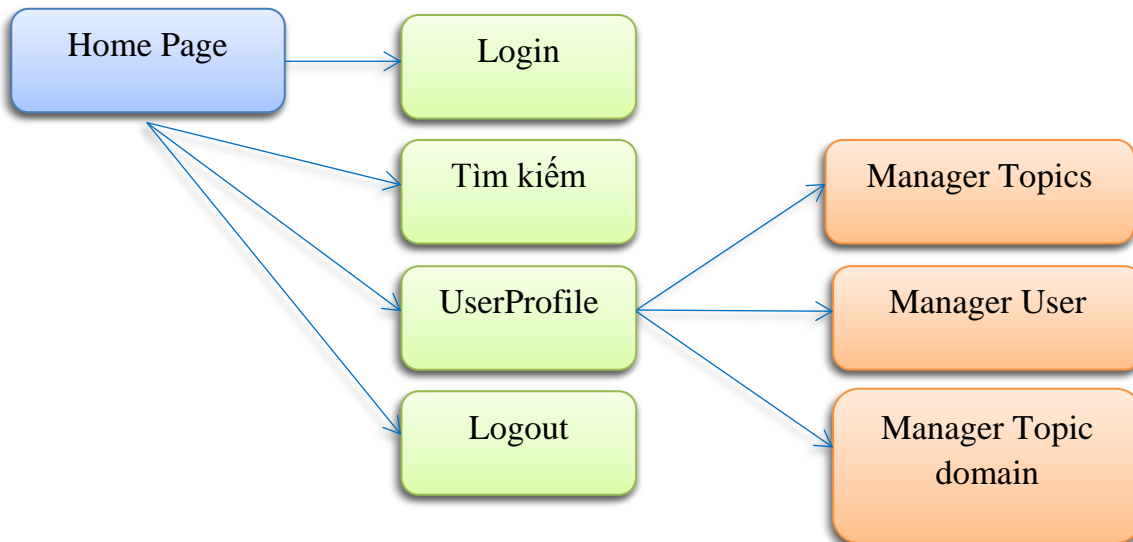
Các tin tức của hệ thống bao gồm :

- Thông báo chung
- Thông báo tìm việc làm
- Thông báo tìm nhân viên
- Tin nhắn riêng

#### *b. Lĩnh vực tin tức*

Các thông báo sẽ được phân chia theo các lĩnh vực cụ thể để thực hiện chức năng phân loại tin tức. Các lĩnh vực sẽ được quản lý bởi người quản trị hệ thống

### ***I.4. SiteMap***



### I.5. Wire frame – layout

#### Home – main layout

<i>Logo</i>	<i>Banner – tiêu đề chính</i>	
<i>menu</i>	<i>Tin tức do quản trị đăng</i>	<i>Tin</i>
	<i>Tin tức do người dùng hiện</i>	<i>Tin</i>
	<i>Tin tức do người khác đăng</i>	<i>Tin</i>

#### Login –Logout layout

<i>Logo</i>	<i>Banner – tiêu đề chính</i>	
<i>menu</i>	<i>Nhập thông tin</i>	<i>Thông báo của hệ thống</i>

### Profile – main layout

[ Logo ]	[ Banner – tiêu đề chính ]	
[ menu ]	[ Thông tin người dùng ]	[ Tin ]
		[ Tin ]

### Manager Topic Layout

[ Logo ]	[ Banner – tiêu đề chính ]	
[ menu ]	[ Quản lý người dùng ]	
		[ Quản lý tin tức ]
	[ Quản lý lĩnh vực đăng tin ]	

**Sinh viên cần đề xuất thêm các frame khác để chi tiết hoá hệ thống WebSite**

## Chương II Ngôn ngữ HTML

### II.1. Thẻ nội dung - Tag

Trang nội dung HTML được trình bày thông qua các thẻ nội dung (**Tag**). Một nội dung trong html có thể là :văn bản, hình ảnh, phim, âm thanh,các nội dung khác,... Để trình bày một nội dung ta phải dùng một thẻ nội dung để đánh dấu (vị trí, nội dung, hiệu ứng). Mức độ ảnh hưởng của thẻ lồng nhau: nếu công việc giống nhau thì thẻ nào gần nhất sẽ có tác dụng.

Mỗi tag được đánh dấu bằng cú pháp : *<Tag bắt đầu>Nội dung của tag<Tag kết thúc>*.

**Ví dụ :** <p> Chào các bạn </p>

Tag bắt đầu

Tag kết thúc

#### a. Thuộc tính của thẻ nội dung

Mỗi một thẻ đều có các thuộc tính của mình;dùng để đặt ra các quy định và các thông tin kèm theo nội dung. Các thuộc tính luôn được đặt ở thẻ bắt đầu. Mỗi thuộc tính phải bao gồm cả hai thành phần : **tên** và **giá trị**. Tên luôn viết chữ thường, giá trị luôn đặt trong cặp dấu “ ”

Ví dụ :

<a href="http://www.w3schools.com">This is a link</a>

Trong đó thẻ <a> có thuộc tính **href**, giá trị là “**http://www.w3schools.com**”

#### b. Cấu trúc của một trang nội dung

```
<html>
  <head>
    <meta name="Author" content="Nguyễn Văn An">
    <title>Trang WEB đầu tiên của tôi</title>
  </head>
  <body>
    <p>Nội dung của trang WEB</p>
  </body>
</html>
```

## II.2. Các thẻ nội dung trình bày cấu trúc một tập tin html

<http://www.htmldog.com>

### a. <html>

Đây là thẻ dùng để đánh dấu một tập tin html. Trong một trang HTML bắt buộc phải có duy nhất thẻ <html> ...</html>

### b. <head>

Đây là thẻ trình bày các thông tin chung về trang Web, thông tin sẽ không hiển thị trực tiếp trên màn hình. Các website tìm kiếm sẽ căn cứ vào nội dung của thẻ <head> để phân loại và tìm kiếm thông tin.

Trong thẻ <head> có thể chứa các thẻ : <title>, <meta>, <style>, <script>, <base>, <link>

Ví dụ nội dung một thẻ <head>

```
<head>
<title>Vi du lap trinh Web</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<meta name="keywords" content="computer, hardware, student" />
<link rel="icon" type="image/png" href="myicon.png" />
<link rel="stylesheet" type="text/css" href="default.css" />
<script src="/script/somescript.js" type="text/javascript"></script>
<base href="http://www.somedomain.com/directory/" />
<style type="text/css">
    body {
        color: red;
        background-color: yellow;
        font-size: 80%;
    }
</style>
</head>
```

c. `<body>`

Thẻ `<body>` dùng để xác định nội dung được trình bày trong 01 Webpage, nó được bắt đầu sau khi kết thúc `</head>` và kết thúc trước `</html>`. Trong một trang nội dung chỉ có duy nhất một thẻ `<body>`.

Các thuộc tính cơ bản của `<body>` :

Attribute	Value	Description
alink	color	Màu của liên kết khi được kích hoạt
background	URL	Xác định ảnh nền của trang Web
bgcolor	color	Xác định màu nền của trang Web
link	color	Màu của liên kết khi chưa vào lần nào
text	color	Xác định màu của văn bản trong trang
vlink	color	Màu của liên kết sau khi đã vào một lần

d. `<div>`

Thẻ `<div></div>` dùng để xác định một nội dung bất kỳ trong trang web. Thường dùng kết hợp với CSS, để tiến hành trình bày nội dung theo từng khối (từng khung hình trên trang WEB).

e. `<span>`

Thẻ `<span></span>` dùng để xác định một nhóm ký tự trong một đoạn văn. Thường dùng kết hợp với CSS để trình bày các hiệu ứng trên văn bản.

### II.3. Các thẻ trình bày văn bản :

a. `<p>`

Trình bày nội dung của 01 đoạn văn; trong đoạn văn mỗi từ cách nhau bởi 01 khoảng trắng. Trình duyệt tự động loại bỏ các ký tự trắng dư thừa

b. Thẻ trình bày tiêu đề

Gồm có 06 thẻ chính : `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`. Kích thước của văn bản sẽ thay đổi theo thứ tự giảm dần (`h1` → `h6`).

Các trang tìm kiếm trên mạng sẽ căn của theo các thẻ tiêu đề để xác định ý nghĩa của nội dung trong webpage để đưa vào kết quả tìm kiếm. Do đó lập trình viên cần sử dụng đúng ngữ cảnh của các thẻ `<hx>` trong trang web.

c. `<pre>`

Trình bày đoạn văn nhưng giữ nguyên định dạng và các ký tự hệ thống của đoạn văn gốc (bao gồm khoảng trắng, /, \, “, ”).

d. `<br/>`

đầu xuống dòng

```
<!DOCTYPE html>
<html>
<body>
<em>Emphasized text</em><br>
<strong>Strong text</strong><br>
<dfn>Definition term</dfn><br>
<code>A piece of computer code</code><br>
<samp>Sample output from a computer program</samp><br>
<kbd>Keyboard input</kbd><br>
<var>Variable</var>
</body>
</html>
```

Sinh viên soạn thảo một tập tin html theo nội dung trên và rút ra kết luận về các thẻ bên trong

Các thẻ cơ bản để định dạng văn bản :

*p, h1, h2, h3, h4, h5, h6, strong, em, abbr, acronym, address, bdo, blockquote, cite, q, code, ins, del, dfn, kbd, pre, samp, var, br*

#### **II.4. Thẻ liên kết nội dung <a> :**

Thẻ <a> dùng để tạo ra liên kết đến một trang nội dung khác. Có 02 loại liên kết trong một trang: **liên kết nội, và liên kết ngoại.**

Một ví dụ về thẻ <a> :

```
<a href="Đường liên kết">Thông tin hiển thị</a>
```

##### **a. Liên kết nội :**

Trong trang nội dung để sử dụng liên kết nội, trước tiên ta cần xác định vị trí của một văn bản thông qua thẻ : <a id = "vị\_trí"> Nội dung văn bản </a>

Liên kết nội là một liên kết dẫn đến một nội dung trong cùng một trang Webpage, ta sử dụng thẻ <a href="#vị\_trí">Đến nội dung !</a>

Xem ví dụ tại :

[http://www.w3schools.com/tags/tryit.asp?filename=tryhtml5\\_a\\_href\\_anchor](http://www.w3schools.com/tags/tryit.asp?filename=tryhtml5_a_href_anchor)



*b. Liên kết ngoài :*

Liên kết ngoài một liên kết đến một trang nội dung khác. Một thẻ liên kết <a> dùng cho liên kết ngoài như sau :

<a href= “liên kết” target= “\_blank|\_self|\_parent|\_top|frameName”> văn bản </a>

Trong đó có các thuộc tính quan trọng như sau

- ✓ href : xác định liên kết ngoài :
  - liên kết tương đối : đi đến các trang trong cùng một Site. Ví dụ :
    - href= “trang1.html”
  - liên kết tuyệt đối : đi đến các trang (cùng site hoặc khác site)
    - href= “www.cit.ctu.edu.vn/index.PHP”
- ✓ target : xác định vị trí hiển thị trang nội dung liên kết
  - \_blank : hiển thị trên một trình duyệt mới
  - \_self : hiển thị trên cùng trình duyệt (giá trị mặc định)
  - \_parent, \_top: trong trường hợp trang Webpage có nhiều khung (frame), trang liên kết sẽ hiện ra ở trình duyệt chính
  - frameName : trong trường hợp trang Webpage có nhiều khung (frame), trang liên kết sẽ hiện ra ở khung có tên tương ứng frameName

## **II.5. Thẻ trình bày về hình ảnh**

Sử dụng thẻ <img> và kết hợp với các thuộc tính để trình bày :

Attribute	Value	Description
align	top bottom middle left right	Canh lề hình ảnh
alt	text	Thông báo hiện ra khi không xác định được hình ảnh
border	pixels	Xác định độ rộng của khung bao quanh ảnh
height	pixels	Xác định chiều cao hình ảnh
src	URL	Xác định vị trí hình ảnh
width	pixels	Xác định chiều rộng của hình ảnh

Ví dụ :

<img src= “myicon.png” alt= “vidu” height= “128” width= “128”>

## II.6. Trình bày nội dung dạng bảng <table>

Trình bày nội dung của một bảng <table></table>

### a. Các thuộc tính cơ bản của thẻ <table>

Attribute	Value	Description
<u>align</u>	left center right	Canh lề các văn bản bên trong của bảng
<u>bgcolor</u>	<i>rgb(x,x,x)</i> <i>#xxxxxx</i> <i>colorname</i>	Màu nền của bảng
<u>border</u>	1 ""	Độ rộng của đường kẻ trong bảng
<u>cellpadding</u>	<i>pixels</i>	Khoảng cách giữa văn bản và khung của ô nội dung
<u>cellspacing</u>	<i>pixels</i>	Khoảng cách giữa các ô nội dung
<u>width</u>	<i>pixels</i> %	Độ rộng của bảng

### b. <tr>

Trình bày nội dung của một dòng, trong một dòng <tr> còn có nhiều thẻ <td>, <th>.

Các thuộc tính cơ bản của một <tr>

Attribute	Value	Description
<u>align</u>	Right, left, center, justify, char	Canh lề nội dung trên một dòng
<u>bgcolor</u>	<i>rgb(x,x,x)</i> <i>#xxxxxx</i> <i>colorname</i>	Màu sắc trên một dòng
<u>valign</u>	Top, middle, bottom, baseline	Canh lề văn bản theo chiều cao

c. <td>

Trình bày nội dung của một ô (cell) trong một dòng, có thể trình bày nội dung của một bảng khác trong một ô.

Các thuộc tính cơ bản của <td>

Attribute	Value	Description
<u>align</u>	Left, right, center, justify, char	Canh lề nội dung trên một dòng
<u>bgcolor</u>	<i>rgb(x,x,x)</i> <i>#xxxxxx</i> <i>colorname</i>	Màu sắc của ô
<u>colspan</u>	<i>number</i>	Mở rộng ô theo chiều ngang
<u>rowspan</u>	<i>number</i>	Mở rộng ô theo chiều cao
<u>height</u>	<i>pixels</i> <i>%</i>	Chiều cao của một ô
<u>width</u>	<i>pixels</i> <i>%</i>	Độ rộng của một ô
<u>valign</u>	top middle bottom baseline	Canh lề theo chiều cao

d. <th>

Trình bày nội dung của một ô ở dòng tiêu đề

Ví dụ :

```
<table border="1">
<tr>
<th>Môn học</th><th>Điểm đánh giá</th>
</tr>
<tr>
<td>Lập trình căn bản</td><td>A</td>
</tr>
<tr>
<td>Cấu trúc dữ liệu</td><td>B+</td>
</tr>
</table>
```

e. *Các thẻ khác :*

<thead> : xác định tiêu đề của một bảng

<tfoot> : xác định phần cuối của một bảng

<tbody> : xác định phần thân của một bảng

f. *Cấu trúc cơ bản của một bảng*

<table>

<caption>.....</caption>

<thead>

<tr><th>... </th></tr>

</thead>

<tbody>

<tr><td>....</td></tr>

<tr><td>....</td></tr>

</tbody>

<tfoot>

<tr><th>... </th></tr>

</tfoot>

</table>

g. *Các hình thức mở rộng trình bày bảng :*

- Ghép cột : sử dụng thuộc tính **colspan**
- Ghép dòng : sử dụng thuộc tính **rowspan**

## II.7. Bài tập :

**Bài 1 :** Sinh viên thiết kế tập tin có tên là *userprofile.html*, có nội dung như hình sau :

### Sơ yếu lý lịch cá nhân



**Họ tên :** Nguyễn Văn An

**Năm sinh :** 1990

**Nơi sinh :** Cần Thơ

**Bài 2 :** sinh viên thiết kế tập tin có tên *topic.html*, có nội dung như hình sau :

Tin tức 1



#### Ảnh dụng BlackBerry X10 xuất hiện trước lễ công bố

RIM được cho là sẽ trình làng 2 điện thoại khác nhau gồm smartphone với màn hình cảm ứng, không có bàn phím cứng Z10 và smartphone với bàn phím Qwerty X10 vào 30/1.

Bản Z10 đã có nhiều ảnh và video dùng thử, thậm chí được so sánh với iPhone 5... Một ảnh chụp màn hình cơ sở dữ liệu của hãng bán lẻ Carphone ở Anh cho thấy thiết bị sẽ có giá tầm 760 USD, trong khi đó một nguồn tin khác tiết lộ với EnGadget rằng máy sẽ được bán dưới 200 USD kèm hợp đồng thuê bao 2 năm ở Mỹ. Z10 có khả năng chạy chip lõi kép 1,5 GHz, màn hình 1.280 x 768, RAM 2 GB, bộ nhớ trong 16 GB hoặc 32 GB, camera chính 8 megapixel và camera mặt trước 2 megapixel.

**Bài 3 :** sinh viên bổ sung nội dung cho tập tin có tên *userprofile.html*, với các thông tin cá nhân của mình theo cấu trúc như sau :

- Hình ảnh , canh lề phải
- thông tin cá nhân
- Quá trình học tập : (Cấp học, năm học, kết quả)
  - o Phổ thông trung học
  - o Đại học
  - o Các lớp học nghiệp vụ khác (Lập trình WEB, CCNA)
- Kinh nghiệm nghề nghiệp : (chức vụ, phòng, công ty, năm làm việc)

**Bài tập kết thúc :** sinh viên thiết kế các layout của đề tài đã được mô tả ở chương 1

Bài giải:

**Bước 1 :** Tại tập tin **main.html** có nội dung như sau :

```
<html>
<body width="800">
</body>
</html>
```

**Bước 2 :** chèn 01 bảng <table> vào trong phần body trong bước này sinh viên cần lưu ý các kích thước chiều ngang, chiều dọc, sinh viên sử dụng valign="top|middle|bottom" để điều chỉnh vị trí canh lề trên, dưới trong một ô

```
<table id="main" width="100%" border="1">
<tr height="150">
<td width="20%">Logo</td>
<td>banner</td>
</tr>
<tr height="650">
<td>menu</td>
<td>Topics</td>
</tr>
</table>
```

**Bước 3 :** trong các ô : menu, topics. Chèn thêm các bảng mới để phân chia khung nội dung

```
<html>
<body width="800">
<table id="main" width="100%" border="1">
<tr height="150">
<td width="20%">Logo</td>
<td>banner</td>
</tr>
<tr height="450" valign="top">
<td>menu<br>
<table id="menu" border="1" width="100%">
<tr><td>Chức năng 1</td></tr>
<tr><td>Chức năng 2</td></tr>
<tr><td>Chức năng 3</td></tr>
<tr><td>Chức năng 4</td></tr>
</table>
</td>
<td>Topics<br>
<table id="Topic" border="1" height="90%" width="100%">
<tr height="30%"><td width="100%">Chủ đề 1</td></tr>
<tr height="30%"><td width="100%">Chủ đề 2</td></tr>
<tr height="30%"><td width="100%">Chủ đề 3</td></tr>
</table>
</td>
</tr>
</table>
</body>
</html>
```

Sinh viên tiếp tục điều chỉnh chỉnh các nội dung để có các khung Template còn lại

**Bài tập về nhà :** sinh viên nghiên cứu tác dụng của các thẻ tag còn lại tại WebSite : [www.w3schools.com](http://www.w3schools.com)

## Chương III Ngôn ngữ JavaScript

### III.1. JavaScript

#### a. Thẻ Script :

Thẻ <script> dùng để chèn vào trang web những đoạn chương trình xử lý (có hai ngôn ngữ được hỗ trợ chính là : javascript, vbscript)

```
<SCRIPT type="text/vbscript"></SCRIPT>
```

```
<SCRIPT type="text/javascript"></SCRIPT>
```

<script></script> sẽ được hiểu là đoạn chương trình viết bằng ngôn ngữ javascript.

#### Ví dụ :

```
<script>

document.write("<h1>This is a heading</h1>");

document.write("<p>This is a paragraph.</p>");

</script>
```

#### b. Ngôn ngữ javascript :

Một ngôn ngữ script là một phiên bản đơn giản của một ngôn ngữ lập trình cơ bản. JavaScript là một ngôn ngữ lập trình thông dụng trong tiến trình kế WEB, nó có thể được đưa vào trong một trang html. JavaScript được hỗ trợ bởi hầu hết trình duyệt và rất dễ dàng cho việc tìm hiểu nó.

#### Ví dụ :

```
<html>
<body>
<script>
document.write("<h1>This is a heading</h1>");
document.write("<p>This is a paragraph.</p>");
</script>
</body>
</html>
```

#### **Sinh viên tìm kết quả trên màn hình**



### ***JavaScript có thể được sử dụng cho các mục đích sau :***

- ✓ Xuất dữ liệu vào trang html.
- ✓ Lập trình cho các sự kiện
- ✓ Kiểm tra dữ liệu nhập vào.
- ✓ Thay đổi nội dung của một thẻ
- ✓ Thay đổi giá trị của thuộc tính trên thẻ
- ✓ Thay đổi style (hình thức trình bày) của các thẻ trong html.

#### ***c. Biến dữ liệu trong javascript :***

Một biến có thể là một cái tên ngắn gọn (x,y) hoặc một tên đầy đủ (age, sum, totalvolume). Các quy tắc đặt tên biến :

- ✓ Có thể dùng ký tự \$, \_ trong tên biến
- ✓ Có thể dùng các ký tự từ A-Z, a-z
- ✓ Có thể dùng các ký tự từ 0-9
- ✓ Tên biến phải bắt đầu là một ký tự chữ
- ✓ Phân biệt chữ hoa, chữ thường.

#### ***Khai báo biến :***

```
var x;
```

#### ***Đặt giá trị cho biến :***

```
x=1;  
x= "Chao ban";  
x=new array();
```

#### ***Vừa khai báo vừa đặt giá trị :***

```
var name="Doe", age=30, job="carpenter";
```

#### ***d. Kiểu dữ liệu trong JavaScript :***

Các biến dữ liệu trong JavaScript có kiểu dữ liệu động. Nghĩa là cùng một biến khai báo có thể dùng cho nhiều kiểu dữ liệu khác nhau. Kiểu dữ liệu của một biến chỉ được xác định khi ta biết giá trị của nó. Tuy nhiên trong javascript có các kiểu dữ liệu cơ bản như sau : *String, Number, Boolean, Array, Object, Null, Undefined*.

1. Kiểu String : được đánh dấu bằng cặp dấu “ ” hoặc ‘ ’

```
var carname1="Volvo XC60";  
var carname2='Volvo XC60';  
var answer1="It's alright";  
var answer2="He is called 'Johnny'";  
var answer3='He is called "Johnny"';
```

2. Kiểu số Number :

```
var x = 10; y=123e5;  
var y =10.5;z=123e-5;
```

3. Kiểu dữ liệu logic **boolean** :

```
var x=true
```

```
var y=false
```

#### 4. Kiểu dữ liệu mảng array :

```
var cars=new Array();  
cars[0]="Saab";  
cars[1]="Volvo";  
cars[2]="BMW";
```

hoặc :

```
var cars=new Array("Saab","Volvo","BMW");  
var cars=["Saab","Volvo","BMW"];
```

Thêm phần tử vào cuối mảng :

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.push("Kiwi");
```

Xoá phần tử cuối khỏi mảng :

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.pop();
```

Số phần tử trong mảng :

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.length();
```

Tìm phần tử trong mảng :

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];  
fruits.indexOf("Apple");// ket qua bang 2  
fruits.indexOf("Kiwi");// ket qua bang -1
```

Sinh viên nghiên cứu thêm các thao tác khác :

- ✓ Ghép hai mảng thành 01 mảng                      concat()
- ✓ Sắp xếp các phần tử trong mảng                      sort()
- ✓ Xoá hoặc thêm phần tử                                  splice()
- ✓ Lấy một đoạn trong mảng                              slice()

Địa chỉ tham khảo : [http://www.w3schools.com/jsref/jsref\\_obj\\_array.asp](http://www.w3schools.com/jsref/jsref_obj_array.asp)

Kiểu dữ liệu đối tượng :

Dữ liệu đối tượng có tính chất tương tự như một cấu trúc – struct – trong ngôn ngữ C/C++

```
var person={  
  firstname : "John",  
  lastname  : "Doe",  
  id        : 5566  
};
```

Tất cả các kiểu dữ liệu trong javascript đều là kiểu dữ liệu đối tượng. Mỗi một biến dữ liệu tương đương với 01 đối tượng. Mỗi đối tượng có 02 thành phần chính – dữ liệu thành phần, hàm thành phần(chức năng trên đối tượng).

Ví dụ :

```
var s= "Chao ban"; //kiểu dữ liệu String  
s.toUpperCase();//hàm thành viên đổi chữ hoa  
var len=s.length;//dữ liệu thành viên độ dài length
```

Để tìm hiểu chi tiết về từng lớp đối tượng trong javascript sinh viên vào trang nội dung <http://www.w3schools.com/jsref/default.asp>

*e. Câu lệnh trong JavaScript :*

Trong JavaScript, câu lệnh dùng để yêu cầu trình duyệt thực hiện các công việc. Ví dụ :

```
<p id="demo"> Chao Dolly</p>
```

```
<script>
```

```
document.getElementById("demo").innerHTML="Hello Dolly";
```

```
</script>
```

Câu lệnh trong <script> dùng để yêu cầu trình duyệt (document) thay đổi nội dung của phần tử có tên (id) là “demo” với nội dung mới (innerHTML) “Hello Dolly”.

Dấu ; dùng để phân cách 02 câu lệnh javascript.

**JavaScript code** là một tập hợp các câu lệnh javascript.

**JavaScript Code Blocks** – khối lệnh. Các câu lệnh có thể nhóm thành một khối đánh dấu bằng cặp dấu {code blocks}

Trong Javascript các từ khoá phân biệt khác nhau giữa chữ in hoa và in thường.

Trình bày câu lệnh trên nhiều dòng : chúng ta có thể viết xuống dòng trong một câu lệnh bằng cách đặt dấu \ **trong một chuỗi ký tự**

```
document.write("Hello \nWorld!");
```

Tuy nhiên trường hợp sau không chấp nhận được

```
document.write \n  
("Hello World!");
```

*f. Các cấu trúc câu lệnh :*

JavaScript là một phiên bản thu gọn của ngôn ngữ Java do đó nó có tất cả các cấu trúc lập trình của ngôn ngữ Java. Đồng thời các cấu trúc này cũng có tính tương đồng với ngôn ngữ C/C++

Cấu trúc rẽ nhánh : if ... else

```
if (condition)
{
    code to be executed if condition is true
}
```

```

else
{
    code to be executed if condition is not true
}

```

**Cấu trúc rẽ nhánh : switch .. case**

```

switch(n)
{
case 1:
    execute code block 1
    break;
case 2:
    execute code block 2
    break;
default:
    code to be executed if n is different from case 1 and 2
}

```

**Cấu trúc lặp : for(.....){}**

```

for (var i=0;i<cars.length;i++)
{
    document.write(cars[i] + "<br>");
}

```

**Cấu trúc lặp: while**

```

while (condition)
{
    code block to be executed
}

```

**Cấu trúc lặp: do/while**

```

do
{
    code block to be executed
}
while (condition);

```

**Kiểm soát ngoại lệ : try....catch**

```

try
{
    //Run some code here
}
catch(err)
{
    //Handle errors here
}

```

***[http://www.w3schools.com/js/tryit.asp?filename=tryjs\\_throw\\_error](http://www.w3schools.com/js/tryit.asp?filename=tryjs_throw_error)***

*g. Lập trình hàm trong JavaScript :*

Một hàm trong JavaScript có cấu trúc như sau :

```

function myFunction(var1,var2)
{
    some code;
    [return value;]
}

```

**Các ví dụ về gọi hàm :**

***Ví dụ 1***

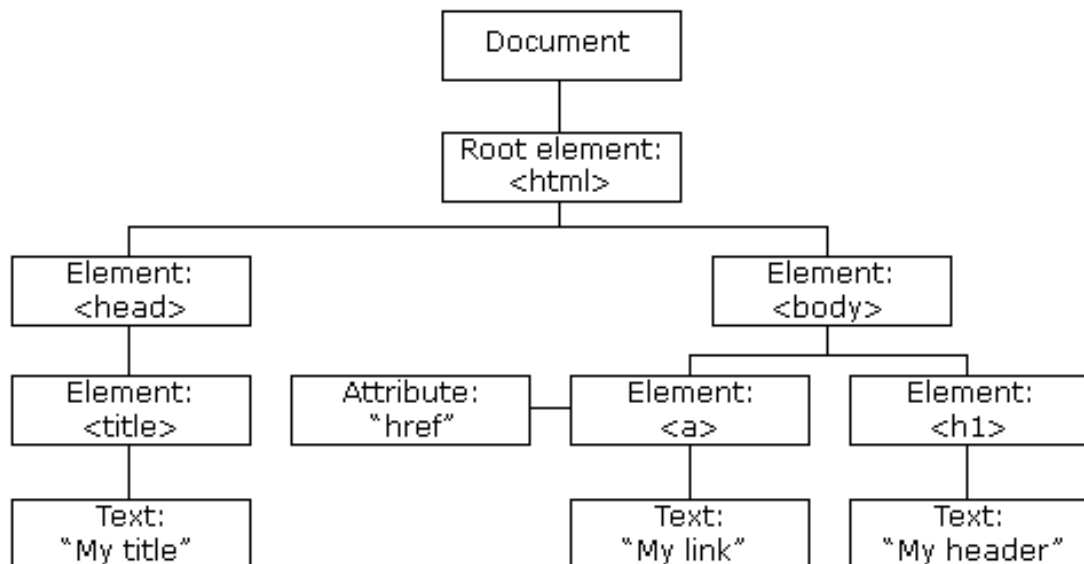
```
<script>  
Some code;  
myFunction(value1, value2);  
</script>
```

**Ví dụ 2**

```
<button onclick="myFunction('Bob','Builder')">Try it</button>
```

### III.2. Mô hình DOM (Document Object Model).

Mô hình DOM được diễn tả thông qua hình sau :



Trong đối tượng **document** ta có thể thao tác trên các đối tượng **element** (mỗi element tương ứng với một thẻ – tag). Trong các đối tượng element có thể chứa các đối tượng : element, attr.

Các hàm thông dụng của đối tượng **document** :

Tên hàm	Diễn giải
<b>createAttribute()</b>	Tạo đối tượng thuộc tính
<b>createElement()</b>	Tạo một đối tượng thẻ
<b>createTextNode()</b>	Tạo một thẻ văn bản
<b>getElementById()</b>	Lấy các thẻ theo thuộc tính id
<b>getElementsByTagName()</b>	Lấy các thẻ theo tên thẻ
<b>write()</b>	Hiển thị văn bản không xuống dòng
<b>writeln()</b>	Hiển thị văn bản có xuống dòng

Các thuộc tính thông dụng của **document** :

Tên hàm	Diễn giải
<b>activeElement</b>	Element đang làm việc (input,textarea)
<b>bgcolor</b>	Màu nền của trang
<b>body</b>	Element body
<b>documentElement</b>	Element html
<b>location</b>	Địa chỉ của trang nội dung
<b>title</b>	Element title
<b>head</b>	Element head

Đối tượng thuộc lớp **element** được dùng để thao tác trên các thẻ nội dung của trang web(có thể tương ứng 01 element tương ứng với một thẻ nội dung).

Trích lọc thẻ nội dung – element – từ đối tượng **document** :

```
var x = document.getElementById("vidu");
var y = document.getElementsByTagName ("p");
```

Các thuộc tính thường sử dụng trên đối tượng element :

Tên thuộc tính	Diễn tả
<b>className</b>	Đặt giá trị cho thuộc tính <b>class</b>
<b>id</b>	Đặt giá trị cho thuộc tính <b>id</b>
<b>innerHTML</b>	Đặt nội dung của thẻ tag (element)

Các hàm thường sử dụng trên đối tượng element :

Method	Description
getAttribute(attributename)	Lấy giá trị của một thuộc tính
getAttributeNode(attributename)	Lấy thuộc tính của một node
getElementsByTagName()	Lấy các element con của thẻ tag
hasAttribute()	Kiểm tra có thuộc tính hay không
removeAttribute()	Xoá thuộc tính
removeAttributeNode()	Xoá thuộc tính của một node
setAttribute()	Đặt giá trị cho một thuộc tính
setAttributeNode()	Đặt một thuộc tính cho một node
appendChild()	Thêm một element hoặc một attribute
hasChildNodes()	Kiểm tra xem có node con hay không
removeChild()	Xoá node con
replaceChild()	Thay thế node con

Đối tượng attr - Attribute

Các thuộc tính của attr

Thuộc tính	Diễn giải
isId	Kiểm tra có phải là <i>id</i>
name	Tên của thuộc tính
ownerElement	Node cha của thuộc tính
value	Giá trị của thuộc tính

### ***III.3. Lập trình cho các sự kiện trong javascript :***

Trong javascript chúng ta có thể lập trình sự kiện(event) cho các thẻ nội dung với các sự kiện như sau :

Tên sự kiện	Diễn giải
onload	Sự kiện khi đối tượng được hiển thị
onunload	Sự kiện khi đối tượng được đóng
onchange	Sự kiện khi đối tượng đổi giá trị
onmouseover	Sự kiện khi di chuyển chuột trong diện tích đối tượng
onmouseout	Sự kiện khi di chuyển chuột ra khỏi diện tích đối tượng
onmousedown	Sự kiện khi ấn chuột trong diện tích đối tượng
onmouseup	Sự kiện khi buông chuột trong diện tích đối tượng
onclick	Sự kiện khi nhấp chuột trái chuột trong diện tích đối tượng

### III.4. Bài tập

Bài 1. Sinh viên sử dụng đối tượng **document** để hiển thị thông tin trong

```
<html>
<html>
<body>
<script language="javascript">
document.write("<h1>Vi du 1 cua javascript</h1>");
document.write("<p>Đoạn văn</p>");
</script>
</body>
</html>
```

Bài 2. Hiển thị hộp thông báo khi click chuột vào đoạn văn

```
<html>
<body>
<p onclick="myFunction('Sinh vien', 'Thuc tap')">
    Try it
</p>
<script>
function myFunction(name, job)
{
    alert("Chao " + name + ", lam " + job);
    //hien thi thong bao
}
</script>
</body>
</html>
```

Bài 3. Kiểm tra giá trị nhập vào từ input

```
<!DOCTYPE html>
<html>
<head>
<script>
function validateForm()
{
    var kq;
    kq=true;
    var x=document.forms["myForm"]["fname"].value;
    if (x==null || x=="")
    {
        alert("First name must be filled out");
        kq=false;
    }
    var y=document.forms["myForm"]["fage"].value;
    if (isNaN(y)){
        alert("Age must be number");
        kq=false;
    }
    return kq;
}
</script>
```



```

</head>
<body>
<form name="myForm" action="demo_form.asp" method="post">
First name: <input type="text" name="fname">
<br>
Age:
<input type="text" name="fage">
<input type="button" value="Submit" onclick="return
validateForm()" ">
</form>
</body>
</html>

```

- Bài 4. Sinh viên tạo một webpage sử dụng javascript để thực hiện yêu cầu :  
***nhập vào 03 hệ số  $a, b, c$ , dùng javascript để hiển thị kết quả biện luận nghiệm của phương trình  $ax^2+bx+c=0$ .***
- Bài 5. Sinh viên tạo một form nhập ngày tháng năm trong một webpage, dùng javascript để kiểm tra tính hợp lệ của 03 giá trị ngày tháng năm.
- Bài 6. Sinh viên sử dụng các đối tượng : document, element, attr tạo mainlayout (bài tập cuối chương 1)

## Chương IV CSS - Cascading Style Sheets

### IV.1. Định nghĩa CSS:

CSS - Cascading Style Sheets – là ngôn ngữ thiết kế dùng để đơn giản hoá việc xử lý trình bày nội dung trên WebPages. Nó cung cấp các công cụ có khả năng trình bày các thành phần nội dung trong một trang WEB một cách hiệu quả. CSS được dùng với mục đích :

- Thay đổi màu sắc nền, ảnh nền,...
- Font chữ trình bày
- Diện tích trình bày
- Phong cách trình bày thẻ nội dung
- Các hiệu ứng trình bày khác.

Các lợi ích khi sử dụng CSS :

- Tiết kiệm thời gian thiết kế
- Tốc độ trình bày nội dung nhanh
- Dễ dàng cho việc bảo trì,
- Dễ dàng thay đổi hình thức trình bày
- Tạo sự đồng nhất về hình thức trong một website.
- Được hỗ trợ bởi nhiều trình duyệt và thiết bị
- Đáp ứng được chuẩn trình bày về Web.

### IV.2. Cú pháp định nghĩa CSS

Một quy định CSS được hiểu như là đặt giá trị cho các thuộc tính của một loại thẻ nội dung của trang WEB, cú pháp định nghĩa CSS được xác định như sau

```
<style>
Selector {
attributeName1 : attributeValue1;
attributeName1 : attributeValue1;
attributeName1 : attributeValue1;
.....
}
</style>
```

Trong đó :

- Selector : xác định thành phần nội dung cần trình bày; ta có thể xác định thành phần này thông qua : **tagName**, **id**, **className**, **attributeName**. Ta có thể nhóm nhiều bộ lựa chọn lại trong một quy định ví dụ :
  - Selector1,selector 2, selector 3{ .....}

- Một quy định gồm 02 phần : tên thuộc tính và giá trị thuộc tính.

Một số ví dụ về CSS :

```
<style>
p{
    color:blue;
}
#myId{
    color:red;
}
.myClass{
    color:yellow;
}
p.myClass {
    color:black;
}
</style>
<body>
<p>
In our CSS tutorial you will learn how to use CSS </p>
<p id="myId">
Paragraph with id = "myId".
</p>
<p class="myClass">
Paragraph with class name = "myClass".
</p>
<p class="myClass" id="myId" >
In our CSS tutorial you will learn how to use CSS </p>
<center class = "myClass">
center text with class name = "myClass"</center>
</body>
```

### IV.3. Các quy định CSS theo ngữ cảnh :

#### a. Descendent selector context :

Một selector có chứa nhiều phân cấp : phân cấp theo thứ tự từ trái sang phải, các selector phải cách nhau bởi 01 khoảng trắng

```
Selector1 selector2 selector3 {
.....
}
```

Ví dụ :

```
<style>
p cite {color:red;}
cite{color:blue;}
}
</style>
<body>
<p>
In our CSS tutorial you will learn how to use CSS <cite> to control the style and
layout of multiple Web pages all at once.</cite>
</p>
<cite> This CSS tutorial contains hundreds of CSS examples </cite>
</body>
```

### b. *Style for children*

Quy định cho các selector con trực tiếp của một selector

```
Selector1>selector2{
.....
}
```

Ví dụ :

```
<style>
p cite {color:red;}
cite{color:blue;}
p>em{
color:yellow;
}
}
</style>
<body>
<p>
In our CSS tutorial you will <em>learn how to use CSS </em><cite> to control the
style and layout of multiple Web pages all at once.</cite>
</p>
<cite> This CSS tutorial <em>contains hundreds</em> of CSS examples </cite>
</body>
```

### c. *Style for sibling*

Quy định cho các selector liền kề phía sau của một selector

```
Selector+selector{
.....
}
```

Ví dụ :

```
<style>
cite{color:red;}
cite+em{color:blue;}
em{
color:yellow;
}
</style>
<body>
<cite>Learn from over 150 examples!</cite><em> With our editor, you can edit the CSS,
and click on a button to view the result </em>
<br>
<em>Learn from over 150 examples!</em><cite> With our editor, you can edit the CSS,
and click on a button to view the result </cite>
</body>
```

### d. *Pseudo – ngôn ngữ giả*

- i. Link pseudo-class: xác định kiểu cho các trạng thái của liên kết. Các trạng thái có thể có của một liên kết:
  - Link :trạng thái khi liên kết chưa được kích hoạt
  - Visited :trạng thái khi liên kết đã được kích hoạt.
  - Hover :trạng thái khi người dùng di chuyển chuột qua
  - Active :trạng thái khi người dùng click chuột vào

Cú pháp : `a:pseudo-class{... ..}`

Ví dụ :

```
<style>
a:link{color:red;}
a:hover{color:blue;}
</style>
<a href = "www.ctu.edu.vn">Liên kết website đại học cần thơ</a>
```

- ii. Dynamic pseudo-class : xác định kiểu cho một thành phần html phụ thuộc vào cách người dùng tương tác với nó.

Các tương tác của người dùng : hover, active, focus.

Ví dụ :

- iii. Pseudo-element: xác định kiểu cho ký tự đầu tiên (first-letter) hay dòng đầu tiên (first-line ) trong một khối văn bản

Cú pháp: <selector>:<pseudo-elements>{property:value;}

Ví dụ :

```
<style>
p:first-letter{
color:blue;
}
p:first-line{
color:red;
}
</style>
<p>You can use the :first-line pseudo-element to add a special effect to the first line of a
text.</p>
```

Bảng liệt kê các Pseudo Classes/Elements

Selector	Example	Example description
:link	a:link	Selects all unvisited links
:visited	a:visited	Selects all visited links
:active	a:active	Selects the active link
:hover	a:hover	Selects links on mouse over
:focus	input:focus	Selects the input element which has focus
:first-letter	p:first-letter	Selects the first letter of every <p> element
:first-line	p:first-line	Selects the first line of every <p> element
:first-child	p:first-child	Selects every <p> elements that is the first child of its parent

:before	p:before	Insert content before every <p> element
:after	p:after	Insert content after every <p> element
:lang( <i>language</i> )	p:lang(it)	Selects every <p> element with a lang attribute value starting with "it"

#### IV.4. Các vị trí dùng để đặt CSS :

##### a. *Inline style:*

Các khai báo css được đặt trực tiếp vào một thẻ HTML, sử dụng thuộc tính style. Ví dụ :

```
<h1 style="color: red; font-family: Georgia; text-align: center; ">
Through the Looking-Glass
</h1>
```

##### b. *Internal style sheet (embedded):*

Các luật css được đặt trong thẻ <style> để định dạng cho trang hiện hành. Có thể đặt thẻ<style> trong thẻ<head> hay <body>. Nên đặt trong thẻhead, và trước các đoạn mã javascript.

##### c. *External style sheet:*

Các luật css được đặt trong các file riêng biệt (.css) có thể được truy cập bởi bất kỳ trang web nào bằng cách dùng thẻ <link> hay lệnh @import

Trong file html dùng thẻ

```
<link href="URL" type="text/css" media="all" />
<link href="default.css" type="text/css" media="all" />
```

Trong tập tin CSS dùng thẻ :

```
@import url(URL)
@import url(default.css);
```

#### **IV.5. Các thuộc tính thường sử dụng**

- a. Nhóm thuộc tính *Background*
- b. Nhóm thuộc tính *Font*
- c. Thuộc tính *Color*

#### **IV.6. Mô hình Box Model**

- a. Nhóm thuộc tính *border*
- b. Nhóm thuộc tính *padding*
- c. Nhóm thuộc tính *margin*
- d. Thuộc tính *display*
- e. Thuộc tính *float*

#### **IV.7. Bài tập :**

- Bài 1. Sinh viên thực hiện bài tập được cung cấp kèm theo tài liệu
- Bài 2. Sinh viên trình bày lại ý nghĩa tất cả các thuộc tính có trong bài thực tập kèm theo.
- Bài 3. Tạo main layout bằng mô hình Box model

## Chương V Ngôn ngữ PHP

### V.1. Chương trình Hello World

Một đoạn chương trình PHP phải được đặt trong thẻ : `<?PHP.....?>`.

Ví dụ :

```
<?PHP
echo "chao cac ban";
?>
```

Trong đó lệnh echo sẽ giúp chúng ta chuyển văn bản từ chương trình PHP thành văn bản của ngôn ngữ html.

### V.2. Khai báo biến trong PHP

Các biến trong PHP được khai báo ngay khi sử dụng, không khai báo kiểu dữ liệu; kiểu được xác định tại thời điểm gán trị. Việc sử dụng biến chưa khởi tạo sẽ gây lỗi. Mỗi biến sẽ có phạm vi hoạt động của mình thông qua các khối lệnh {}.

Khai báo `$ten_bien = gia_tri;`

Các quy cách đặt tên biến

- Có thể bao gồm các ký tự (A..Z,a..z), số (0..9), \_, \$
- Không được bắt đầu bằng số (0..9)
- Phân biệt chữ hoa, chữ thường.

### V.3. Kiểu dữ liệu trong PHP

Trong PHP hỗ trợ các kiểu dữ liệu như sau :

- Kiểu dữ liệu cơ bản :
  - String chuỗi ký tự
  - Boolean kiểu luận lý
  - Float, double kiểu số thực
  - Int kiểu số nguyên
- Kiểu dữ liệu kết hợp :
  - Array kiểu mảng
  - Object kiểu đối tượng

***Một biến dữ liệu có thể chứa bất kỳ kiểu dữ liệu nào. Tính chất của biến dữ liệu sẽ được xác định thông qua giá trị của chính nó.***

```
$a = "chao bạn"; kiểu chuỗi
$a=10; số nguyên
$a=10.5 số thực
```



*a. Kiểu dữ liệu chuỗi ký tự*

Giá trị văn bản được đặt trong cặp dấu **“Văn bản”** hoặc **‘văn bản’**. Bao gồm tất cả các ký tự hiện có.

Có 02 cách ghép chuỗi :

```
$a = "Chao ban";  
$b= "Thanh Binh";  
$c = $a.$b;  
$d= "Chao ban $b";
```

Sinh viên thử nghiệm kết quả hiện ra màn hình của các chuỗi ký tự sau :

```
$a="john";  
$b='david';  
$c= "hello $a";  
$d= "Chao '$a' !! ";  
$e= 'Bonjour "$b" ?';
```

Các hàm thường sử dụng trong chuỗi ký tự :

- strlen()
- trim()
- strtolower()
- strcmp()
- str\_replace()
- strtoupper()
- substr()
- strpos()

Sinh viên nghiên cứu các hàm tại địa chỉ :

[http://www.w3schools.com/PHP/PHP\\_ref\\_string.asp](http://www.w3schools.com/PHP/PHP_ref_string.asp)

*b. Kiểu dữ liệu mảng :*

Khởi tạo mảng :

<pre>\$a[0]=1; \$a[1]=2; \$a[2]=3;</pre>	<pre>\$a=array(1,2,3);</pre>	Khai báo mảng có 03 phần tử giá trị :1,2,3
<pre>\$a["ten"]="Binh"; \$a["holot"]="Nguyen van";</pre>	<pre>\$a=array("ten"=&gt;"Binh", "holot"=&gt;"Nguyen Van");</pre>	Khai báo mảng có 02 phần tử, chỉ số là "ten", "holot", giá trị "Binh", "Nguyen Van"
<pre>\$a[0][0]=1;\$a[0][1]=2; \$a[1][0]=3;\$a[1][1]=4; \$a[2][0]=5;\$a[2][1]=6;</pre>	<pre>\$a = array(     array(1,2),     array(3,4),     array(5,6) );</pre>	Ta có mảng 02 chiều 1     2 3     4 5     6
<pre>\$a["a"][0]=1;\$a["a"][1]=2; \$a["b"][0]=3;\$a["b"][1]=4;</pre>	<pre>\$a = array(     "a"&gt;array(1,2),     "b"&gt;array(3,4));</pre>	Chỉ số kết hợp \$a["a"][1]= ?

Cách truy xuất đến một phần tử trong mảng :

`$a[0]` hoặc `$a["ten"]` hoặc `a[i][j]`

Các hàm cơ bản trên mảng Array :

- `array_push(array, elements)` : *Thêm elements vào cuối mảng*
- `array_pop(array)` : *Lấy phần tử cuối ra khỏi mảng*
- `array_unshift(array, elements)` : *Thêm elements vào đầu mảng*
- `array_shift(array)` : *Lấy phần tử đầu ra khỏi mảng*
- `array_merge(array, array)` : *kết 2 mảng lại và trả ra mảng mới*
- `shuffle(array)` : *Sort random mảng*

#### V.4. Cấu trúc lập trình trong PHP

Trong lập trình PHP, chúng ta được phép sử dụng các cấu trúc lập trình :

a. *Tuần tự {.....} :*

Cấu trúc tuần tự được xác định trong một cặp dấu {...}, các câu lệnh cách nhau bởi dấu ;

b. *Cấu trúc rẽ nhánh if (điều kiện){...}else {...}*

```
<?PHP
$a = 4;
$b=10;
if ($a>$b){
    echo "$a > $b;
}
else {
    echo "$a <= $b";
}
?>
```

c. *Cấu trúc rẽ nhánh switch ... case*

```
<?PHP
switch ($color){
    case ("red") :
        echo "Mau do";
        break;
    case ("yellow") :
        echo "Mau vang";
        break;
    default :
        echo "Mau khong xac dinh";
}
?>
```

d. *Cấu trúc lặp*

*for (...;...;...){...}*

```
<?PHP
for ($i=1; $i<=5; $i++)
{
    echo "The number is " . $i . "<br>";
}
?>
```

*while(...){...}*

```
<?PHP
$i=1;
while($i<=5)
{
    echo "The number is " . $i . "<br>";
    $i++;
}
?>
```

### **do...while()**

```
$i=1;
do
{
    echo "The number is " . $i . "<br>";
    $i++;
}
while($i<=5)

?>
```

**foreach() {...}** – dùng để truy xuất từng phần tử của mảng.

```
<?PHP
$x=array("one","two","three");
foreach ($x as $value)
{
    echo $value . "<br>";
}
?>
```

## V.5. Lập trình hàm

Cú pháp lập trình hàm :

```
function <tên hàm>($tham so) {  
    câu lệnh tính toán  
    return $ketqua;  
}
```

Trong quá trình định nghĩa hàm, danh sách tham số không bắt buộc, kết quả trả về (**return \$ketqua;**) cũng không bắt buộc. Chúng ta có 03 trường hợp định nghĩa hàm :

- Hàm đơn giản – không có tham số, không có kết quả trả về
- Hàm có tham số
- Hàm có kết quả trả về.

## V.6. Lập trình hướng đối tượng:

**Tham khảo tại <http://PHP.net/manual/en/language.oop5.PHP>**

a. Khai báo lớp trong PHP :

Cú pháp khai báo :

```
class <tên class>{  
    <phạm vi> $tenbien;  
    public function __construct( /*...*/ ){  
        /* định nghĩa hàm xây dựng*/  
    }  
    <phạm vi> function tenham(){ .....}  
}
```

Ví dụ :

```
class Topic{  
    private $title;  
    private $content;  
    public function __construct($t,$c){  
        $this->title = $t;  
        $this->content=$c;  
    }  
    public function write(){  
        echo "Tieu de : ".$this->title. "<br>";  
        echo "Noi dung : ".$this->content. "<br>";  
    }  
}
```

Cách sử dụng :

```
<?PHP
$chude=new Topic("Hello", "Xin chao cac ban");
$chude->write();
?>
```

*b. Thừa kế*

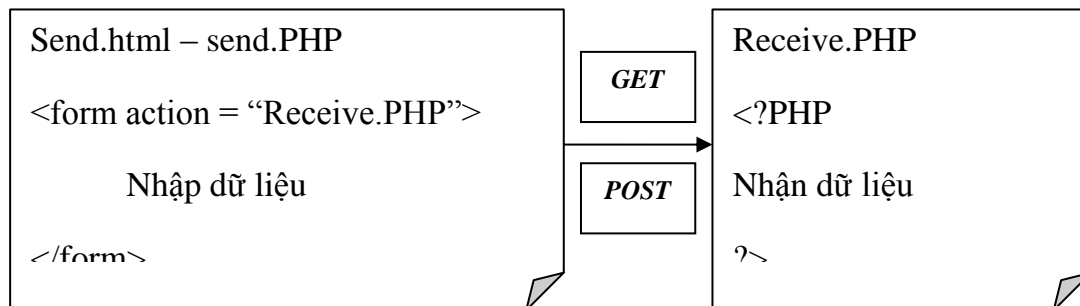
```
class <sub_Class> extends <classname>{
.....
}
```

### **V.7. Bài tập**

- Bài 1. Khai báo, gán giá trị của 03 hệ số \$a, \$b, \$c, viết chương trình giải phương trình  $ax^2+bx+c=0$ ;
- Bài 2. Khai báo \$n, \$m. Sử dụng vòng lặp hiển thị bảng trên webpage có \$n dòng, \$m cột.
- Bài 3. Khai báo lớp Topic, giống như ví dụ. Sinh viên thiết kế hàm main khai báo 01 mảng cho trước có 03 phần tử topic, hiển thị lần lượt từng topic.
- Bài 4. Khai báo một lớp đối tượng thể hiện các thành phần của một Webtemplate.

## Chương VI Gửi nhận dữ liệu trong PHP

### VI.1. Mô hình gửi nhận dữ liệu



### VI.2. Form nhập liệu

Chúng ta sử dụng thẻ form trong html để tiến hành nhập giá trị

```
<form>...</form>
```

Trong đó các thuộc tính quan trọng như sau :

- **Action= “URL”** : là thuộc tính xác định nơi nhận dữ liệu được nhập trong Form
- **Method = “post | get”** : là thuộc tính xác định phương thức gửi dữ liệu (chỉ sử dụng 01 trong 02 giá trị post hoặc get)
- **Target = “giá trị”** : có các giá trị như sau :
  - \_blank : trang web thứ hai sẽ mở trên một tab mới (trình duyệt mới)
  - \_self : trang web thứ hai sẽ mở tại vị trí đang xem
- **Name = “giá trị”** : tên của Form dùng để phân biệt trên một trang có nhiều Form.

### VI.3. Thẻ <input>

Để có thể nhập dữ liệu ta dùng thẻ <input>. Các thuộc tính cơ bản của thẻ:

- Name
- Value
- Type

a. Nhập văn bản :

Để nhập văn bản ta có thể sử dụng <input type = “text”> hoặc dùng thẻ <textarea>

- Thẻ <input type = “text”/> cho phép nhập dữ liệu trên 01 dòng
- Thẻ <textarea rows=“...” cols=“...”> .... </textarea> cho phép nhập văn bản trên nhiều dòng .

**b. Nhập password :**

Để ẩn dấu thông tin khi nhập password ta dùng thẻ `<input type= "password">`

**c. Nhập nút chọn – 01 lựa chọn**

thẻ `<input type= "radio">` sẽ hiển thị đối tượng cho phép chọn dạng radio button.

Ví dụ:

```
<div class = "quiz">
<div class = "question"> Nội dung câu hỏi </div>
<input type= "radio" name = "solution" value = "a"/>Phương án a<br/>
<input type= "radio" name = "solution" value = "b"/>Phương án b<br/>
<input type= "radio" name = "solution" value = "c"/>Phương án c<br/>
<input type= "radio" name = "solution" value = "d"/>Phương án d<br/>
</div>
```

**d. Nhập ô chọn – chọn nhiều lựa chọn :**

thẻ `<input type= "checkbox">` sẽ hiển thị đối tượng cho phép chọn dạng checkbox.

Ví dụ:

```
<div class = "quiz">
<div class = "question"> Nội dung câu hỏi </div>
<input type= "checkbox" name = "solution" value = "a"/>Phương án a<br/>
<input type= "checkbox" name = "solution" value = "b"/>Phương án b<br/>
<input type= "checkbox" name = "solution" value = "c"/>Phương án c<br/>
<input type= "checkbox" name = "solution" value = "d"/>Phương án d<br/>
</div>
```

**e. Submit**

`<input type = "submit" value = "gửi dữ liệu"/>`

Nút lệnh submit cho phép chúng ta yêu cầu gửi dữ liệu đến trang Web thứ hai.

**f. Cancel**

`<input type = "reset" value = "xóa dữ liệu"/>`

Nút lệnh cancel cho phép chúng ta yêu cầu xóa dữ liệu đã nhập.

**g. Dữ liệu ẩn :**

Sử dụng thẻ `<input type= "hidden">` giúp ta ẩn dấu một thông tin không hiện ra trên trang web gửi. Trong trang web nhận vẫn có thể truy xuất thông tin.

#### **VI.4. Thẻ <select>**

```
<select name="weekday">
<option value="Sunday">Sunday</option>
<option value="Monday" selected>Monday</option>
<option value="Tuesday">Tuesday</option>
<option value="Wednesday">Wednesday</option>
<option value="Thursday">Thursday</option>
<option value="Friday">Friday</option>
<option value="Saturday">Saturday</option>
</select>
```

**VI.5.      *Phương thức gửi dữ liệu GET/POST***

*a.    Phương thức GET*

*b.    Phương thức POST*

**VI.6.      *Trang gửi dữ liệu là trang nhận***



## Chương VII Phiên làm việc

### ***VII.1. Định nghĩa phiên làm việc***

Một phiên làm việc của người dùng được đánh dấu từ lúc đăng nhập đến khi đăng xuất. Một số trường hợp phiên làm việc được giới hạn theo thời gian làm việc của người dùng trong hệ thống tính từ khi đăng nhập.

Phiên làm việc được dùng để lưu trữ thông tin hoạt động của người dùng (các tác vụ, dữ liệu tạm thời). Ngoài ra phiên làm việc cũng là một phương thức truyền thông tin giữa các trang.

Trong một phiên làm việc cần lưu trữ tối thiểu các thông tin sau :

- Thông tin tài khoản
- Thời điểm đăng nhập : ngày giờ
- Thời gian làm việc : tính theo giây.
- Trang Web trước, trang hiện tại
- Dữ liệu tạm thời.

Mỗi một Webpage trước khi thực hiện cần kiểm tra trạng thái của phiên làm việc. Nếu phiên làm việc còn hiệu lực thì hiển thị, ngược lại thì chuyển đến trang login.

### ***VII.2. Lưu trữ phiên làm việc bằng liên kết***

Sử dụng phương thức truyền thông tin bằng phương thức ghép, lập trình viên có thể cung cấp thông tin tạm thời thông qua các tham số.

Ví dụ : tạo ra các liên kết lưu trữ thông tin về tên người dùng :

<http://localhost/topic.PHP?username=lcnghuyen>

trong tập tin **topic.PHP** nhận thông tin username thông qua câu lệnh :

```
$username=$_GET[ 'username' ] ;
```

### VII.3. Lưu trữ phiên làm việc bằng Cookies

#### a. Cookies :

Cookies là một tập tin được lưu trữ tại máy client (thông qua trình duyệt WEB Browser). Chúng ta sử dụng cookies với mục đích lưu trữ thông tin của phiên làm việc trên Website. Cookies có ưu điểm là thông tin sẽ được truy xuất nhanh, ngược lại có nhược điểm là không an toàn thông tin, người dùng có thể vô hiệu hoá cookies bằng cách cài đặt lại chương trình duyệt WEB.

Chúng ta có thể thay đổi thông tin vào cookies thông qua câu lệnh `setcookie()`.

```
setcookie(name, value, expire, path, domain);  
@setcookie(name, value, expire, path, domain);
```

Để có thể sử dụng câu lệnh `setcookie()`, ta cần phải thêm câu lệnh `<?php ob_start(); ?>` trước thẻ `<html></html>` và bất kỳ ký tự nào trong tập tin php.

#### Ví dụ :

```
<?php ob_start(); ?>  
<html>  
    <?php  
        setcookie("username", "ntbinh", time()+3600);  
    ?>  
</html>
```

#### b. Thêm một mẫu tin cookies :

```
<?php  
$n=60*60; //thời gian làm việc tính bằng giây  
$expire=time()+$n;  
//thời gian hết hạn : $expire được tính từ thời điểm hiện thời cộng  
thêm $n (giây)  
setcookie("username", "ntbinh", $expire);  
?>
```

#### c. Xóa thông tin của cookie

Một mẫu tin cookies chỉ bị xóa khi thời điểm hết hạn nhỏ hơn thời điểm hiện tại

```
<?php  
$n=60*60; //thời gian làm việc tính bằng giây  
$expire=time()-$n; // expire <time()  
setcookie("username", "ntbinh", $expire);  
?>
```

#### d. Xem thông tin của cookie :

```
<?php  
$n=60*60; $expire=time()+$n;  
setcookie("username", "ntbinh", $expire);  
  
$taikhoan = $_COOKIE["username"]; //ket qua ntbinh  
  
?>
```

Sinh viên tham khảo thêm

<http://php.net/manual/en/function.setcookie.php>

#### **VII.4. Lưu trữ phiên làm việc bằng Session**

Session là một đối tượng phần mềm được tạo ra trong quá trình đáp ứng các yêu cầu của trình duyệt WEB. Session được tạo ra và lưu trữ trong bộ nhớ RAM của máy chủ.

##### *a. Yêu cầu khởi tạo một phiên làm việc bằng lệnh :*

Để khởi tạo đối tượng session trong môi trường WEBServer ta sử dụng câu lệnh `session_start()`. Lưu ý “Câu lệnh `session_start()` phải được gọi trước cặp thẻ `<html/>`”.

##### *b. Thêm một biến giá trị vào session (parameter)*

Ta sử dụng biến môi trường : `$_SESSION`.

Thêm một biến có tên là username :

`$_SESSION['username'] = "admin";`

Ví dụ :

```
<?php
session_start();
$_SESSION['parameter'] = value;
?>
<html>
<?php
echo "Parameter = " . $_SESSION['parameter'];
?>
</html>
```

##### *c. Xóa một biến môi trường trong session :*

Sử dụng lệnh ***unset(...)***. Cần kiểm tra sự tồn tại của biến trước khi xóa (sử dụng câu lệnh ***isset(...)*** ).

```
if (isset($_SESSION['username'])) {
    unset($_SESSION['username']);
}
```

##### *d. Xóa đối tượng session trong bộ nhớ :*

Khi ta xóa đối tượng session trong bộ nhớ thì biến môi trường `$_SESSION` sẽ không có hiệu lực. Tất cả các thông tin trong session sẽ biến mất.

Câu lệnh xóa : `session_destroy();`

#### **VII.5. Lưu trữ phiên làm việc bằng CSDL.**

Việc lưu trữ phiên làm việc trên cơ sở dữ liệu sẽ giúp lưu thông tin không giới hạn thời gian, giới hạn dung lượng.

## **VII.6. Bài tập**

- Bài 1. Sinh viên trình bày ưu điểm, nhược điểm của từng loại lưu trữ phiên làm việc.
- Bài 2. Sinh viên lập trình lại các tập tin : login.php, welcome.php, topic.php, logout.php. Mỗi tập tin cần phải xem thông tin username, thêm thông tin của trang hiện hành.
- Bài 3. Sinh viên lập trình trang chọn hồ sơ, cho phép nhà tuyển dụng chọn hồ sơ ứng tuyển, có thể chọn nhiều lần.

## **Chương VIII Lập trình cơ sở dữ liệu**

### **VIII.1. Vòng đời sử dụng mối kết cơ sở dữ liệu**

Quá trình sử dụng cơ sở dữ liệu trong ngôn ngữ PHP được trải qua 03 giai đoạn :

- Tạo nối kết
- Thao tác trên dữ liệu: sử dụng ngôn ngữ SQL chuẩn để thao tác trên dữ liệu
  - ✓ Tạo bảng mới
  - ✓ Thêm mẫu tin
  - ✓ Xoá mẫu tin
  - ✓ Thay đổi giá trị mẫu tin
  - ✓ Tìm kiếm, thống kê, tổng hợp mẫu tin
- Đóng nối kết

Hệ quản trị khác nhau, có câu lệnh nối kết và thoát nối kết khác nhau.

#### *a. MySQL*

- Nối kết dữ liệu :

```
$connection=mysqli_connect("example.com","peter","abc123","my_db");
```

- Đóng kết nối :

```
mysqli_close($connection);
```

### *b. Microsoft SQL*

- Nối kết dữ liệu :

```
$connetion= mssql_connect($server, $user, $password);
```

- Đóng kết nối :

```
mssql_close($connetion);
```

### *c. Oracle*

- Nối kết dữ liệu :

```
$connection = oci_connect("phphol", "welcome", "///localhost/orcl");
```

- Đóng kết nối :

```
oci_close($connection);
```

## **VIII.2. Sử dụng câu lệnh truy xuất cơ sở dữ liệu MySQL**

### *a. Nối kết Cơ sở dữ liệu :*

```
<?php  
$address="localhost";  
$user = "root";  
$password = "root";  
$dbname = "vidu";
```

```
$connection=mysqli_connect("example.com","peter","abc123","my_db");
```

```
mysqli_close($connection);  
?>
```

### *b. Thực hiện câu lệnh SQL chuẩn :*

Trong PHP để có thể làm việc với cơ sở dữ liệu hiệu quả, ta cần nắm vững kiến thức cơ bản về cơ sở dữ liệu :

- Địa chỉ máy chủ
- Tài khoản đăng nhập cơ sở dữ liệu
- Tên cơ sở dữ liệu muốn truy xuất
- Tên bảng dữ liệu
- Tên các trường
- Nội dung câu lệnh SQL
  - ✓ CREATE DATABASE
  - ✓ CREATE TABLE
  - ✓ SELECT
  - ✓ INSERT
  - ✓ UPDATE
  - ✓ DELETE
  - ✓ DROP

Ví dụ :

```
<?php
$address="localhost";
$user = "root";
$password = "root";
$dbname = "vidu";
$connection=mysqli_connect("example.com","peter","abc123","my_db");

$SQLCommand = "select * from user";

$result = mysqli_query($connection,$SQLCommand);

mysqli_close($connection);
?>
```

*c. Xem kết quả của câu lệnh SELECT :*

Ta sử dụng câu lệnh `mysqli_fetch_array($result)` để có thể xem từng dòng của câu truy vấn kết quả.

```
<?php
$address="localhost";
$user = "root";
$password = "";
$dbname = "vidu";
$connection=mysqli_connect($address,$user,$password,$dbname);
$SQLCommand = "select * from user";
$result = mysqli_query($connection,$SQLCommand);

while($row = mysqli_fetch_array($result))
{
    echo $row['username'] . " " . $row['password'];
    echo "<br />";
}
mysqli_close($connection);
?>
```

*d. Đóng kết nối.*

Sau khi sử dụng xong nối kết đến một server cơ sở dữ liệu, ta cần tiến hành đóng nối kết, mục đích tiết kiệm tài nguyên máy tính, đảm bảo tính toàn vẹn của dữ liệu, tránh bị hư dữ liệu khi gặp các sự cố. Ta sử dụng câu lệnh :

```
mysqli_close($connection);
```

*e. Một số câu lệnh hỗ trợ khác :*

Địa chỉ tham khảo tại <http://www.php.net/manual/en/ref.mysql.php>

```
mysqli_connect_errno()
mysql_errno()
mysql_error()
mysql_fetch_array()
mysql_fetch_assoc()
mysql_fetch_field()
mysql_fetch_lengths()
mysql_fetch_object()
mysql_fetch_row()
```

## Chương IX Kết thúc môn

Sinh viên hoàn thành các công đoạn phát triển WEBSITE tìm kiếm việc làm trực tuyến

## Mục lục

Chương I	Đặt vấn đề.....	1
I.1.	Yêu cầu đề tài:.....	1
I.2.	Người dùng trong Website:.....	1
I.3.	Các đối tượng thông tin khác : .....	2
a.	Tin tức : .....	2
b.	Lĩnh vực tin tức.....	2
I.4.	SiteMap .....	2
I.5.	Wire frame – layout .....	3
Chương II	Ngôn ngữ HTML .....	5
II.1.	Thẻ nội dung - Tag.....	5
a.	Thuộc tính của thẻ nội dung.....	5
b.	Cấu trúc của một trang nội dung.....	5
II.2.	Các thẻ nội dung trình bày cấu trúc một tập tin html.....	6
a.	<html>.....	6
b.	<head> .....	6
c.	<body> .....	7
d.	<div>.....	7
e.	<span>.....	7
II.3.	Các thẻ trình bày văn bản :.....	7
a.	<p> .....	7
b.	Thẻ trình bày tiêu đề .....	7
c.	<pre> .....	7
d.	  .....	7
II.4.	Thẻ liên kết nội dung<a> : .....	8
a.	Liên kết nội : .....	8
b.	Liên kết ngoài : .....	9
II.5.	Thẻ trình bày về hình ảnh .....	9
II.6.	Trình bày nội dung dạng bảng <table> .....	10
a.	Các thuộc tính cơ bản của thẻ <table>.....	10



b. <tr>.....	10
c. <td> .....	11
d. <th> .....	11
e. Các thẻ khác : .....	12
f. Cấu trúc cơ bản của một bảng .....	12
g. Các hình thức mở rộng trình bày bảng : .....	12
II.7. Bài tập : .....	13
Chương III  Ngôn ngữ JavaScript.....	16
III.1. JavaScript .....	16
a. Thẻ Script : .....	16
b. Ngôn ngữ javascript :.....	16
c. Biến dữ liệu trong javascript : .....	17
d. Kiểu dữ liệu trong JavaScript : .....	17
e. Câu lệnh trong JavaScript : .....	19
f. Các cấu trúc câu lệnh : .....	19
g. Lập trình hàm trong JavaScript : .....	20
III.2. Mô hình DOM (Document Object Model). .....	21
III.3. Lập trình cho các sự kiện trong javascript : .....	23
III.4. Bài tập .....	24
Chương IV  CSS - Cascading Style Sheets.....	26
IV.1. Định nghĩa CSS: .....	26
IV.2. Cú pháp định nghĩa CSS.....	26
IV.3. Các quy định CSS theo ngữ cảnh : .....	27
a. Descendent selector context : .....	27
b. Style for children.....	28
c. Style for sibling .....	28
d. Pseudo – ngôn ngữ giả .....	28
IV.4. Các vị trí dùng để đặt CSS : .....	30
a. Inline style: .....	30
b. Internal style sheet (embedded): .....	30
c. External style sheet: .....	30

IV.5. Các thuộc tính thường sử dụng.....	31
a. Nhóm thuộc tính Background.....	31
b. Nhóm thuộc tính Font.....	31
c. Thuộc tính Color .....	31
IV.6. Mô hình Box Model .....	31
a. Nhóm thuộc tính border .....	31
b. Nhóm thuộc tính padding .....	31
c. Nhóm thuộc tính margin .....	31
d. Thuộc tính display .....	31
e. Thuộc tính float.....	31
IV.7. Bài tập :.....	31
Chương V Ngôn ngữ PHP .....	32
V.1. Chương trình Hello World.....	32
V.2. Khai báo biến trong PHP .....	32
V.3. Kiểu dữ liệu trong PHP.....	32
a. Kiểu dữ liệu chuỗi ký tự .....	33
b. Kiểu dữ liệu mảng : .....	33
V.4. Cấu trúc lập trình trong PHP.....	34
a. Vòng lặp {.....} : .....	34
b. Cấu trúc rẽ nhánh if (điều kiện){...} else {...} .....	34
c. Cấu trúc rẽ nhánh switch ... case.....	34
d. Cấu trúc lặp .....	34
V.5. Lập trình hàm.....	36
V.6. Lập trình hướng đối tượng:.....	36
Tham khảo tại <a href="http://PHP.net/manual/en/language.oop5.PHP">http://PHP.net/manual/en/language.oop5.PHP</a> .....	36
a. Khai báo lớp trong PHP : .....	36
b. Thừa kế .....	37
V.7. Bài tập .....	37
Chương VI Gởi nhận dữ liệu trong PHP .....	38
VI.1. Mô hình gởi nhận dữ liệu .....	38
VI.2. Form nhập liệu.....	38

VI.3. Thẻ <input> .....	38
a. Nhập văn bản :.....	38
b. Nhập password :.....	39
c. Nhập nút chọn – 01 lựa chọn .....	39
d. Nhập ô chọn –chọn nhiều lựa chọn : .....	39
e. Submit .....	39
f. Cancel.....	39
g. Dữ liệu ẩn : .....	39
VI.4. Thẻ <select>.....	39
VI.5. Phương thức gửi dữ liệu GET/POST.....	40
a. Phương thức GET .....	40
b. Phương thức POST .....	40
VI.6. Trang gửi dữ liệu là trang nhận.....	40
Chương VII Phiên làm việc .....	41
VII.1. Định nghĩa phiên làm việc.....	41
VII.2. Lưu trữ phiên làm việc bằng liên kết.....	41
VII.3. Lưu trữ phiên làm việc bằng Cookies .....	42
a. Cookies :.....	42
b. Thêm một mẫu tin cookies :.....	42
c. Xóa thông tin của cookie.....	42
d. Xem thông tin của cookie : .....	42
VII.4. Lưu trữ phiên làm việc bằng Session .....	43
a. Yêu cầu khởi tạo một phiên làm việc bằng lệnh :.....	43
b. Thêm một biến giá trị vào session (parameter).....	43
c. Xóa một biến môi trường trong session : .....	43
d. Xóa đối tượng session trong bộ nhớ : .....	43
VII.5. Lưu trữ phiên làm việc bằng CSDL. ....	43
VII.6. Bài tập.....	44
Chương VIII Lập trình cơ sở dữ liệu .....	44
VIII.1. Vòng đời sử dụng mối kết cơ sở dữ liệu.....	44
a. MySQL.....	44

b. Microsoft SQL .....	45
c. Oracle .....	45
VIII.2. Sử dụng câu lệnh truy xuất cơ sở dữ liệu MySQL .....	45
a. Nối kết Cơ sở dữ liệu : .....	45
b. Thực hiện câu lệnh SQL chuẩn : .....	45
c. Xem kết quả của câu lệnh SELECT : .....	46
d. Đóng kết nối.....	46
e. Một số câu lệnh hỗ trợ khác :.....	46
Chương IX Kết thúc môn .....	47