

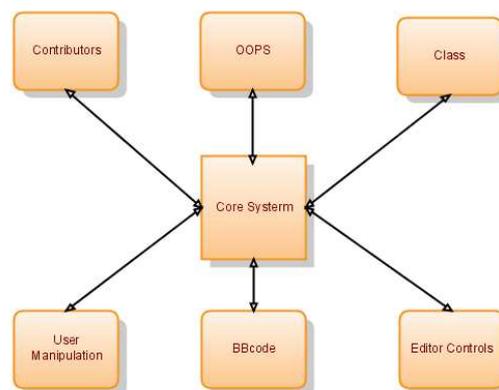
Trung Tâm Đào Tạo Công Nghệ Mạng Việt Chuyên

Tài Liệu Lưu Hành Nội Bộ

www.qhonline.info

GIÁO TRÌNH

PHP & MYSQL



Biên Soạn:

GV. Bùi Quốc Huy

Lời Nói Đầu

Quyển giáo trình này được tổng hợp từ những tài liệu và kinh nghiệm thực tiễn qua nhiều năm giảng dạy của giảng viên, và được soạn thảo cho phù hợp với nội dung môn học.

Khác với giáo trình mang tính thương mại, các từ ngữ, cách diễn giải mang xu thế thuật ngữ, giáo trình được xây dựng trên hình thức dễ đọc, dễ hiểu và gần gũi với thực tế, tuy nhiên điều đó không có nghĩa học viên có thể chỉ tham khảo giáo trình mà không đi học. Nếu các bạn muốn hiểu cặn kẽ kiến thức và ý nghĩa của giáo trình, bạn nên tham gia các khóa học PHP đầy đủ tại trung tâm đào tạo mạng Việt Chuyên.

Mặc dù đã kiểm tra độ xác thực, cũng như tính đúng đắn của giáo trình, nhưng khi biên soạn chắc chắn sẽ có ít nhiều lỗi liên quan đến từ ngữ, cú pháp. Rất mong nhận được nhiều sự đóng góp của các bạn.

Mọi ý kiến đóng góp của các bạn vui lòng truy cập vào website: www.qhonline.info (là website cá nhân của tác giả). Hoặc www.ddcntt.vn (là website của trung tâm).

Giảng viên

Bùi Quốc Huy

TP.Hồ Chí Minh. Ngày 15 Tháng 06 Năm 2006

Phần A: Cơ Bản Về HTML & Javascript

I – Các Thẻ HTML cơ bản

1- Cấu trúc chuẩn:

```
<HTML>
<HEAD><TITLE>Tiêu Đề Website</TITLE>
</HEAD>
<BODY>
Nội dung website được viết tại đây
</BODY>
</HTML>
```

+ Một cấu trúc HTML luôn bao gồm thẻ đóng và thẻ mở.

Ví dụ : ℒ thẻ Mở

 ℒ Thẻ Đóng - Được phân biệt bằng dấu / ngay trước thẻ

+ Các thẻ HTML hoàn toàn có thể viết hoa hoặc thường. Tuy nhiên nên theo 1 quy tắc chung (viết hoa thì viết hoa hết, ngược lại).

+ Một website càng ít thẻ thì sẽ càng chạy nhanh hơn.

2- Các thẻ cơ bản :

ü Tiêu Đề: <H>Nội Dung</H> (ví dụ: <H1>Welcome, admin</H1>).

Thẻ H bao gồm các thuộc tính từ (H1 à H6). Nhưng chỉ Sử dụng từ H1 à H3.

ü In Đậm: Nội Dung (ví dụ: Welcome, admin)

ü In Nghiêng: <I>Nội Dung</I> (ví dụ: <i>welcome, Admin</i>)

ü Gạch Dưới : <U>Nội Dung</U> (Ví dụ: <u>welcome, Admin</u>)

ü Xuống 1 Dòng :
 - Thẻ này không có thẻ đóng.

ü Xuống 2 Dòng : <P> - Có hoặc không thẻ đóng cũng được.

ü Định Dạng Chữ Bằng Thẻ

+ Kích thước chữ : Nội Dung

Ví dụ: Hello, How Are You ?

+ Kiểu Chữ : Nội Dung

Ví dụ: Hello, How Are You ?

+ Màu Chữ : Nội Dung

Ví dụ: ` Hello, How Are You ?`

ü Tạo Liên Kết : ` Nội Dung`

Ví Dụ: `Hello, Huy`

+ Mở 1 cửa sổ mới với liên kết: `TARGET=_BLANK`

Ví dụ: `Hello, Huy`

+ Mở liên kết đó ngay trên trang hiển thị: `TARGET=_SEFT`

Ví dụ: `Hello, Huy`

ü Định dạng thuộc tính : `ALIGN=LEFT, RIGHT, CENTER, JUSTIFY`

+ Sử dụng kết hợp với thẻ `<P>`

Ví dụ: `<p align=center>Nội dung à Văn bản sẽ ở giữa`

+ Sử dụng kết hợp với các thẻ khác như ``, `<table>`, `<div>`,....

ü Chèn hình ảnh vào website:

`<IMG SRC="URL Hình" BORDER="Thông Số" HSPACE="Thông Số"
VSPACE="Thông Số" ALIGN="Thuộc Tính" WIDTH="Thông Số"
HEIGHT="Thông Số">`

+ `` cú pháp chính để xuất ra hình ảnh.

+ `` cú pháp được sử dụng khi hình ảnh đi kèm với liên kết.

Ví dụ : ``

Từ câu lệnh trên sẽ cho ra hình def.gif. Tuy nhiên bao bọc hình ảnh đó sẽ có 1 đường viền.

Ví dụ: ``

Từ câu lệnh trên sẽ cho ra hình def.gif. Nhưng không có viền xung quanh

+ `Hspace` : khoảng cách từ lề phải của hình đến lề trái văn bản

+ `Vspace`: khoảng cách từ TOP của hình đến BOTTOM của văn bản

+ `Align` : Định dạng trái, phải, giữa của hình (canh lề).

+ `Width` : Chiều rộng của tấm hình.

+ `Height` : Chiều cao của tấm hình.

II- Tạo bảng trong HTML :

Để tạo 1 bảng chúng ta nên hình dung được dòng và cột. Và áp đặt 1 số quy luật, để việc thực thi dễ hơn so với yêu cầu đặt ra.

Một bảng bao gồm 2 thuộc tính là dòng và cột. Vậy ta có quy tắc để có 1 bảng dữ liệu. Ta đi từ bảng trước, sau đó kẻ dòng và chia cột trên dòng đó...Làm tuần tự cho đến hết yêu cầu việc tạo bảng.

1 Bảng bắt đầu và kết thúc sẽ có cú pháp : <table> </table>

1 Dòng bắt đầu và kết thúc sẽ có cú pháp : <tr> </tr>

1 Cột bắt đầu và kết thúc sẽ có cú pháp : <td> </td>

Ví dụ:

A	B	C	D
1	2	3	4

Như bảng ở trên ta sẽ xây dựng như sau :

+ Gồm 1 bảng.

+ Gồm 2 dòng

+ Gồm 4 cột trên 1 dòng

Vậy ta có cú pháp tạo bảng cơ bản như sau :

```
<Table>
  <tr>
    <td>A</td>
    <td>B</td>
    <td>C</td>
    <td>D</td>
  </tr>
  <tr>
    <td>1</td>
    <td>2</td>
    <td>3</td>
    <td>4</td>
  </tr>
</table>
```

Các thuộc tính trên Bảng (Table).

- Width : Chiều rộng của bảng.

- Height: Chiều cao của bảng.
- Align : Các thuộc tính trái, phải, giữa theo chiều ngang của bảng
- Valign : Các thuộc tính trên, dưới, giữa theo chiều dọc của bảng.
- Background : Màu nền của bảng bằng 1 hình ảnh. (chỉ có tác dụng trên cột và bảng).
- Bgcolor : Màu nền bảng bằng những màu theo thông số (chỉ có tác dụng trên cột và bảng). Ví dụ: white, blue, green, red,.....
- Cellpadding : Khoảng cách từ chân chữ đến cạnh dưới.
- Cellspacing : Khoảng cách của các cạnh.
- Rowspan : Gộp theo dòng

Ví dụ :

- Colspan : Gộp theo cột

Ví dụ:

* Các thuộc tính của table phía trên đều có thể áp dụng đối với cột (td). Nhưng không thể áp dụng đối với dòng.

Bài Tập Áp Dụng: Thiết kế 1 bảng bằng HTML như hình bên dưới :

Thứ 2	Thứ 3	Thứ 4	Thứ 5	Thứ 6	Thứ 7
Kinh tế Chính Trị Phòng 301			Toán A1	Anh Văn	PHP&MYSQL
			ASP.NET	CCNA	Vật Lý A1

Yêu cầu :

- 1 – Viết trên HTML, không dùng bất kỳ tool hỗ trợ nào.
- 2 – Thời gian trong 15 phút

III- Tạo Form nhập liệu trong HTML :

Form nhập liệu được dùng để lấy dữ liệu từ phía người truy cập, khách hàng,...Thông qua hệ thống website. Cú pháp cơ bản của Form cũng có thể bắt đầu và kết thúc :

Cú pháp chính :

```
<form method="POST" action="link" name="Tên">
```

Nội dung của form

```
</form>
```

+ Method : Là phương thức truyền nhận dữ liệu. Gồm 2 phương thức là POST và GET.

+ Action : Là đường dẫn đến liên kết xử lý form.

+ Name : là tên của Form đó.

Cơ cấu của Form bao gồm những thuộc tính sau :

1- *Dạng Text* :_(Như nhập họ tên, địa chỉ, số Điện Thoại,...)

Cú pháp :

```
<input type="Thuộc tính" name="Tên" size="chiều dài" value="giá trị">
```

+ Type : Thuộc tính có hai loại là Text (khi người nhập sẽ thấy được nội dung đang nhập và Password (người nhập sẽ không thấy được nội dung đang nhập)

+ Name : tên của hộp nhập liệu (ví dụ: name, address, phone,...)

+ Size : Chiều dài của hộp nhập liệu.

+ Value: Giá trị muốn hiển thị (thường với hộp nhập liệu thì ít sử dụng).

2- *Dạng Văn bản – Textarea* : (như nhập thông tin cá nhân, ghi chú,...)

Cú pháp :

```
<textarea name="textarea" cols="45" rows="5" ></textarea>
```

+ Có thể bắt đầu và kết thúc.

+ Name: Tên của hộp nhập liệu.

+ Cols : Được xem là chiều rộng của hộp nhập liệu

+ rows: Được xem là chiều cao của hộp nhập liệu

3- *Dạng Lựa Chọn* : (như chọn lựa giới tính nam hoặc nữ)

Cú pháp :

```
<input type="radio" value="N" name="hot" checked>
```

+ Type : Radio là định dạng dành cho sự lựa chọn

+ Value : Giá trị mà chúng ta muốn truyền khi user lựa chọn

+ Name : Tên của định dạng lựa chọn

+ Thuộc tính "checked". Nếu add vào sẽ mặc định chọn lựa

Ví dụ:

```
<input type="radio" value="m" name="sex" checked> Male
```

```
<input type="radio" value="f" name="sex" > Female
```

Từ ví dụ trên ta thấy nếu trong 1 form chọn lựa giới tính. Thì thuộc tính "name" phải giống nhau. Ở đây. Thuộc tính male được lựa chọn mặc định.

4- *Dạng Thanh cuộn* : (như chọn quốc tịch Việt Nam,....)

Cú pháp:

```
<select size="thông số" name="tên">
```

```
<option value="giá trị" selected>Lựa chọn </option>
```

```
</select>
```

+ Size : là kích thước của hộp thoại mà bạn muốn. Thường là 1.

+ Name : là tên của thanh cuộn ấy.

+ Option: là những thuộc tính bạn muốn chọn

+ Thuộc tính "selected" Nếu thêm vào sẽ mặc định chọn lựa.

+ Giá trị : là thông tin bạn muốn truyền vào khi chúng ta chọn lựa.

Ví dụ:

```
<select size="1" name="country">
```

```
<option value="VN" selected>Việt Nam </option>
```

```
<option value="CN">Trung Quốc </option>
```

```
</select>
```

Ở ví dụ này chúng ta thấy. Việt Nam là quốc gia được lựa chọn.

5- *Dạng Chọn Nhiều* : (như bảng đánh dấu sở thích: music, sport, game,.....)

Cú Pháp:

```
<input type="checkbox" name="tên" value="giá trị">
```

+ Type= Checkbox là định dạng của hộp thoại

+ Tên : là tên của mục chọn lựa

+ Giá trị: Là thông tin cần truyền vào.

6- *Dạng Nút Nhấn* : (như chấp nhận hoặc làm lại,.....)

Cú pháp :

```
<input type="Lựa chọn" value="Giá trị" name="Tên">
```

Type có hai loại chính là : submit và reset

+ Submit cho phép gửi dữ liệu lên trình duyệt để xử lý.

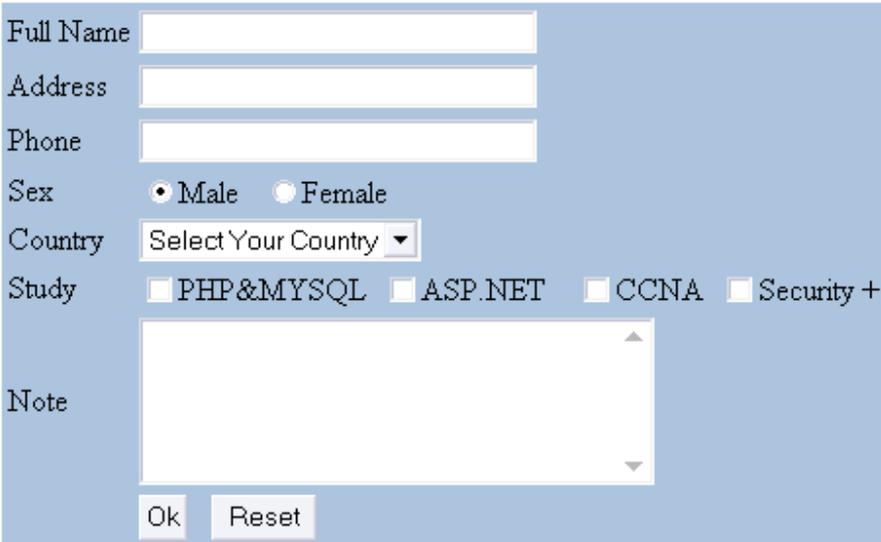
+ Reset : xóa toàn bộ dữ liệu từ các form trên.

Giá trị: là tên hiển thị trên nút

Tên : là tên của nút nhấn.

Bài tập Áp Dụng :

Viết lệnh hiển thị 1 form như bên dưới :



The image shows a web form titled "Enter Your Information" with a blue background. The form contains the following fields and controls:

- Full Name:
- Address:
- Phone:
- Sex: Male Female
- Country:
- Study: PHP&MYSQL ASP.NET CCNA Security +
- Note:
- Buttons:

IV- Các thẻ nâng cao:

1- Tao chữ chạy :

Cú pháp cơ bản : <Marquee> Nội Dung </Marquee>

Mô tả : Văn bản nằm giữa hai thẻ này sẽ di chuyển tùy theo các giá trị thuộc tính được thiết lập. Một số thuộc tính của Marquee là:

Direction: Có giá trị bằng Left hoặc Right là hướng chuyển động của văn bản.

ScrollDelay: Giá trị nguyên dương này là số mili giây văn bản ngừng trước khi chuyển động tiếp. Giá trị càng nhỏ, chuyển động càng nhanh.

Behavior: Loại chuyển động của text. Scroll, Slide, Alternate

Align With text: Đoạn văn bản được canh chỉnh trong vùng diện tích của marquee. Có 3 giá trị: Top, Middle, Bottom.

Loop: Giá trị nguyên thiết lập số chu kỳ chuyển động của đoạn văn bản.

Ví dụ :

```
<Marquee align="middle" direction="right" scrolldelay="60">
```

Chào mừng bạn đã đến website của chúng tôi

```
</Marquee>
```

2 – Sử dụng Frame:

Bằng cách dùng frame (khung), ta có thể ngắt trang web thành các vùng riêng biệt (của số) và thể hiện các tập tin HTML khác nhau trong mỗi vùng. Với cách này, mỗi vùng có thể chuyển sang trang web khác trong khi các vùng khác vẫn không thay đổi.

Cú pháp cơ bản :

```
<iframe src="URL" Name="Tên" Frameborder="Thông số" height="Thông số"  
width="thông số" scrolling="Thông số" allowtransparency> </iframe>
```

SRC : URL của trang muốn hiển thị.

Name : Tên của frame

Frameborder : Đường viền của frame (thường là 0 hoặc 1)

Height : chiều cao của frame

Width : Chiều rộng của frame.

Scrolling : Hiển thị thanh trượt (có 3 thuộc tính là "No", "Yes", "Auto").

Allowtransparency: Cho phép background của frame trong suốt.

Ví dụ :

```
<iframe src="test_page.html" Name="test_frame" Frameborder="0" height="300" width="300" scrolling="No" allowtransparency></iframe>
```

3- Tạo danh sách:

HTML có một tập các thẻ cho phép tạo ra các danh sách dạng phân cấp. Có hai dạng danh sách: không có thứ tự và có thứ tự. Các danh sách không có thứ tự là danh sách dạng nút, trong khi đó danh sách có thứ tự thường là danh sách có đánh số. Ta cũng có thể kết hợp hai loại này để tạo ra danh sách phức tạp hơn.

a) Danh sách có thứ tự :

Các mục của một danh sách có thứ tự được hiển thị với các con số thay vì các nút. Sử dụng thẻ (Order List) để thiết lập một danh sách có thứ tự, như được thể hiện trong đoạn mã sau. Hình ảnh thể hiện cách mà trình duyệt hiển thị đoạn mã này.

Cú Pháp : <OL type="[a] [i] [I]">

Ví dụ :

```
<OL>
  <LI>This is the first level-1 item in the list.
  <LI>This is the second level-1 item in the list.
  <OL>
    <LI>This is the first level-2 item in the list.
    <LI>And this is the second level-2 item in the list.
  </OL>
  <LI>This is the third level-1 item in the list.
</OL>
```

b) Danh sách không có thứ tự :

Cú pháp : <UL type="[square] [circle] [disc]">

Ví dụ :

```
<UL Type="Disc">This is the first level header.
  <LI>This is the first level-1 item in the list.
  <LI>This is the second level-1 item in the list.
  <UL>
    <LI>This is the first level-2 item in the list.
    <LI>And this is the second level-2 item in the list.
  </UL>
  <LI>This is the third level-1 item in the list.
</UL>
```

4- Sử dụng thẻ <DIV>:

DIV Được xem như là các thẻ chứa (thẻ mang). Chúng rất hữu dụng trong việc phân chia các khối văn bản với phần xung quanh nó mà không làm ảnh hưởng đến định dạng của riêng nó. Các thẻ này thường được dùng với các định dạng style sheet CSS và đang được ưa chuộng trong thiết kế layout của những trang web phức tạp.

Ví dụ :

Here is <DIV STYLE="font:bold 14pt">some text</DIV> in a DIV.

V- CASCADING STYLE SHEET (CSS)

Định nghĩa về bảng kiểu xếp chồng (css) sử dụng để tách biệt format trang và nội dung của trang web. Với CSS, ta có thể biến đổi mọi thứ từ kích cỡ, kiểu, màu sắc của văn bản cho đến khoảng cách giữa các chữ cái và các dòng, đường biên và phần đệm xung quanh phần tử, vị trí chính xác của đối tượng trên trang.

1- Định nghĩa trực tiếp :

Bảng kiểu toàn cục (hoặc bảng kiểu được nhúng vào) được kèm theo như là một phần của hồ sơ HTML. Loại bảng kiểu này xác định biên giới của nó bằng thẻ mở <STYLE> và thẻ đóng </STYLE>, nó thường được đặt trong phần head của trang HTML.

Ví dụ:

```
<HTML>
<HEAD>
<STYLE>
  H1 {font-size: 16pt; font-weight: bold; color: red}
  H2 {font-style: italic; font-size: 24pt; color: green}
  DIV {font-weight: bold; font-style: italic}
</STYLE>
</HEAD>
<BODY>
<DIV>This text is in a DIV.</DIV>
<H1>This text is in an H1.</H1>
<H2>This text is in an H2.</H2>
</BODY>
</HTML>
```

Hiển Thị :

This text is in a SPAN.
This text is in an H1.
This text is in an H2.

Mô tả thuộc tính :

Color: màu chữ

Font-size : Kích thước của chữ

Font-style : Định dạng chữ (italic : Nghiêng, blod: in đậm, underline: gạch dưới).

Font-family: Kiểu chữ

Background-color : màu nền

Text –decoration: Định dạng link.

+ None : không định dạng gì cả

+ Underline: gạch dưới liên kết.

+ Overline: gạch trên liên kết

2- Định nghĩa gián tiếp:

Khả năng định dạng được đề cập trong những phần trước không chỉ dừng lại ở mức độ của một trang web đơn mà còn có thể kiểm soát kiểu trên nhiều trang web hay có thể trên toàn bộ website bằng cách sử dụng bảng kiểu liên kết. Một bảng kiểu liên kết (hay bảng kiểu bên ngoài) chỉ đơn giản là một tập tin văn bản, bao gồm những định nghĩa kiểu. Tập tin này được lưu với phần mở rộng là CSS. Tập này (nằm ngoài tập tin HTML) có thể được tham chiếu hoặc liên kết bởi văn bản HTML bằng cách sử dụng thẻ <LINK> của trang HTML.

Ví dụ: Đầu tiên, sử dụng notepad để soạn thảo tập tin có nội dung như sau:

```
H1 {font-size: 16pt; font-weight: bold; color:red}
```

```
H2 {font-style: italic; font-size:24pt ; color:green}
```

```
DIV {font-weight: bold; font-style: italic}
```

Lưu tập tin với tên bất kỳ có phần mở rộng là CSS (ví dụ: style.css). Cũng cần chú ý rằng, bảng kiểu liên kết này không kèm theo các thành phần <Style> và </Style> (tức trong file CSS không có các thẻ này).

Bây giờ tập tin tạo một file HTML chứa một liên kết đến bảng kiểu bên ngoài đã định nghĩa.

Ví dụ:

```
<HTML>
<HEAD>
<LINK HREF="styles.css" REL="STYLESHEET" TYPE="text/css">
</HEAD>
<BODY>
<DIV>This text is in a DIV.</DIV>
<H1>This text is in an H1.</H1>
<H2>This text is in an H2.</H2>
<H2 STYLE="font-size: 36pt">This is modified H2 text.</H2>
</BODY>
</HTML>
```

Trong ví dụ này tập tin sử dụng thẻ <LINK> trong phần đầu của hồ sơ HTML. Thẻ này chứa thuộc tính HREF xác định vị trí của tập tin đích. Thuộc tính REL sẽ định rõ rằng liên kết này là một tham chiếu đến một bảng kiểu và thuộc tính TYPE định rõ loại bảng kiểu. Thời điểm hiện tại thì text/css là loại bảng kiểu được hỗ trợ phổ biến và duy nhất. Bảng kiểu liên kết có ảnh hưởng tới hồ sơ HTML giống như bảng kiểu toàn cục (bảng kiểu nhúng). Mọi phần tử trong hồ sơ HTML mà thẻ của chúng được định nghĩa trong tập tin style.css sẽ được thể hiện bằng cách sử dụng kiểu đã được xác định.

Vì không nằm trong bất kỳ tập tin HTML đặc biệt nào, nên bảng kiểu liên kết ngoài có thể được liên kết với số lượng hồ sơ không giới hạn. Ví dụ, tập tin có thể tạo bảng kiểu cho toàn bộ Website và liên kết từng trang với bảng kiểu đó. Khả năng này rất hữu ích khi tập tin muốn duy trì cách thể hiện, cũng như cách cảm nhận nhất quán trên toàn bộ website. Nó cũng rất tiện lợi khi muốn thay đổi cách thể hiện tổng thể của Website.

3- Định Nghĩa Trên 1 Lớp:

Là một biến định nghĩa format của các thuộc tính trong các thẻ nhưng mọi thẻ đều có thể dùng được.

Cú pháp : `.tênlớp {các thuộc tính}`.

Ví dụ:

```
<HTML>
<HEAD>
<STYLE>
  .large
  {
    font-size: 24pt;
    font-style : bold;
    font-family : tahoma;
    color: green;
    Background-color: blue;
  }
  .small {font-size: 8pt}
</STYLE>
</HEAD>
<BODY>
<H2 CLASS="small">The small class on an H2 element.</H2>
<DIV CLASS="large">The large class on a DIV element.</DIV><BR>
<P CLASS="small">The small class on a P element.</P>
<B CLASS="large">The large class on a B element.</B><BR>
</BODY>
</HTML>
```

Phần B: PHP & MYSQL

Bài 1: Tổng Quan Về PHP

I - Giới thiệu về PHP.

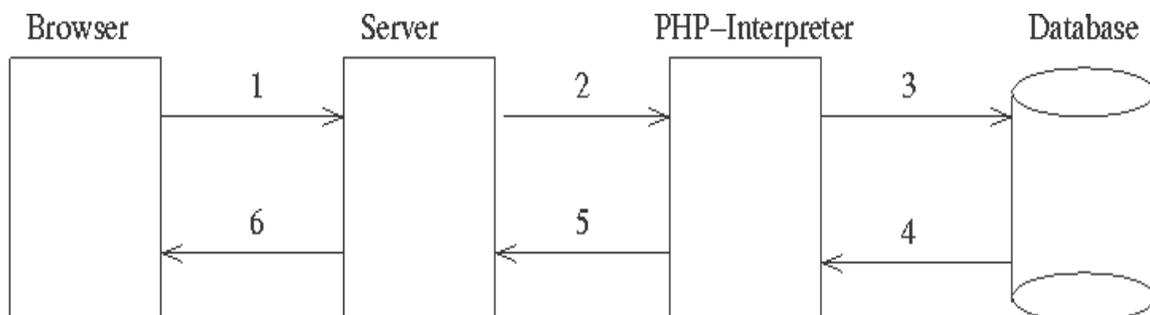
PHP (Hypertext Preprocessor) là ngôn ngữ script trên server được thiết kế để dễ dàng xây dựng các trang Web động. Mã PHP có thể thực thi trên Webserver để tạo ra mã HTML và xuất ra trình duyệt web theo yêu cầu của người sử dụng.

Ngôn ngữ PHP ra đời năm 1994 Rasmus Lerdorf sau đó được phát triển bởi nhiều người trải qua nhiều phiên bản. Phiên bản hiện tại là PHP 5 đã được công bố 7/2004.

Có nhiều lý do khiến cho việc sử dụng ngôn ngữ này chiếm ưu thế xin nêu ra đây một số lý do cơ bản :

- Mã nguồn mở (open source code)
- Miễn phí, download dễ dàng từ Internet.
- Ngôn ngữ rất dễ học, dễ viết.
- Mã nguồn không phải sửa lại nhiều khi viết chạy cho các hệ điều hành từ Windows, Linux, Unix
- Rất đơn giản trong việc kết nối với nhiều nguồn DBMS, ví dụ như : MySQL, Microsoft SQL Server 2000, Oracle, PostgreSQL, Adabas, dBase, Empress, FilePro, Informix, InterBase, mSQL, Solid, Sybase, Velocis và nhiều hệ thống CSDL thuộc Hệ Điều Hành Unix (Unix dbm) cùng bất cứ DBMS nào có sự hỗ trợ cơ chế ODBC (Open Database Connectivity) ví dụ như DB2 của IBM.

Sơ đồ về lập trình server side của PHP.



- 1: Trình duyệt gửi yêu cầu tới trang PHP.
- 2: Web server gửi các yêu cầu đó tới trình thông dịch PHP.
- 3-4: Trình thông dịch PHP thực thi các đoạn mã PHP. Quá trình này có thể liên quan đến nhiều tài nguyên như filesystem, database...
- 5: Kết quả của quá trình thông dịch là các mã HTML được trả về cho Server..
- 6: Server gửi mã kết quả HTML về lại trình duyệt.

II- Kiến Thức Cơ bản

1- Cú pháp chính :

PHP cũng có thể bắt đầu và kết thúc giống với ngôn ngữ HTML. Chỉ khác, đối với PHP chúng ta có nhiều cách để thể hiện.

Cách 1 : Cú pháp chính:

```
<?php Mã lệnh PHP ?>
```

Cách 2: Cú pháp ngắn gọn

```
<? Mã lệnh PHP ?>
```

Cách 3: Cú pháp giống với ASP.

```
<% Mã lệnh PHP %>
```

Mặc dù có 3 cách thể hiện. Nhưng đối với 1 lập trình viên có kinh nghiệm thì việc sử dụng cách 1 vẫn là lựa chọn tối ưu.

Trong PHP để kết thúc 1 dòng lệnh chúng ta sử dụng dấu “;”

Để chú thích 1 đoạn dữ liệu nào đó trong PHP ta sử dụng dấu “//” cho từng dòng. Hoặc dùng cặp thẻ “/*.....*/” cho từng cụm mã lệnh.

Ví dụ:

```
<?
Echo “PHP is simple”; //day la vi du ve code PHP
/* Voi cu phap nay chung ta
Co the chu thich 1 cum ma lenh */
?>
```

2- Xuất dữ liệu ra trình duyệt

Để xuất dữ liệu ra trình duyệt chúng ta có những dòng cú pháp sau :

+ Echo “Thông tin”;

+ Printf “Thông tin”;

Thông tin bao gồm : biến, chuỗi, hoặc lệnh HTML

Ví dụ :

```
<?php
Echo “Hello word”;
Printf“<br><font color=red>Who Are You ?</font>”;
?>
```

Nếu giữa hai chuỗi muốn liên kết với nhau ta sử dụng dấu “.”

Ví dụ :

```
<?php
Echo “Hello”.”who are you ?”;
?>
```

3- Khái niệm biến, hằng, chuỗi và các kiểu dữ liệu.

a) Biến trong PHP.

Biến được xem là vùng nhớ dữ liệu tạm thời. Và giá trị có thể thay đổi được. Biến được bắt đầu bằng ký hiệu “\$”. Và theo sau chúng là 1 từ, 1 cụm từ nhưng phải viết liền hoặc có gạch dưới.

1 biến được xem là hợp lệ khi nó thỏa các yếu tố :

+ Tên của biến phải bắt đầu bằng dấu gạch dưới và theo sau là các ký tự, số hay dấu gạch dưới.

+ Tên của biến không được phép trùng với các từ khóa của PHP.

Trong PHP để sử dụng 1 biến chúng ta thường phải khai báo trước, tuy nhiên đối với các lập trình viên khi sử dụng họ thường xử lý cùng một lúc các công việc, nghĩa là vừa khai báo vừa gán dữ liệu cho biến.

Bản thân biến cũng có thể gán cho các kiểu dữ liệu khác. Và tùy theo ý định của người lập trình mong muốn trên chúng.

Một số ví dụ về biến :

```
$a= 100 // biến a ở đây có giá trị là 100.
$a= “PHP is easy” // Biến a ở đây có giá trị “PHP Is easy”.
Biena=123 //Có lỗi vì bắt đầu 1 biến phải có dấu “$”
$123a=”PHP” //Có lỗi vì phần tên bắt đầu của biến là dạng số.
```

b) Khái niệm về hằng trong PHP.

Nếu biến là cái có thể thay đổi được thì ngược lại hằng là cái chúng ta không thể thay đổi được. Hằng trong PHP được định nghĩa bởi hàm define theo cú pháp: define (string tên_hằng, giá_trị_hằng).

Cũng giống với biến hằng được xem là hợp lệ thì chúng phải đáp ứng 1 số yếu tố :

- + Hằng không có dấu "\$" ở trước tên.
- + Hằng có thể truy cập bất cứ vị trí nào trong mã lệnh
- + Hằng chỉ được phép gán giá trị duy nhất 1 lần.
- + Hằng thường viết bằng chữ in để phân biệt với biến

Ví dụ :

```
define ("C", "COMPANY");  
define ("YELLOW", "#ffff00");
```

c) Khái niệm về chuỗi:

Chuỗi là một nhóm các kỹ tự, số, khoảng trắng, dấu ngắt được đặt trong các dấu nháy.

Ví dụ:

'Huy'

"welcome to VietNam"

Để tạo 1 biến chuỗi, chúng ta phải gán giá trị chuỗi cho 1 biến hợp lệ.

Ví dụ:

```
$first_name= "Nguyen";
```

```
$last_name= 'Van A';
```

Để liên kết 1 chuỗi và 1 biến chúng ta thường sử dụng dấu "."

Ví dụ:

```
<?php  
$test="VietNam";  
echo "welcome to".$test;  
echo "<br><font color=red>welcome to".$test."</font><br>";  
?>
```

d) Kiểu dữ liệu trong PHP

Các kiểu dữ liệu khác nhau chiếm các lượng bộ nhớ khác nhau và có thể được xử lý theo cách khác nhau khi chúng được theo tác trong 1 script.

Trong PHP chúng ta có 6 kiểu dữ liệu chính như sau :

Kiểu Dữ Liệu	Ví dụ	Mô Tả
Integer	10	Một số nguyên
Double	5.208	Kiểu số thực
String	“How are you ?”	Một tập hợp các ký tự
Boolean	True or False	Giá trị true hoặc false
Object	Hướng đối tượng trong PHP	
Array	Mảng trong PHP, chứa các phần tử.	

Chúng ta có thể sử dụng hàm dựng sẵn `gettype()` của PHP4 để kiểm tra kiểu của bất kỳ biến.

Ví dụ:

```
<?php
$a= 5;
Echo gettype($a); // Integer.
$a="<br>";
Echo gettype($a); //String
?>
```

III- Tổng Kết:

Sau bài này các bạn đã có những khái niệm đầu tiên về PHP, các cú pháp, các kiểu dữ liệu, và cách làm việc với môi trường PHP như thế nào. Ở bài sau, chúng ta sẽ tiếp tục tiếp cận với các thuật toán và cú pháp PHP một cách rõ ràng và quen thuộc trong các ngôn ngữ lập trình.