

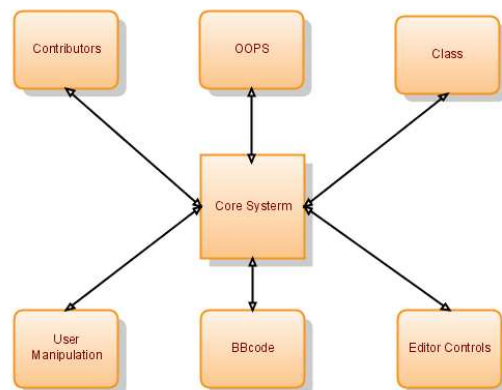
Trung Tâm Đào Tạo Công Nghệ Mạng Việt Chuyên

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 light blue background. The form contains the following elements:

- Full Name**: A text input field.
- Address**: A text input field.
- Phone**: A text input field.
- Sex**: Two radio buttons labeled "Male" and "Female".
- Country**: A dropdown menu with the text "Select Your Country".
- Study**: Four checkboxes labeled "PHP&MYSQL", "ASP.NET", "CCNA", and "Security +".
- Note**: A large text area for notes.
- Buttons**: Two buttons at the bottom labeled "Ok" and "Reset".

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, bold: 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.

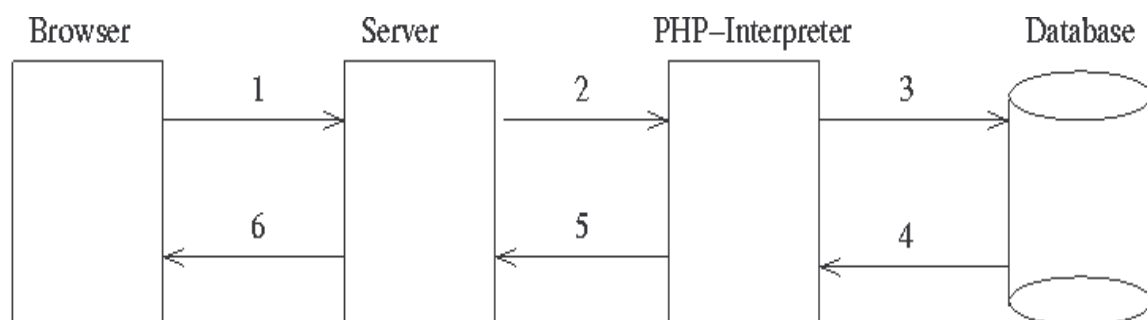
PHP (Hypertext Preprocessor) là ngôn ngữ script trên server được thiết kế để dễ dàng xây dựng các trang Web động. Mã PHP có thể thực thi trên Webserver để tạo ra mã HTML và xuất ra trình duyệt web theo yêu cầu của người sử dụng.

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 ký 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.

Bài 2 : Toán Tử Và Biểu Thức

I – Toán tử cơ bản.

1- Toán tử gán:

Chúng ta đã từng tiếp xúc với toán tử này bởi việc khởi tạo 1 biến. Nó gồm ký tự đơn =.

Toán tử gán lấy giá trị của toán hạng bên phải gán nó vào toán hạng bên trái.

Ví dụ: \$name = “Johnny Nguyen”;

2- Toán tử số học:

Ta có bảng các phép toán số học như sau:

Toán Tử	Lý giải	Ví dụ	Kết quả
+	Cộng hai số hạng	10+8	18
-	Trừ hai số hạng	10-8	2
*	Nhân hai số hạng	10*8	80
/	Chia hai số hạng	10/3	3.33333333
%	Trả về số dư	10%3	1

3- Toán tử so sánh:

Ta có các phép so sánh cơ bản như sau:

Phép Toán	Tên	Giải Thích	Ví Dụ
= =	Bằng	Hai số hạng bằng nhau	\$a= =5
!=	Không bằng	Hai số hạng không bằng nhau	\$a != 5
= = =	Đồng nhất	Hai số bằng nhau và cùng kiểu	\$a = = = 5
>	Lớn hơn	Vế trái lớn hơn vế phải	\$a > 5
>=	Lớn hơn hoặc bằng	Vế trái lớn hơn hoặc bằng vế phải	\$a >= 5
<	Nhỏ hơn	Vế trái nhỏ hơn vế phải	\$a < 5
<=	Nhỏ hơn hoặc bằng	Vế trái nhỏ hơn hoặc bằng vế phải	\$a <= 5

4- Toán tử logic

Toán tử logic là các tổ hợp các giá trị boolean. Ví dụ: toán tử or trở về true nếu toán tử trái hoặc toán tử phải là true.

True || false à true.

Ta có bảng các toán tử như sau:

Toán Tử	Tên	Trả về True Nếu	Ví dụ	Kết quả
	Or	Vế trái hoặc vế phải là True	True false	True
Or	Or	Vế trái hoặc vế phải là True	True false	True
Xor	Xor	Vế trái hoặc vế phải là True nhưng không phải cả hai	True True	False
&&	And	Vế trái và vế phải là true	True && false	False
And	And	Vế trái và vế phải là true	True && false	False
!	Not	không phải là true	!true	False

5- Phép gán kết hợp:

Khi tạo mã PHP, chúng ta sẽ thường nhận thấy cần phải tăng hoặc giảm lượng biến một số nguyên nào đó. Bạn sẽ thường thực hiện điều này khi chúng ta đếm 1 giá trị nào đó trong vòng lặp.

Phép Toán	Ví dụ	Lý Giải
++	\$a++	Bằng với \$a= \$a + 1
--	\$a--	Bằng với \$a= \$a - 1
+=	\$a+=\$b	Bằng với \$a= \$a + \$b
-=	\$a-=\$b	Bằng với \$a= \$a - \$b
=	\$a=\$b	Bằng với \$a= \$a * \$b
/=	\$a/=\$b	Bằng với \$a= \$a / \$b

II- Các biểu thức cơ bản:

1- Biểu thức điều kiện:

Cú pháp: If(Điều kiện)

```
{
hành động
}
```

Ví dụ:

```
<?php
$a=5;
```

```
$b=7;
If($a < $b)
{
    Echo” Bien A co gia tri nho hon bien B”;
}
Else
{
    Echo “Bien A co gia tri lon hon bien B”;
}
?>
```

2- Vòng lặp trong PHP:

a) While.....

Cú pháp:

While(điều kiện)

```
{
    Hành động – thực thi
}
```

Ví dụ:

```
<?php
$a=1;
While($a < 10)
{
    Echo “gia tri $a la”;
    $a++;
}
?>
```

Bài tập áp dụng:

- 1- Viết 1 trang web xuất ra giá trị từ 1->20. Với định dạng font màu đỏ, kiểu Tahoma, canh phải.
- 2- Viết 1 trang web có giá trị từ 1->20. Hãy xuất ra trình duyệt những số chẵn nằm trong khoảng 1->20 đó.

Đáp Án:Câu 1:

```
<?php
$a=1;
while($a <= 20)
{
    echo "<br>$a";
    $a++;
}
?>
```

Câu 2:

```
<?php
$a=1;
while($a <= 20)
{
    if($a%2==0)
        echo "<br>$a";
    $a++;
}
?>
```

b) Do.....while:Cú pháp:

Do

{

Hành động thực thi

}while(điều kiện)

Ví dụ:

```
<?php
$a=5;
do{
    echo "day la gia tri cua $a";
    $a++;
}while($a >6)
?>
```

c) For.....LoopCú pháp:

For(giá trị ; điều kiện ; biến tăng hoặc giảm)

{ Hành động }

Ví dụ :

```
<?php
$a=2;
for($i=1; $i<=10; $i++)
{
    echo"$a x $i =".$a*$i."<br>";
}
?>
```

Bài tập ví dụ:

Xây dựng 1 website thỏa yêu cầu xuất ra bảng cửu chương từ 2 à 10.

Đáp án:

```
<?php
for($i=2; $i <= 10; $i++)
{
    echo "Bang cuu chuong cua $i<br>";
    for($j=1; $j <=10; $j++)
    {
        echo "$i x $j =".$i*$j."<br>";
    }
    echo "<p>";
}
?>
```

d) Phát biểu switch:

Cú pháp:

Switch(biến)

```
{
    Case giá trị 1: Hành động; Break;
    .....
    Case giá trị N: Hành động; Break;
    Default: Hành động; Break;
}
```

Ví dụ:

```
<?php
```

```
$a=5;
switch($a)
{
    case 1: echo "day la gia tri $a"; break;
    case 2: echo "day la gia tri $a"; break;
    case 3: echo "day la gia tri $a"; break;
    case 4: echo "day la gia tri $a"; break;
    case 5: echo "<font color=red>day la gia tri $a</font>"; break;
    default: echo "Khong co gia tri phu hop"; break;
}
?>
```

III – Tổng kết:

Kết thúc bài học này, các bạn ít nhiều đã nắm được những thuộc tính cơ bản của các phép toán học trong PHP, ngoài ra chúng ta cũng từng bước hiểu được cú pháp của từng biểu thức.

Đối với các vòng lặp, chúng ta cần hiểu và vận dụng một cách linh hoạt và mềm dẻo, phân biệt được while và do...while. Sử dụng hàm switch trong trường hợp có quá nhiều giá trị if...else trả về.

Bài 3: Mảng Trong PHP (Array)

I- Những khái niệm cơ bản:

Chúng ta có thể tạo 1 mảng với các phần tử không đồng nhất về mặt dữ liệu. Đây là một ưu điểm của PHP so với các loại ngôn ngữ khác như C++ hay Java

1- Cách khởi tạo :

có nhiều cách

```
$countries[] = "cr";
```

```
$countries[] = "de";
```

```
$countries[] = "ca";
```

Lưu ý : giữa tên biến mảng và dấu [] không được cách khoảng trắng mà phải viết liền với nhau, nếu không khi lập trình CT sẽ không báo lỗi nhưng sẽ cho kết quả sai.

Chỉ số mặc định trong khởi tạo mảng nếu không chỉ định cụ thể sẽ là bắt đầu từ 0, 1, 2... hoặc chúng ta sẽ viết tường minh hơn như sau :

```
$countries[0] = "cr";
```

```
$countries[1] = "de";
```

```
$countries[2] = "ca";
```

Chúng ta cũng có thể khởi tạo mảng 1 cách không tuần tự như sau :

```
$countries[50] = "cr";
```

```
$countries[20] = "de";
```

```
$countries[10] = "ca";
```

```
$countries[] = "uk"; // sẽ mang chỉ số là 51
```

Hoặc :

```
$countries = array ("cr", "de", "us");
```

```
echo ($countries[2]); // sẽ in ra "us"
```

Nếu muốn lấy chỉ số đầu là 1 ta khởi tạo như sau :

```
$countries = array (1=> "cr", "de", "us");
```

```
echo ($countries[2]); // sẽ in ra "de"
```

Toán tử => có thể thay đổi vị trí

ví dụ :

```
$countries = array ("cr", 7=>"de", "us"); /* index 0 là "cr", index 7 là "de",
```

index 8 là “us” */

Để đếm số phần tử của mảng ta dùng hàm count, như ví dụ ở trên nếu ta ghi

```
echo (count ($countries)); // sẽ in ra 3
```

2- Đối với các mảng loại tuần tự:

Ta có thể in ra theo đoạn CT sau:

ví dụ :

```
$countries = array (“cr”, “de”, “us”);  
$num_elements = count ($countries);  
for ($idx = 0; $idx < $num_elements; $idx++) {  
echo (“$countries[$idx]” . “<BR>”);  
}
```

3- Đối với các mảng loại không tuần tự:

Cách 1 :

ví dụ :

```
$countries[50] = "cr";  
$countries[20] = "de";  
$countries[10] = "us";  
$countries[] = "uk";  
reset ($countries); /* Đưa pointer của mảng về vị trí khai báo đầu tiên là index 50 */  
while (list ($key, $value) = each ($countries)) {  
echo ("Element $key equals $value <BR>"); /* Lần lượt hiển thị theo thứ tự đã khai  
báo ban đầu là index 50, kế tiếp là index 20 rồi index 10, cuối cùng là index 51 */  
}
```

Cơ chế list () : Gán chỉ số index của mảng cho đối số thứ nhất của list là \$key, giá trị tương ứng với chỉ số index đó của mảng cho đối số thứ hai của cơ chế

list () là \$value

each () : Cứ mỗi lần được gọi sẽ di chuyển pointer của mảng đến thành phần kế tiếp của mảng, gán chỉ số index của mảng cho \$key và giá trị tương ứng với chỉ số index đó của mảng cho \$value của cơ chế list () .

Cách 2 :Ví dụ :

```
function println ($s) {
    echo (" $s <BR>");
}
$countries = array ("ca", "cr", "de", "us");
array_walk ($countries, println);
```

4- Mảng có chỉ số là kiểu chuỗi:Ví dụ :

```
$countries["ca"] = "Canada";
$countries["uk"] = "United Kingdom";
$countries["us"] = "United States";
$countries["cr"] = "Costa Rica";
reset ($countries);
while (list ($key, $value) = each ($countries)) {
    echo ("Element $key equals $value <BR>");
}
```

5- Mảng nhiều chiều:Ví dụ :

```
$continents = array ("europe"=>array ("de", "uk"), "north america"=>array ("ca",
"cr", "us"));
echo ($continents["europe"][1]); // in ra "uk"
echo ($continents["north america"][2]); // in ra "us"
sơ đồ biểu diễn mảng 2 chiều ở trên như sau :
```

\$continents["europe"]		\$continents["north america"]		
[0]	[1]	[0]	[1]	[2]
"de"	"uk"	"ca"	"cr"	"us"

Phép duyệt cho mảng ở trên được thực hiện như sau :

ví dụ :

```
while (list ($key1) = each ($continents)) {  
    echo ("$key1 : <BR>");  
    while (list ($key2, $value) = each ($continents["$key1"])) {  
        echo ("- $value<BR>");  
    }  
}
```

II- Các hàm sắp xếp trên mảng:

Quy luật chung trong sắp xếp mảng, chúng ưu tiên sắp số trước, kế đến là các ký tự đầu, rồi mới đến các ký tự chữ, sắp xếp theo alphabetical

1- Khi chỉ số (index) là số :

Hàm sort () :

Ví dụ :

```
$countries = array ("us", "uk", "ca", "cr", "de");  
sort ($countries);  
while (list ($key, $value) = each ($countries)) {  
    echo ("Element $key equals $value <BR>");  
}
```

sẽ hiển thị ra như sau :

Element 0 equals ca

Element 1 equals cr

Element 2 equals de

Element 3 equals uk

Element 4 equals us

* Ghi Chú: sắp xếp giá trị tăng theo alphabetical, chỉ số index bắt đầu từ 0.

2- Khi chỉ số (index) là chuỗi :

Hàm asort () :

Ví dụ :

```
$countries = array ("us"=>"United States", "uk"=>"United Kingdom",  
"ca"=>"Canada", "cr"=>"Costa Rica", "de"=>"Germany");  
asort ($countries);  
while (list ($key, $value) = each ($countries)) {  
echo ("Element $key equals $value <BR>");  
}
```

Kết quả :

Element ca equals Canada

Element cr equals Costa Rica

Element de equals Germany

Element uk equals United Kingdom

Element us equals United States

hàm asort () giữ nguyên chỉ số chuỗi, chỉ sắp xếp trên giá trị chuỗi của chỉ số chuỗi tương ứng.

Các hàm rsort () và arsort () cách thức sử dụng giống như hàm sort () và asort () nhưng sắp xếp theo hướng ngược lại với trật tự trước đó.

3- Sắp xếp tăng dần theo chỉ số (index) :

Hàm ksort () :

Ví dụ :

```
$countries = array ("e"=>"United States", "d"=>"United Kingdom",  
"c"=>"Canada", "b"=>"Costa Rica", "a"=>"Germany");  
ksort ($countries);  
while (list ($key, $value) = each ($countries)) {  
echo ("Element $key equals $value <BR>");  
}
```

Kết quả :

Element a equals Germany

Element b equals Costa Rica

Element c equals Canada

Element d equals United Kingdom

Element e equals United States

Lưu ý : không có hàm sắp xếp giảm dần theo chỉ số (index), muốn sắp xếp như thế ta sử dụng hàm ksort (), rồi sau đó sử dụng hàm arsort () để đạt được mục đích nêu trên.

4- Hàm sắp xếp theo quy định trước của người sử dụng:

a) Chỉ liên quan đến giá trị tương ứng với chỉ số (index) :

Hàm usort () : không giữ lại chỉ số (index) cũ

Ví dụ :

```
function by_length ($a, $b) {  
    $l_a = strlen ($a);  
    $l_b = strlen ($b);  
    if ($l_a == $l_b) return 0;  
    return ($l_a < $l_b) ? -1 : 1;  
}  
$countries = array ("e"=>"United States", "d"=>"United Kingdom",  
"c"=>"Canada", "b"=>"Costa Rica", "a"=>"Germany");  
usort ($countries, by_length);  
while (list ($key, $value) = each ($countries)) {  
    echo ("Element $key equals $value <BR>");  
}
```

Kết quả :

Element 0 equals Canada

Element 1 equals Germany

Element 2 equals Costa Rica

Element 3 equals United States

Element 4 equals United Kingdom

Việc sử dụng hàm usort () trong sắp xếp cho phép người sử dụng đưa vào quy định sắp xếp của riêng họ như trong ví dụ trên là về chiều dài của câu: câu ngắn ở trên, câu dài ở dưới, nhưng hàm usort () không giữ lại được chỉ số (index) nguyên thủy như lúc

ban đầu của mảng, muốn giữ nguyên chỉ số (index) như lúc ban đầu ta hãy sử dụng hàm uasort ().

b) Chỉ liên quan đến việc sắp xếp chỉ số (index) theo quy định trước bởi người sử dụng : ta sử dụng hàm uksort () (cách sử dụng thì tương tự như hàm vừa học ở trên)

c) Hàm tạo dãy số nguyên ngẫu nhiên trong phạm vi từ a -> b :
(điều kiện: a : integer, b : integer và a < b)

Ví dụ : Tạo dãy số nguyên ngẫu nhiên trong phạm vi từ 1 đến 10

```
$int = range (1, 10);  
srand (time ());  
shuffle ($int);  
while (list ( , $num) = each ($int)) {  
    echo (" $num <BR>");  
}
```

III – Tổng kết:

Qua bài học này, chúng ta tìm hiểu kỹ về các thành phần của mảng một chiều, nhiều chiều và các hàm cơ bản về xử lý dữ liệu. Hiểu rõ cấu trúc mảng, sự sắp xếp đồng bộ sẽ giúp ích cho các bạn rất nhiều trong những ứng dụng lớn.

Bài 4: Hàm Và Gọi Lại Hàm

I- Tạo Hàm Trong Php:

Ta nên tạo các hàm với lý do: Để tăng tính nhất quán, độ tin cậy, giảm thời gian viết mã và tăng tính mềm dẻo cho chương trình.

Trong php, hàm (có giá trị trả về) và thủ tục (không có giá trị trả về) đều được gọi chung là hàm. Định nghĩa một hàm được xây dựng bởi người sử dụng như sau:

```
function tênhàm([thamsố 1, tham số 2, ...)
{
    //Nội dung phần định nghĩa của hàm
}
```

Trong đó: function: là từ khoá.

- Tên hàm: Là một tên hợp lệ được đặt theo qui tắc như đặt tên biến và không trùng với các tên hàm đã xây dựng trong php.
- Một hàm có thể có một hay nhiều tham số hay không có tham số nào. Các tham số (nếu có) sẽ không có kèm theo kiểu dữ liệu mà kiểu của nó sẽ tùy thuộc vào giá trị nhận được khi gọi hàm.
- Phần nội dung hàm nằm trong cặp dấu bao khối { }.
- Hàm sẽ nhận giá trị trả về thông qua phát biểu return giá trị.
- return; sẽ thoát khỏi hàm và không trả về giá trị nào.
- return *biểu thức*; sẽ thoát khỏi chương trình và trả về giá trị *biểu thức*.
- Hàm có thể đặt bất cứ vị trí nào trong trang và chỉ được thi hành khi ta thực hiện gọi hàm.

Ví dụ: Viết hàm kiểm tra xem một số có phải là số nguyên tố hay không. Hàm sẽ nhận giá trị trả về: true hay false tương ứng với tham số nhận vào là số nguyên tố hay không.

```
<?
//Hàm có giá trị trả về: true/false.
function IsNguyenTo($num)
{
```

```
for($i=2; $i<$num; $i++)
    if ($num % $i==0) return false;
return true;    }
// Hàm không có giá trị trả về. Kết quả in ngay trong hàm
function IsNguyenTo2($num)
{
    $i=2;
    $nto = 1;
    while ($i<$num)
    {
        if ($num % $i==0)
        {
            $nto = 0;
            break; //Thoát khỏi while
        }
        $i++;
    }
    if ($nto) echo "l " . $num;
}
$ArrNum = array(2, 3, 5, 6, 9, 10, 23, 28, 29);
echo " Danh sach cac so nguyen to trong mang :<br>";
//Duyệt qua các phần tử trong mảng
for ($i=0; $i<9; $i++)
//Truyền tham số và gọi hàm.
if (IsNguyenTo($ArrNum[$i]))
    echo "l " . $ArrNum[$i];
echo "<br>";
for ($i=0; $i<9; $i++)
    IsNguyenTo2($ArrNum[$i]);
?>
```

II- Truyền Các Giá Trị Mặc Định:

Các hàm trong php có thể nhận các tham số mặc định. Các tham số mặc định khi định nghĩa hàm, chúng được gán giá trị. \$thamsố = giá trị. Các hàm có tham số mặc định nên chuyển các tham số mặc định sang phải.

```
function tênhàm($thamsố_1, $thamsố_2 = giátri_2, $thamsố_3=giátri_3)
```

Các tham số \$thamsố_2 và \$thamsố_3 là các giá trị mặc định. Khi gọi hàm, nếu ta chỉ truyền 1 giá trị cho hàm thì giá trị đó là của \$thamsố_1. Các tham số \$thamsố_2 và \$thamsố_3 nhận các giá trị là giátri_2 và giátri_3.

Ví dụ: Hàm function F(\$num1, \$num2 = 5) { ... } ta sẽ gọi F(3) ↔ F(3, 5).

III- Truyền Tham Biến Và Tham Trị:

Tất cả các ví dụ trước, tất cả việc truyền biến đều là tham trị. Để truyền tham biến (biến được truyền có thể thay đổi nội dung khi ra khỏi hàm), khi gọi hàm, ta đặt thêm dấu & trước biến cần truyền tham biến.

Ví dụ:

```
<?    function F($a, $b)
      {
        $a = 5;
        $b = 5;
        echo "<br>Trong ham F(), a = " . $a . " va b= " . $b;
      }
      $a = 8;
      $b = 8;
      echo  "<br>1. Ngoai ham F(), a = " . $a . " va b= " . $b;
//Gọi hàm tham trị.
      F($a, $b);
      echo " <br>2. Ngoai ham F(), a = " . $a . " va b= " . $b;
//Gọi hàm với biến $a là tham biến
      F(&$a, $b);
      echo " <br>3. Ngoai ham F() a = " . $a . " va b= " . $b;
?>
```

Kết quả:

- | |
|--|
| <ol style="list-style-type: none">1. Ngoai ham F(), a = 8 va b= 8
Trong ham F(), a = 5 va b= 52. Ngoai ham F(), a = 8 va b= 8
Trong ham F(), a = 5 va b= 53. Ngoai ham F() a = 5 va b= 8 |
|--|

IV- Tầm vực của biến:

Một biến trong một hàm chỉ có phạm vi trong hàm đó. Tuy nhiên, nếu trong một hàm, ta cần truy cập đến một biến mà không thông qua truyền tham số qua gọi hàm, ta phải sử dụng từ khoá global trước biến như minh hoạ trong ví dụ sau: Hàm local_Func có truy cập đến một biến ngoài nhưng không sử dụng từ khoá global. Ngược lại hàm global_Func cũng truy cập đến cùng một biến value, nhưng tác động ở hai hàm sẽ khác nhau.

Ví dụ:

```
<?
function localFunc()
{
    $value = 100;
    echo "<p>Trong local, value co gia tri la :>" + $value;
}
function globalFunc()
{
    global $value;
    echo "<p>Trong Global, value co gia tri la :>" + $value;
}
$value = 5;
echo "<p>1.Gtri bien value = >" . $value;
localFunc();
echo "<p>2.Gtri bien value = >" . $value;
globalFunc();
```

```
echo "<p>3. Gtri bien value = " . $value  
?>
```

Kết quả:

1. Gtri bien value = 5

Trong local, value có giá trị là: 100

2. Gtri bien value = 5

Global, value có giá trị là :5

3. Gtri bien value = 5

V- Sử dụng lại mã PHP:

Để sử dụng lại mã trong nhiều trang web (các script), ta thường viết các hàm, định nghĩa các hằng hay các đoạn mã và lưu vào một file riêng. Khi đó, một trang php nào cần sử dụng các đoạn mã đã viết, có thể chèn vào vị trí cần thiết. Để chèn một file có sẵn vào trang web, ta sử dụng một trong bốn câu lệnh sau: include (filename), require, include_once, require_once.

Ví dụ: <? Include ("functions.php") ?> sẽ chèn file functions.php từ cùng thư mục với trang đang viết vào ngay vị trí đặt lệnh include.

Ví dụ: có file includefile.php có nội dung

```
<?php  
echo 'You have included a file';  
?>
```

Tạo file mainfile.php có nội dung.

```
<?php  
include 'includefile.php';  
echo 'I should have included a file.';  
?>
```

File mainfile.php tương đương với file có nội dung:

```
<?php  
echo 'You have included a file';
```

```
echo 'I should have included a file.';
?>
```

Khác biệt giữa include, include_once, require, require_once

Include: Đọc file filename vào ngay tại vị trí câu lệnh. Trường trường hợp không tìm thấy file filename, Chương trình phát ra cảnh báo nhưng vẫn tiếp tục thực thi các câu lệnh sau include.

Require: Thực thi giống include, tuy nhiên khi không tìm thấy filename, chương trình sẽ không tiếp tục thực thi các câu lệnh tiếp theo.

Require_once và Include_once: Làm việc giống như require và include nhưng nó chỉ cho phép filename chỉ được chèn vào trang web một lần.

VI- Tổng kết:

Kết thúc bài học này, hẳn các bạn đã có những khái niệm cơ bản về hàm và cách sử dụng lại mã PHP. Trên ứng dụng thực tế việc dùng các hàm để triệu gọi đóng vai trò khá quan trọng, nó giúp mã nguồn của người sử dụng logic và dễ dàng chỉnh sửa hơn, việc ứng dụng nó trên website như thế nào cũng sẽ tùy biến hơn so với cách viết trực tiếp trên từng file.

Bài 5: Làm Việc Với Chuỗi

I- Những khái niệm cơ bản:

Trong PHP chuỗi được xác định theo 3 cách:

+ Dấu nháy đơn : VD: 'abc';

+ Dấu nháy kép : VD "abc";

+ Heredoc

Ví dụ:

```
<?php
$str = <<<EOD
Example of string
spanning multiple lines
using heredoc syntax.
EOD;
?>
```

Khuyến khích sử dụng kiểu nháy kép vì nó hỗ trợ nhiều ký tự đặc biệt hơn kiểu nháy đơn.

1- Phương thức nối chuỗi và ký tự đặc biệt trong PHP:

Để nối 2 chuỗi với nhau, PHP cho phép người sử dụng dùng cú pháp: dấu (.) để nối hai chuỗi với nhau thành 1 chuỗi dài hơn.

Ví dụ:

```
<?php
$a="abc";
Echo "12345 ".$a. "mnb";
?>
```

Để xử lý những ký tự lạ trong chuỗi như dấu “ hoặc \$, PHP cung cấp cho ta 1 phương thức xử lý hiệu quả. Đó là thêm dấu “\” sau mỗi giá trị.

Ví dụ:

```
<?php
Echo "gia tri \$a trong PHP that \"de hieu\" phai khong nao";
?>
```

Ta có bảng ký tự sau:

<code>\n</code>	Ký tự xuống dòng
<code>\r</code>	Trở về đầu dòng
<code>\t</code>	Ký tự tab
<code>\\</code>	backslash
<code>\\$</code>	dollar sign
<code>\"</code>	double-quote

II- Các hàm về chuỗi cơ bản:

1- Hàm *substr()* :

cú pháp : `string substr (string source, int begin, int length);`

trả về 1 chuỗi con trong chuỗi cần lấy, đối số thứ 3 là tùy chọn, vị trí đầu tiên của chuỗi được tính là vị trí 0.

ví dụ 1:

```
echo (substr ("christopher", 1)); // in ra "hristopher"
```

```
echo (substr ("christopher", -2)); // in ra "er"
```

//Nếu đối số thứ hai là số âm nó sẽ đếm từ cuối chuỗi ngược lại phía trước số ký tự bằng với giá trị tuyệt đối của số âm đó.

```
echo (substr ("christopher", -5, 3)); // in ra "oph"
```

```
echo (substr ("christopher", -3)); // in ra "her"
```

```
echo (substr ("christopher", 2, 3)); // in ra "ris"
```

```
echo (substr ("christopher", 2, -3)); // in ra "ristop"
```

```
echo (substr ("christopher", -6, -3)); // in ra "top"
```

```
echo (substr ("christopher", 7, -8)); // in ra "" chuỗi rỗng
```

2- Hàm *strlen()*:

cú pháp : `int strlen (string str);`

xác định chiều dài của chuỗi

Ví dụ :

```
echo (strlen ("Christopher")); // in ra số 11
```

3- Hàm thay đổi kiểu chữ:

+ Hàm `strtoupper ()` : đổi chữ thường thành hoa

ví dụ : `strtoupper ("toi");` // in ra "TOI"

+ Hàm `strtolower ()` : đổi chữ hoa thành thường

ví dụ : `strtolower ("TOI");` // in ra "toi"

+ Hàm `ucfirst ()` : đổi chữ in ký tự đầu tiên của câu

ví dụ : `ucfirst ("in house");` // in ra "In house"

+ Hàm `ucwords ()` : đổi chữ in ký tự đầu tiên của mỗi từ

ví dụ : `ucwords ("in house");` // in ra "In House"

4- Hàm so sánh chuỗi:

Có thể dùng toán tử so sánh `=` (bằng), `!=`, `<`, `<=`, `>`, `>=`, `===` để so sánh các chuỗi.

Ngoài ra, ta còn có một số phương thức so sánh chuỗi như sau:

Các hàm `strcmp()`, `strcasecmp()`, `strnatcmp()` sẽ so sánh các chuỗi theo phép so sánh tự điển.

Hàm `int strcmp(string S1, string S2)`: So sánh 2 chuỗi S1, S2 có phân biệt

hoa/thường. Hàm này trả về giá trị `>0` nếu `S1>S2`, `0` nếu `S1=S2` và giá trị `<0` nếu `S1<S2`. Nói chung hàm này có thể sử dụng giống toán tử so sánh trong biểu thức.

Hàm `int strcasecmp(string S1, string S2)`: Giống hàm `strcmp()` nhưng không phân biệt chữ hoa và chữ thường.

Ví dụ

```
<?php
    $var1 = "Hello";
    $var2 = "hello";
    echo "Length $var1:" . strlen($var1);

    if (!strcasecmp ($var1, $var2))
        echo "$var1 = $var2 in a case-insensitive "

    if (strcmp($var1, $var2) >0)
        echo "$var1 > $var2";
    if (strcmp($var1, $var2) <0)
        echo "$var1 < $var2";
    if (strcmp($var1, $var2) ==0)
```

```
echo "$var1 = $var2";  
?>
```

Kết quả hiển thị:

Length Hello:5

Hello = hello in a case-insensitive

Hello < hello

5- Hàm tìm kiếm và thay thế chuỗi:

+ Tìm kiếm chuỗi bên trong chuỗi : để tìm kiếm 1 chuỗi bên trong 1 chuỗi khác, bạn có thể sử dụng 1 trong 3 hàm `strstr()`, `strchr()`, `strrchr()` hay `stristr()`, các hàm trên sẽ trả về chuỗi `str1`, nếu tìm thấy chuỗi `str1` trong chuỗi `str2`, ngược lại hàm trả về giá trị `false`, nếu có nhiều chuỗi con `str1` trong `str2`, thì hàm này chỉ tìm kiếm chuỗi `str1` đầu tiên.

Hàm `strstr()` :

cú pháp : `string strstr (string str1, string str2);`

ví dụ :

```
<?php  
$str = "Your customer service is excellent";  
echo (strstr ($str, "is") . "<BR>"); // is excellent  
?>
```

+ Hàm tạo chuỗi con:Hàm **`string substr (string S, int start [, int length])`**: trả về chuỗi con được thành lập từ chuỗi `S` với các tham số `start` và `length`. Nếu `start >= 0`: sẽ trả về chuỗi bắt đầu từ vị trí `start` trong `S`.

Ví dụ:

```
<?php  
$rest = substr ("abcdef", 1); // returns "bcdef"  
$rest = substr ("abcdef", 1, 3); // returns "bcd"  
?>
```

Nếu `start < 0`: sẽ trả về chuỗi bắt đầu từ vị trí `start` của `S` tính từ cuối chuỗi.

```
<?php
```

```
$rest = substr ("abcdef", -1); // returns "f"  
$rest = substr ("abcdef", -2); // returns "ef"  
$rest = substr ("abcdef", -3, 1); // returns "d"  
?>
```

Nếu `length >= 0` (giá trị mặc định là 0), chuỗi trả về có chiều dài `length` ký tự tính từ vị trí `start`.

Nếu `length < 0`: Chuỗi trả về chuỗi từ `start` tới vị trí kết thúc cách cuối chuỗi `length` ký tự.

+ Hàm thay thế chuỗi con :

Hàm `str_replace ()` :

Cú pháp : `string str_replace (string str1, string str2, string str);`

Hàm thay thế chuỗi `S1` bằng chuỗi `S2` trong chuỗi `S`. Trong trường hợp có nhiều `S1` trong `S`, tất cả các `S1` đều được thay bởi `S2` (thay thế tất cả).

```
<?php  
$bodytag = str_replace ("%body%", "black", "<body text=%body%>");  
//$bodytag = "<body text=%body%>"  
?>
```

Hàm `substr_replace ()` :

Cú pháp : `string substr_replace (string str, string replacement, int start, int [length]);`

Hàm này sẽ thay thế chuỗi `str` bằng chuỗi `replacement`, phần nào của chuỗi `str` được thay thế sẽ tùy thuộc vào 2 tham số là `start` và `length`.

+ `start` : vị trí bắt đầu của chuỗi thay thế `replacement` trong chuỗi `str`

Nếu `start >= 0` : vị trí thay thế sẽ được tính từ đầu chuỗi trở đi, `start` tính bằng đơn vị ký tự (vị trí thứ nhất của chuỗi là 0).

Nếu `start < 0` : Nó sẽ được tính bắt đầu từ cuối chuỗi.

+`length` : là tham số tùy chọn, cho biết điểm dừng việc thay thế. Nếu không có đối số này thì việc thay thế sẽ là từ vị trí `start` đến cuối chuỗi.

Nếu `length = 0` : việc thay thế sẽ là chèn vào vị trí `start` và ký tự trước `start`

Nếu `length > 0` : cho biết số ký tự mà bạn muốn thay thế với `replacement`

Nếu `length < 0` : cho biết điểm kết thúc việc thay thế tính từ vị trí cuối chuỗi trở lại

Các hàm được giới thiệu phía trên là những hàm thông dụng và thường sử dụng. Để biết thêm thông tin nhiều hàm hơn, các bạn có thể truy cập vào website : php.net để lấy thông tin đầy đủ về các hàm trong chuỗi.

III- Tổng kết:

Ở bài này chúng ta đã làm tìm hiểu kỹ về kiểu dữ liệu chuỗi, những hàm phụ trợ quan trọng trong việc lập trình ứng dụng như tìm kiếm, thay thế, so sánh,...Đối với PHP chuỗi là một trong những mảng có phong phú về các hàm, để sử dụng tốt chúng bạn cần hiểu rõ nguyên tắc hoạt động của từng hàm, nắm vững cấu trúc cơ bản trong các hàm chuỗi để từ đó chúng ta dễ dàng vận dụng chúng trong các ứng dụng lớn.

Bài 6: Các Thao Tác Trên File

I- Cơ bản về File:

1- Mở file : Sử dụng hàm fopen ()

Cú pháp : int fopen (string filename, string mode)

Đối với hệ điều hành Unix hay Linux, thư mục gốc được định nghĩa bằng dấu /, nhưng đối với hệ điều hành Windows, thư mục gốc được định nghĩa trên một ổ đĩa cụ thể nào đó ví dụ như C:\, dấu phân cách giữa các thư mục và tập tin trong Unix hoặc Linux chúng ta sử dụng dấu /, còn trong windows chúng ta sử dụng \

ví dụ :

trong Linux :

```
$fl = fopen ("/website/customers.txt", "w");
```

trong Windows :

```
$fw = fopen ("C:\webdesign\sales.txt", "r");
```

a. *Các chế độ mở file* :

Mode	Diễn giải
r	Read only
r+	Read_Write
w	Write Only
w+	Write_Read. Nếu file này tồn tại, nội dung cũ sẽ bị xóa đi, còn nếu không tồn tại nó sẽ được tạo mới
a	Mở dưới dạng append dữ liệu, chỉ có write, nếu file tồn tại, sẽ ghi tiếp vào phần dưới của nội dung có sẵn, nếu file không tồn tại sẽ được tạo mới.
a+	Mở dưới dạng append dữ liệu (write và read), nếu file tồn tại, dữ liệu sẽ ghi tiếp vào phần bên dưới của nội dung cũ, nếu file không tồn tại sẽ được tạo mới
b	Mở dưới chế độ file binary

Trong trường hợp không biết file nằm ở vị trí nào trên hệ thống, bạn có thể khai báo thêm tham số là số 1 cho hàm fopen () như sau :

```
$fw = fopen ("customers.txt", "a", 1);
```

ví dụ :

```
<html>
<body>
  Open File <BR>
  <?
      $f = "C:\thuchanh\vanban.txt"
      $fw = fopen ($f, "a");
  ?>
</body>
</html>
```

b. Mở file thông qua ftp hay http :

Ví dụ :

```
$f = "ftp://www.qhonline.info/orders.txt";
$fw = fopen ($f, "a");
```

hoặc :

```
$f = "http://www.qhonline.info.com/orders.txt";
$fw = fopen ($f, "a");
```

c. Mở file và việc sử dụng toán tử @ để bắt lỗi

Xem đoạn chương trình sau đây :

```
<html>
  <body>
    Open file and Debug <BR>
    <?
        $f = "C:\thuchanh\vanban.txt";
        @$fw = fopen ($f, "a");
        if (!$fw) {
            echo ("File bị trục trặc, không mở được");
        }
    ?>
```

```
?>
</body>
</html>
```

2- Đóng file : hàm fclose ()

Cú pháp : int fclose (int fw)

\$fw là biến nhận giá trị trả về khi gọi hàm mở file \$fw = fopen (...);

3- Hiển thị file : hàm fpassthru ()

Cú pháp : int fpassthru (int fw)

Hàm này có thể hiển thị cả file ảnh

Ví dụ :

```
if (!$fw = fopen ("picture.gif", "rb")) {
    echo ("could not open file");
}
else {
    fpassthru ($fw);
}
```

4- Đọc file :

a) *Hàm fread ()*

Cú pháp : string fread (int fw, int length);

mỗi lần đọc length ký tự

Ví dụ :

```
if (!$fw = fopen ("vanban.txt", "r")) {
    echo ("could not open file");
}
else {
    $text = fread ($fw, 10); // Đọc 10 ký tự đầu của file
    fclose ($fw);
}
```


Nếu như file có số ký tự ít hơn length ký tự trong fread (), nó sẽ đọc đúng số ký tự của file.

b) Hàm fgetc ():

Cú pháp : string fgetc (int fw)

Đọc đúng 1 ký tự của file. Nếu file rỗng (không có ký tự) thì nó sẽ trả về một chuỗi rỗng (false)

c) Hàm fgets ()

Cú pháp : string fgets (int fw, int length);

sẽ dừng việc đọc file khi gặp ký tự new line hay cuối file.

d) Hàm fgetss ()

Cú pháp và hoạt động y như hàm fgets (), nhưng các ký tự nằm trong cặp dấu <> sẽ bị loại bỏ khi đọc, cho dù vậy nó vẫn được tính vào chiều dài của chuỗi đọc.

e) Hàm file ()

Cú pháp : array file (string filename);

mỗi dòng của file được đọc sẽ là 1 phần tử của mảng, dòng đầu tiên của file sẽ là phần tử 0

Ví dụ :

```
$arrtext = file ("vanban.txt");  
for ($i = 0; $i < count ($arrtext); $i++) {  
    echo ("<P> $arrtext[$i] </P>");  
}
```

5- Ghi lên file :

Hàm fwrite () và fputs ()

Cú pháp : int fwrite (int fw, string str, int length);

int fputs (int fw, string str, int length);

Đối số thứ ba nếu không đưa vào thì toàn bộ chuỗi sẽ được ghi, còn nếu đưa vào đối số thứ ba thì nó sẽ ghi vào file chuỗi str (đối số hai) có độ dài đúng bằng chiều dài length của đối số thứ ba.

II- Các hàm xử lý file:**1- Các hàm dịch chuyển hoặc quản lý con trỏ mẫu tin trong file :****a) Hàm rewind ()****Cú pháp :** int rewind (int fw);

Đưa con trỏ mẫu tin về đầu file

b) Hàm fseek ()**Cú pháp :** int fseek (int fw, int offset);

Di chuyển con trỏ mẫu tin đến một vị trí xác định trong file

offset : số byte hoặc số character tính từ đầu file

Ví dụ : fseek (\$fw, 1); // di chuyển đến ký tự thứ hai của file

c) Hàm ftell ()**Cú pháp :** int ftell (int fw);

cho biết vị trí hiện thời của con trỏ mẫu tin

Ví dụ :

fseek (\$fw, ftell (\$fw) + 20); từ vị trí hiện thời dịch chuyển thêm 20 ký tự

d) Hàm feof ()**Cú pháp :** int feof (int fw);

cho biết vị trí cuối file

Ví dụ :

```
if (!$fw = fopen ("vanban.txt", "r")) {  
    echo ("could not open file");  
}  
else {  
    while (!feof ($fw)) {  
        echo (fgetc ($fw));  
    }  
}
```

2- Copy, Delete, Rename file:**a) Hàm Copy ()****Cú pháp :** int copy (string source, string destination);

b) Hàm unlink ()

Cú pháp : `int unlink (string filename);`

delete file, file được delete phải đóng (không mở)

c) Hàm rename ()

Cú pháp : `int rename (string oldname, string newname);`

Đặt tên file lại

II- Tổng kết:

Việc sử dụng file một cách thành thạo sẽ giúp bạn dễ dàng vận hành các ứng dụng mang quy mô vừa và nhỏ như: website nhiều ngôn ngữ, bộ đếm,... và cả những công nghệ web mới như XML một cách dễ dàng,

Qua bài học này chúng ta cũng hiểu được nguyên lý hoạt động, trình tự xử lý 1 file dữ liệu khi chúng được triệu gọi trong tài liệu PHP.

Bài 7: Xử Lý Form Trong PHP

I- Tương tác với người sử dụng:

PHP cho phép người sử dụng tương tác với chúng thông qua form nhập liệu được viết thuần túy trên HTML. Tuy nhiên để làm được điều ấy PHP yêu cầu form phải đáp ứng 1 số quy định chung đặt ra.

Chúng ta cùng phân tích thẻ form trong HTML sau:

```
<form name="Tên form" action="link xử lý" method="Phương thức">
```

Chúng ta thấy rằng 1 form phải bao gồm:

Tên form để dễ dàng tách biệt với giá trị của chúng.

Action: hành động chuyển tiếp đến link xử lý.

Method: Là phương thức truyền bao gồm POST và GET.

Ví dụ:

```
<form action = "check.php" method = post name=reg>
Please type your name here : <BR>
<input type = text name = "username"><BR><BR>
<input type = submit value = "Submit Data" name="add">
</form>
```

Vậy làm cách nào để chúng ta lấy được giá trị vừa nhập liệu nào ?.

PHP cho phép ta lấy giá trị dựa vào 2 phương thức POST và GET.

Đối với POST ta có : `$_POST['Giá trị']`

Đối với GET ta có : `$_GET['Giá trị']`

Vậy với đoạn code trên có thể lấy được biến xử lý là : `$_POST['username']`;

II- Xử lý dữ liệu từ trang người nhập liệu.

1- Phương thức POST:

Phương thức này được sử dụng để lấy dữ liệu từ form nhập liệu. Và chuyển chúng lên trình chủ webserver.

2- Phương thức GET:

Phương thức này cũng được dùng để lấy dữ liệu từ form nhập liệu. Tuy nhiên nhiệm vụ chính của nó vẫn là lấy nội dung trang dữ liệu từ web server.

Bài tập ví dụ:

Xây dựng 1 trang HTML với nội dung gồm form nhập liệu họ và tên. Sau đó dùng 1 file php để xuất ra thông tin họ và tên mà người sử dụng vừa nhập liệu.

Đáp Án:

Tạo file userform.htm với nội dung sau:

```
<html>
<form action = "processform.php" method = post>
Please type your name here : <BR>
<input type = text name = "username"><BR><BR>
<input type = submit value = "Submit Data">
</form>
</html>
```

Tạo file processform.php để xuất ra dữ liệu

```
<?
echo "Welcome, " . $_POST['username'] . "!";
?>
```

Bài tập áp dụng :

Tạo 1 trang web với hộp thoại nhập liệu username và password. Nếu người sử dụng nhập thông tin username/password là admin/12345 thì xuất ra thông báo "welcome, admin" với kiểu chữ Tahoma, màu đỏ. Ngược lại nếu nhập sai thì xuất thông báo "Username hoặc password sai. Vui lòng nhập lại".

Đáp án:

Tạo trang login.html với nội dung sau:

```
<html>
<head>
<title>login page</title>
</head>
<body>
<form action="checklogin.php" method=POST>
Username <input type=text name="user" size=15>
<br>Password <input type=password name="pass" size=15>
<br><input type=submit name=submit value="Login">

</form>
</body>
</html>
```

Tiếp tục tạo trang checklogin.php với nội dung sau:

```
<?php
$username=$_POST['user'];
$password=$_POST['pass'];
if($username == "admin" && $password="12345")
{
    echo "<font color=red face=tahoma>Welcome to, ".$username."<font>";
}
else
{
    echo "<font color=red face=tahoma>Username hoặc password không chính xác, vui lòng đăng nhập lại<font>";
}
?>
```

III- Tổng Kết:

Kết thúc bài này các bạn đã nắm được kỹ thuật kiểm tra thông tin dựa trên PHP. Kỹ thuật xây dựng trang upload, kiểm tra tính hợp lệ của người sử dụng 1 cách dễ dàng bởi sự tùy biến trong các biểu thức của PHP.

Bài 8: Quản Lý Phiên Làm Việc

I- Tổng quan về cookie:

Cookie là 1 đoạn dữ liệu được ghi vào đĩa cứng hoặc bộ nhớ của máy người sử dụng. Nó được trình duyệt gửi ngược lên lại server mỗi khi browser tải 1 trang web từ server.

Những thông tin được lưu trữ trong cookie hoàn toàn phụ thuộc vào website trên server. Mỗi website có thể lưu trữ những thông tin khác nhau trong cookie, ví dụ thời điểm lần cuối ta ghé thăm website, đánh dấu ta đã login hay chưa, v.v...

Cookie được tạo ra bởi website và gửi tới browser, do vậy 2 website khác nhau (cho dù cùng host trên 1 server) sẽ có 2 cookie khác nhau gửi tới browser. Ngoài ra, mỗi browser quản lý và lưu trữ cookie theo cách riêng của mình, cho nên 2 browser cùng truy cập vào 1 website sẽ nhận được 2 cookie khác nhau.

1- Thiết lập cookie trong PHP:

Để thiết lập cookie ta sử dụng cú pháp:

Setcookie(“tên cookie”,”giá trị”)

Tên cookie là tên mà chúng ta đặt cho phiên làm việc.

Giá trị là thông số của tên cookie.

Ví dụ:

Setcookie(“username”,”admin”)

Như ví dụ trên ta thấy với tên là username và giá trị là admin.

Chú ý: Kịch bản cookie phải đặt trên mọi giá trị trả về bao gồm thẻ HTML và lệnh echo.

2- Sử dụng cookie trong PHP:

Để sử dụng lại cookie vừa thiết lập, chúng ta sử dụng cú pháp:

Cú pháp: \$_COOKIE[“tên cookies”]

Tên cookie là tên mà chúng ta thiết lập phía trên.

Ví dụ:

Tạo trang cookie.php với nội dung sau:

<?php

```
setcookie("name","Kenny Huy");
?>
<html>
<head>
<title>Test page 1</title></head>
<body>
<b><a href=cookie2.php>Click here</a></b>
</body>
</html>
```

Tiếp tục tạo trang cookie2.php với nội dung sau:

```
<html>
<head><title>Result Page</title></head>
<body>
<?php
echo "Ten cua ban la <b>".$_COOKIE['name']. "</b>";
?>
</body>
</html>
```

3- Định dạng các tham số khác trong cookie:

Cú pháp cơ bản:

Setcookie(“Tên Cookie”, “Giá trị”, “Thời Gian”, “Đường dẫn”, “Tên miền”, “An toàn”

Tên cookie là tên mà chúng ta đặt cho phiên làm việc.

Giá trị là thông số của tên cookie.

Thời gian là thời hạn dùng để thiết lập sự tồn tại của cookie. Được tính theo số giây so với mốc ngày 1-1-1970. Thường được tính bằng thời gian hiện tại cộng với 1 khoảng thời gian nào đó muốn sử dụng cookie

Đường dẫn là thư mục nào thì cookie có tác dụng.

Tên miền là tên trang web mà chúng ta thiết lập cookie .

An toàn là sự lựa chọn cookie có được gửi theo giao thức HTTPS hay không. Chọn 1 là có và 0 là kết nối thông thường(HTTP).

Ví dụ:

```
Setcookie(“user_id”, “1301”, time()+3600, “/”, “”, 0);
```

Tên cookie: User_id

Giá trị: 1301

Thời gian: Cookie tồn tại trong 1 giờ tính từ thời gian thiết lập.

Đường dẫn: trên mọi thư mục

Tên miền: Bỏ qua thuộc tính này

An toàn: Chấp nhận giao thức HTTP(thông thường).

4- Hủy Cookie:

Để hủy 1 cookie đã được tạo ta có thể dùng 1 trong 2 cách sau:

a) Cú pháp: setcookie("Tên cookie")

Gọi hàm setcookie với chỉ duy nhất tên cookie mà thôi

b) Dùng thời gian hết hạn cookie là thời điểm trong quá khứ.

Ví dụ: setcookie("name","Kenny Huy",time()-3600);

Ví dụ:

Tiếp tục tạo trang cookie3.php với nội dung sau:

```
<?php
setcookie("name","Kenny Huy",time()-360);
?>
<html>
<head>
<title>Test page 1</title></head>
<body>
<b><a href=cookie2.php>Click here</a></b>
</body>
</html>
```

Bài tập áp dụng:

Xây dựng website sử dụng tính năng cookie thỏa mãn những yêu cầu sau:

- 1- Tạo trang đăng nhập với username/password là : admin/vietchuyen. Nếu user đăng nhập thành công thì sẽ xuất ra dòng welcome, admin.
- 2- Tạo trang đăng xuất thoát khỏi phần quản trị

II- Tổng quan về SESSION

Một cách khác quản lý người sử dụng là session. Session được hiểu là khoảng thời gian người sử dụng giao tiếp với 1 ứng dụng. Một session được bắt đầu khi người sử dụng truy cập vào ứng dụng lần đầu tiên, và kết thúc khi người sử dụng thoát khỏi ứng dụng. Mỗi session sẽ có được cấp một định danh (ID) khác nhau và nội dung được lưu trong thư mục thiết lập trong file php.ini (tham số session.save_path).

Trong PHP, session được xem như một điều khiển phía server để lưu trữ và quản lý phiên làm việc. Để làm được điều này, khi một người sử dụng truy cập vào một website, webserver cung cấp cho người sử dụng một chuỗi giá trị là SessionID gọi là chuỗi định danh- ví dụ 9b02405434373d6d96c8db11ed819a60. Giá trị chuỗi này một giá trị ngẫu nhiên và nó được gửi tới máy người sử dụng thông qua cookie có tên PHPSESSID. Giá trị này tồn tại cho đến khi nào chu kỳ sống của session kết thúc hoặc người sử dụng di chuyển tới website khác. Chu kỳ sống của session do webserver qui định. Ta có thể điều chỉnh chu kỳ này khi cấu hình webserver. Bên phía server, một tập tin có tên tương ứng PHPSESSID cũng được tạo ra. Các biến session sẽ được lưu trong một tập tin văn bản này và ở tại vị trí được qui định trong file php.ini ở dòng session.save_path.

1- Thiết lập 1 session trong PHP:

Để thiết lập 1 session ta sử dụng cú pháp: session_start()

Đoạn code này phải được nằm trên các kịch bản HTML. Hoặc những lệnh echo, printf.

Để thiết lập 1 giá trị session, ngoài việc cho phép bắt đầu thực thi session. Chúng ta còn phải đăng ký 1 giá trị session. Để tiện cho việc gán giá trị cho session đó.

Ta có cú pháp sau: session_register("Name")

Ví dụ:

```
<?php
Session_start();
Session_register("username");
?>
```

2- Sử dụng giá trị của session:

Giống với cookie. Để sử dụng giá trị của session ta sử dụng mã lệnh sau:

Cú pháp: \$_SESSION["name"]

Với Name là tên mà chúng ta sử dụng hàm session_register("name") để khai báo.

Ví dụ:

Tạo trang session.php với nội dung sau:

```
<?php
session_start();
session_register("name");
$_SESSION["name"] = "Kenny Huy";
?>
<html>
<head>
<title>Test page 1</title></head>
<body>
<b><a href=session2.php>Click here</a></b>
</body>
</html>
```

Tạo trang session2.php với nội dung sau:

```
<?
session_start();
?>
<html>
<head><title>Result Page</title></head>
<body>
<?php
echo "Ten của bạn là <b>".$_SESSION["name"]."</b>";
?>
</body>
</html>
```

3- Hủy bỏ session trong PHP:

Để hủy bỏ giá trị của session ta có những cách sau:

Session_destroy() // Cho phép hủy bỏ session

Session_unset()// Cho phép hủy bỏ session (bao gồm tất cả những gì có liên quan).

Chú ý: Nên cẩn trọng khi sử dụng hàm unset.

Ví dụ

Tạo trang session3.php với nội dung sau:

```
<?php
session_start();
session_destroy();
?>
<html>
<head>
<title>Test page 1</title></head>
<body>
<b><a href=session2.php>Click here</a></b>
</body>
</html>
```

Bài tập áp dụng:

Xây dựng website sử dụng tính năng session thỏa mãn những yêu cầu sau:

- 1- Tạo trang đăng nhập với username/password là : admin/vietchuyen. Nếu user đăng nhập thành công thì sẽ xuất ra dòng welcome, admin.
- 2- Tạo trang đăng xuất thoát khỏi phần quản trị

III- Tổng Kết:

Sau bài học này chúng ta đã nắm được cách điều khiển phiên làm việc giữa cookie và session. Sử dụng chúng trong từng trường hợp cụ thể. Từ đó có thể áp dụng để viết những ứng dụng nhỏ như kiểm soát người đăng nhập, làm giỏ hàng online,.....

Bài 9: Tổng Quan Về MYSQL

I- Những khái niệm cơ bản:

Mysql là hệ quản trị dữ liệu miễn phí, được tích hợp sử dụng chung với apache, PHP. Chính yếu tố phát triển trong cộng đồng mã nguồn mở nên mysql đã qua rất nhiều sự hỗ trợ của những lập trình viên yêu thích mã nguồn mở.

1- Cách khởi động và sử dụng MYSQL.

Chúng ta sử dụng command như sau:

Mysql -hname -uuser -ppass

Để truy cập vào cơ sở dữ liệu.

Hoặc sử dụng bộ appserv để vào nhanh hơn theo đường dẫn sau:

Start/ Appserv/ Mysql command Line client

Sau đó nhập password mà chúng ta đã đặt vào.

2- Những định nghĩa cơ bản:

a) Định nghĩa cơ sở dữ liệu, bảng, cột:

Cơ sở dữ liệu: là tên của cơ sở dữ liệu chúng ta muốn sử dụng

Bảng: Là 1 bảng giá trị nằm trong cơ sở dữ liệu.

Cột là 1 giá trị nằm trong bảng. Dùng để lưu trữ các trường dữ liệu.

Thuộc tính

Ví dụ:

Bảng user	
User_id	1234
First_name	Nguyen
Last_name	Hai Anh
Username	Haianh123
Password	12345
Email	test@yahoo.com
Date	2007-12-12 16:25:30

Như vậy ta có thể hiểu như sau:

1 cơ sở dữ liệu có thể bao gồm nhiều bảng.

1 bảng có thể bao gồm nhiều cột

1 cột có thể có hoặc không có những thuộc tính.

b) Định nghĩa 1 số thuật ngữ:

NULL : Giá trị cho phép rỗng.

AUTO_INCREMENT : Cho phép giá trị tăng dần (tự động).

UNSIGNED : Phải là số nguyên dương

PRIMARY KEY : Cho phép nó là khóa chính trong bảng.

c) Loại dữ liệu trong Mysql:

Ở đây chúng ta chỉ giới thiệu 1 số loại thông dụng: 1 số dữ liệu khác có thể tham khảo trên trang chủ của mysql.

Kiểu dữ liệu	Mô Tả
Char	Định dạng text có chiều dài từ 0->255
Varchar	Định dạng text có chiều dài từ 0->255
Text	Định dạng text có chiều dài 0->65535
Longtext	Định dạng text có chiều dài 0->4294967215
INT	Định dạng số có chiều dài từ 0->4294967215
Float	Định dạng số thập phân có chiều dài nhỏ
Double	Định dạng số thập phân có chiều dài lớn
Date	Định dạng thời gian theo định dạng: YYYY-MM-DD
DateTime	Định dạng thời gian theo định dạng: YYYY-MM-DD HH:MM:SS

3- Những cú pháp cơ bản:

Cú pháp tạo 1 cơ sở dữ liệu:

CREATE DATABASE tên_cơ_sở_dữ_liệu

Cú pháp sử dụng cơ sở dữ liệu: Use tên_database;

Cú pháp thoát khỏi cơ sở dữ liệu: Exit

Cú pháp tạo 1 bảng trong cơ sở dữ liệu:

CREATE TABLE user (<tên_cột> <mô_tả>,...,<tên_cột_n>.....<mô_tả_n>)

Ví dụ:

```
mysql> create table user(user_id INT(15) UNSIGNED NOT NULL
AUTO_INCREMENT, user
```

```
ame VARCHAR(255) NOT NULL, password CHAR(50) NOT NULL, email  
VARCHAR(200) NOT N  
LL, PRIMARY KEY (user_id));  
Query OK, 0 rows affected (0.01 sec)
```

Hiển thị có bao nhiêu bảng: show tables;

Hiển thị có bao nhiêu cột trong bảng: show columns from table;

Thêm 1 cột vào bảng :

ALTER TABLE tên_bảng ADD <tên_cột> <thuộc_tính> AFTER <tên_cột>

Ví dụ:

```
mysql> alter table user add sex varchar(200) NOT NULL after email;  
Query OK, 2 rows affected (0.01 sec)  
Records: 2 Duplicates: 0 Warnings: 0
```

II- Những cú pháp sql cơ bản:

1- Thêm giá trị vào bảng:

Cú pháp: INSERT INTO Tên_bảng(tên_cột) VALUES(Giá_trị_tương_ứng);

Ví dụ:

```
mysql> insert into user(username,password,email,sex,home) values("Lanna","12345"  
,"lanna@yahoo.com","F","www.abc.com");  
Query OK, 1 row affected (0.00 sec)
```

2- Truy xuất dữ liệu:

Cú pháp: SELECT tên_cột FROM Tên_bảng;

Ví dụ:

```
mysql> select user_id,username from user;  
3 rows in set (0.00 sec)
```

4- Truy xuất dữ liệu với điều kiện:

Cú pháp: SELECT tên_cột FROM Tên_bảng WHERE điều_kiện;

Ví dụ:

```
mysql> select user_id,username from user where user_id=2;
```

Bài tập áp dụng:

Tạo cơ sở dữ liệu với thông tin sau:

ID	Fname	Lname	Email	Password	Sex	Phone
1	Nguyen	Van Hoa	vanhoa@yahoo.com	12345	F	0903234532
2	Tran	Hoa	Hoai2@yahoo.com	23456	M	97234512
3	Nguyen	Thi Thuy	Thuy2a@yahoo.com	24625	F	98374642
4	Hoang	Phi Hong	phihong@hotmail.com	57265	M	94673257
5	Le	Van Lam	vanlam@gmail.com	45625	M	93562535
6	Nguyen	Cam Nhung	cnhung@gmail.com	52672	F	93525135
7	Hoang	Phi Yen	pyen@gmail.com	63456	F	94754363

Thực hiện các công việc sau:

- 1- Tạo 1 bảng với các cột trên.
- 2- Nhập liệu thông tin theo như bảng trên.
- 3- Liệt kê tất cả thông tin của những ai có sex là female.
- 4- Liệt kê thông tin của người có Lname là “Van Hoa”.
- 5- Liệt kê tất cả thông tin của những ai có sex là male và có họ là “Nguyễn”.
- 6- Liệt kê fname,lname, email của những ai có số phone “0903234532”.
- 7- Dùng hàm mã hóa để bảo mật password của người dùng.

5- Truy cập dữ liệu và sắp xếp theo trình tự

Cú pháp:

```
SELECT tên_cột FROM Tên_bảng
WHERE điều kiện (có thể có where hoặc không)
ORDER BY Theo quy ước sắp xếp.
```

Trong đó quy ước sắp xếp bao gồm hai thông số là ASC (từ trên xuống dưới), DESC (từ dưới lên trên).

Ví dụ:


```
mysql> select user_id,username from user order by username ASC ;
```

Câu lệnh trên lấy ra thông tin user_id và username từ bảng user và chúng được sắp xếp theo username trình tự từ trên xuống dưới.

6- Truy cập dữ liệu có giới hạn :

Cú pháp:

```
SELECT tên_cột FROM Tên_bảng  
WHERE điều kiện (có thể có where hoặc không)  
LIMIT vị trí bắt đầu, số record muốn lấy ra
```

Ví dụ:

```
mysql> select user_id,username from user order by username ASC limit 0,10 ;
```

Câu lệnh trên lấy ra thông tin user_id và username từ bảng user. Chúng được sắp xếp theo username trình tự từ trên xuống dưới. Và lấy ra 10 record từ vị trí 0.

7- Cập nhật dữ liệu trong bảng:

Cú pháp:

Update tên_bảng set tên_cột=Giá trị mới

WHERE (điều kiện).

Nếu không có ràng buộc điều kiện, chúng sẽ cập nhật toàn bộ giá trị mới của các record trong bảng.

Ví dụ:

```
mysql> update user set email="admin@qhonline.info" where user_id=1 ;
```

8- Xóa dữ liệu trong bảng:

Cú pháp:

DELETE FROM tên_bảng WHERE (điều kiện).

Nếu không có ràng buộc điều kiện, chúng sẽ xóa toàn bộ giá trị của các record trong bảng.

Ví dụ

```
mysql> delete from user where user_id=1 ;
```

III- Mysql và PHP:1- Kết nối cơ sở dữ liệu:Cú pháp:

Mysql_connect("hostname","user","pass")

2- Lựa chọn cơ sở dữ liệu:Cú pháp:

Mysql_select_db("tên_CSDL")

Ví dụ:

```
$conn=mysql_connect("localhost","root","root") or die(" không thể kết nối");  
Mysql_select_db("demo");
```

3- Thực thi câu lệnh truy vấn:Cú pháp:

Mysql_query("Câu lệnh SQL")

4- Đếm số record trong bảng:Cú pháp:

Mysql_num_rows();

5- Lấy dữ liệu từ database:Cú pháp:

Mysql_fetch_array();

Ví dụ :

```
<?php  
$conn=mysql_connect("localhost","root","root") or die(" không thể kết nối");  
Mysql_select_db("demo");  
$sql="select * from user";  
$query=mysql_query($sql)  
if(mysql_num_rows($query) != 0)  
{
```

```
$row=mysql_fetch_array($query)
}
?>
```

Bài tập áp dụng:

Xuất thông tin người dùng từ cơ sở dữ liệu ở bài tập trên bằng PHP.

ID	Fname	Lname	Email	Password	Sex	Phone
1	Nguyen	Van Hoa	vanhoa@yahoo.com	12345	F	0903234532
2	Tran	Hoa	Hoal2@yahoo.com	23456	M	97234512
3	Nguyen	Thi Thuy	Thuy2a@yahoo.com	24625	F	98374642
4	Hoang	Phi Hong	phihong@hotmail.com	57265	M	94673257
5	Le	Van Lam	vanlam@gmail.com	45625	M	93562535
6	Nguyen	Cam Nhung	cnhung@gmail.com	52672	F	93525135
7	Hoang	Phi Yen	pyen@gmail.com	63456	F	94754363

IV- Tổng kết:

Ở bài này chúng ta đã làm việc với sql và các kỹ thuật kết nối với database của PHP. Các hàm PHP hỗ trợ mysql hiện nay có rất nhiều. Tuy nhiên về tính năng cũng như phương thức sử dụng thì chúng cũng giống nhau. Do vậy các bạn không nên lo lắng khi làm việc với những hàm mà chúng ta đang nghiên cứu.

Phần C: Project Training

Với những kiến thức đã học được, bạn hãy áp dụng để xây dựng hoàn chỉnh một ứng dụng web cụ thể và nộp chúng cho giảng viên của bạn.

Một số project tiêu biểu khuyến khích các học viên làm:

Đề tài 1: Xây dựng trang tin tức điện tử với các tính năng chuyên nghiệp như (chuyên mục tin, tin tức,...