

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



TÌM HIỂU VỀ PHP, MYSQL
ĐƠN VỊ: CÔNG TY CỔ PHẦN VCCORP

Giáo viên hướng dẫn: TS. Nguyễn Quang Hưng

Sinh viên : Nguyễn Quốc Đại - B21DCCN025

CHƯƠNG 1: TÌM HIỂU NGÔN NGỮ PHP VÀ MYSQL

1.1.Tìm hiểu ngôn ngữ PHP

1.1.1.Giới thiệu.

a) Lịch sử ra đời:

Năm 1995, Rasmus Lerdorf tạo ra PHP nhằm giải quyết việc viết lặp đi lặp lại cùng đoạn mã khi tạo các trang. Ban đầu tác giả chỉ có ý định tạo bộ phân giải đơn giản để thay thế các thẻ lệnh trong file HTML bằng các đoạn mã lệnh viết bằng C.

b) PHP là gì?

PHP là chữ viết tắt của “Personal Home Page” do Rasmus Lerdorf tạo ra năm 1994.Vì tính hữu dụng của nó và khả năng phát triển, PHP bắt đầu được sử dụng trong môi trường chuyên nghiệp và nó trở thành “PHP :Hypertext Prerocessor”

Thực chất PHP là ngôn ngữ kịch bản nhúng trong HTML, nói một cách đơn giản đó là một trang HTML có nhúng mà PHP,PHP có thể đặt rải rác trong HTML.

PHP là một ngôn ngữ lập trình được kết nối chặt chẽ với máy chủ, là một công nghệ phía máy chủ và không phụ thuộc vào môi trường. Đây là hai yếu tố rất quan trọng,thứ nhất khi nói công nghệ phía máy chủ tức là nói đến mọi thứ trong PHP đều xảy ra trên máy chủ, thứ hai , chính vì tính chất không phụ thuộc vào môi trường cho phép PHP chạy trên hầu hết các hệ điều hành Windows, Unix ...

Khi một trang Web muốn dùng được ngôn ngữ PHP thì phải đáp ứng được tất cả các quá trình xử lý thông tin trong trang Web đó,sau đó đưa ra kết quả ngôn ngữ HTML.

PHP được thiết kế để chỉ thực hiện điều gì đó sau khi một sự kiện xảy ra.

c) Tại sao cần dùng PHP?

PHP được sử dụng để phát triển website động vì nó tốt, nhanh và dễ dàng nghiên cứu hơn các giải pháp khác. PHP có khả năng thực hiện và tích hợp chặt

chế với hầu hết các cơ sở dữ liệu có sẵn, tính năng bền vững linh động và khả năng phát triển không giới hạn. Tất cả các đặc tính trên đều miễn phí vì PHP là mã nguồn mở. PHP vừa dễ với người mới sử dụng và vừa có khả năng làm được mọi thứ, đáp ứng yêu cầu của lập trình viên chuyên nghiệp.

PHP được sử dụng càng ngày càng nhiều và mới đây đã bắt kịp ASP (vốn được xem là ngôn ngữ kịch bản phổ biến nhất hiện nay). PHP là modul thông dụng cho Apache (máy chủ Web phổ biến nhất) và nó đã có mặt trên 12 triệu website.

1.1.2. Sử dụng PHP.

a) Các loại thẻ PHP

Có 4 loại khác nhau của thẻ PHP:

- Kiểu Short: là thẻ mặc định mà các nhà lập trình PHP thường sử dụng.

```
<? echo "Hello world"; ?>
```

- Kiểu định dạng XML: thẻ này có thể sử dụng với văn bản dạng XML.

```
<?php echo "Hello world"; ?>
```

- Kiểu Script: trong trường hợp muốn sử dụng PHP như một script tương tự như khai báo Javascript hay VBScript

```
<script language = 'php'>
```

```
echo "PHP script";
```

```
</script>
```

-Kiểu ASP:

```
<%
```

```
echo "Đây là kiểu ASP";
```

```
%>
```

b) Gọi hàm trong PHP

Để gọi hàm trong PHP, khai báo tương tự như gọi hàm trong các ngôn ngữ lập trình khác, hầu hết các hàm khi gọi cần truyền tham số và giá trị trả về.

VD: Gọi hàm date()

<?

```
echo "Hôm nay là: ".date("d/m/Y");
```

?>

c) Truy cập biến Form

Thông thường khi cần lấy dữ liệu của người dùng nhập, chúng ta sẽ dùng thẻ form trong trang web nhằm ràng buộc tất cả các thẻ input, selec, textarea...

Dữ liệu đến từ script đều là biến PHP, chúng ta có thể nhận biết chúng bằng cách sử dụng dấu \$ trước tên biến. Có hai cách để truy cập dữ liệu trên form thông qua biến.

-Để lấy giá trị của các thẻ trong form theo dạng POST:

```
$_POST['tên_thẻ']
```

-Để lấy giá trị của các thẻ trong form theo dạng GET: \$_GET['tên_thẻ']

d) Khai báo biến

PHP Engine không cần yêu cầu khai báo biến trước khi sử dụng, tuy nhiên nên tập thói quen khai báo và khởi tạo giá trị ban đầu cho biến trước khi sử dụng chúng.

e) Gán giá trị cho biến

Để gán giá trị cho biến, nếu biến đó chưa khai báo trước đó, biến này được coi như vừa khai báo và khởi tạo. Trong trường hợp biến đã khai báo, biến này chỉ thay đổi giá trị.

VD: <?php

```
$qty=20;
```

```
$price=10;
```

```
$total=$qty*$price;
```

```
echo "Total: $total";
```

?>

f) Kiểu dữ liệu của biến trong PHP

PHP hỗ trợ 5 kiểu dữ liệu

- Integer: Sử dụng hầu hết cho giá trị có kiểu dữ liệu là số.
- Double: Sử dụng hầu hết cho giá trị có kiểu dữ liệu là số thực.
- String: Sử dụng hầu hết cho giá trị có kiểu dữ liệu là chuỗi, ký tự.
- Array: Sử dụng hầu hết cho giá trị có kiểu dữ liệu là mảng có các phần tử cùng kiểu dữ liệu.
- Object: Sử dụng hầu hết cho giá trị có kiểu dữ liệu là đối tượng của lớp.

g) Toán tử trong PHP

* Các phép toán số học

Toán tử	Tên	Ví dụ
+	Cộng	$a + b$;
-	Trừ	$a - b$;
*	Nhân	$a * b$;
/	Chia	a / b ;
%	Modulo	$a \% b$;

* Các phép toán so sánh

Phép toán	Sử dụng	Ý nghĩa so sánh
$=$	$a = b$	a bằng b
$===$	$a === b$	a bằng và cùng kiểu dữ liệu với b
$!=$	$a != b$	a không bằng b
$<>$	$a <> b$	a không bằng b
$<$	$a < b$	a nhỏ hơn b
$<=$	$a <= b$	a nhỏ hơn hoặc bằng b
$>$	$a > b$	a lớn hơn b
$>=$	$a >= b$	a lớn hơn hoặc bằng b

* Các phép logic

Phép toán	Ký hiệu	Sử dụng	Ý nghĩa
!	NOT	$!a$	Phép toán phủ định

&&	AND	<code>\$a & \$b</code>	Trả về true nếu cả hai biến có giá trị true. Ngược lại là false.
	OR	<code>\$a \$b</code>	Trả về true nếu \$a hay \$b hay cả hai biến có giá trị true
And	AND	<code>\$a and \$b</code>	Trả về true nếu cả hai biến có giá trị true. Ngược lại là false.
Or	OR	<code>\$a or \$b</code>	Trả về true nếu \$a hay \$b hay cả hai biến có giá trị true.

h) Kiểm tra biến trong PHP

- `is_array()`: Kiểm tra biến là array hay không.
- `is_double()`: Kiểm tra biến là double hay không.
- `is_float()`: Kiểm tra biến là float hay không.
- `is_real()`: Kiểm tra biến là real hay không.
- `is_long()`: Kiểm tra biến là long hay không.
- `is_int()`: Kiểm tra biến là int hay không.
- `is_integer()`: Kiểm tra biến là integer hay không.
- `is_string()`: Kiểm tra biến là string hay không.
- `is_object()`: Kiểm tra biến là object hay không.
- `isset()`: Nếu biến tồn tại hàm trả về giá trị true, ngược lại trả về false.
- `empty()`: Cho phép kiểm tra biến tồn tại và không rỗng, có chiều dài khác 0 trả về true, ngược lại là false.

Để sử dụng tất cả các hàm trên, cần phải truyền vào hàm biến PHP dưới dạng tham số.

i) Phát biểu có điều kiện

- Phát biểu If: Phát biểu if với một điều kiện, nếu điều kiện là true thì khối lệnh trong phát biểu If sẽ được thực hiện, điều kiện được khai báo trong dấu ()

VD: `$a = 10;`

```
$b=6;
```

```
if($a>$b)
```

```
echo $a+$b;
```

Khối lệnh trong bất kỳ phát biểu điều khiển nào cũng có thể sử dụng dấu { và }. Có nghĩa là nếu khối lệnh trong phát biểu lớn hơn 1 thì phải sử dụng hai dấu trên.

- Phát biểu Else: phát biểu else luôn là trường hợp ngược lại của phát biểu if với một điều kiện, nếu điều kiện là true thì khối lệnh trong phát biểu if sẽ được thực hiện, ngược lại khối lệnh trong phát biểu else sẽ được thực hiện.

VD: \$a = 10;

```
$b=6;
```

```
if($a>$b) echo $a+$b;
```

```
else
```

```
$b=$a+1;
```

Tương tự như trong trường hợp phát biểu if, nếu khối lệnh trong phát biểu điều khiển else có hơn 1 dòng lệnh thì sẽ phải khai báo sử dụng dấu { và }.

- Phát biểu Switch: tương tự như phát biểu if, nhưng trong trường hợp có nhiều hơn hai tùy chọn cụ thể cho phép quyết định.

VD: switch(\$diem)

```
{
```

```
case "5": echo "Điểm trung bình";
```

```
break;
```

```
case "7": echo "Điểm khá";
```

```
break;
```

```
case "10": echo "Điểm giỏi";
```

break;

default: echo “Nhập lại”;

}

Nếu đúng điều kiện case, cần khai báo phát biểu break nhằm thoát ra khỏi phát biểu switch. Trong trường hợp không khai báo break trong mỗi phát biểu case, nếu thỏa mãn điều kiện trong case nhưng PHP vẫn tiếp tục thực hiện tiếp các phát biểu case sau đó.

- Phát biểu While: phát biểu vòng lặp đơn giản nhất trong PHP là vòng lặp while cho phép bạn thực thi khối lệnh trong while cho đến khi điều kiện của while là true như cú pháp

While (điều kiện)

{

câu lệnh thực hiện;

}

- Phát biểu do ... while: tương tự như while nhưng kiểm tra điều kiện sau khi thực hiện khối lệnh.

do

{

câu lệnh thực hiện;

}

while (điều kiện)

- Phát biểu vòng lặp for: tương tự như trong phát biểu while, có thể sử dụng vòng lặp for với 1 giới hạn chỉ định.

for (giá trị khởi đầu; điều kiện giới hạn; giá trị lặp của vòng lặp for)

{

khối lệnh thực hiện;

}

j) Lưu trữ và truy vấn dữ liệu bằng PHP

- Mở file bằng PHP:

+ fopen(tên file, chế độ mở): mở file

+ puts(), fwrite(): ghi file

+ close(): Đóng file

+ feof(): Nhận giá trị true nếu con trỏ ở cuối file

+ file_exists(): Kiểm tra file có tồn tại

+ filesize(): kiểm tra dung lượng của file

Chế độ mở file:

Mode	Diễn giải
R	Mở dưới dạng Read-Only
r+	Mở dưới dạng Read-Write
W	Mở dưới dạng Write-Only
w+	Mở dưới dạng Write-Read. Nếu file này tồn tại, nội dung sẽ bị xóa, nếu không tồn tại chúng sẽ được tạo ra.
A	Mở dưới dạng append dữ liệu (write), nếu nội dung tồn tại, bắt đầu cuối cùng của nội dung, dữ liệu sẽ được ghi tiếp phần cuối, trong trường
a+	Mở dưới dạng append dữ liệu (write & read), nếu nội dung tồn tại, bắt đầu phần đầu của nội dung, dữ liệu sẽ được ghi tiếp phần đầu, trong trường hợp

- Thiết lập kết nối cơ sở dữ liệu

Để kết nối cơ sở dữ liệu MySQL, cần cung cấp tên Server hay IP (có thể là localhost) của máy chứa cơ sở dữ liệu MySQL, Username và Password để đăng nhập làm việc trên cơ sở dữ liệu được mở.

-Thiết lập kết nối:

```
Int mysql_pconnect("servername", "username", "password");  
  
if(!$db)  
  
{
```

```
echo “Không kết nối được với CSDL”);
```

```
}
```

hoặc

```
$link = mysql_connect(“servername”, “username”, “password”)
```

```
or die(“Không kết nối được với CSDL”);
```

```
-Mở một CSDL: mysql_select_db(“database name”); hoặc
```

```
mysql_select_db(“database name”, $link);
```

```
-Đóng kết nối cơ sở dữ liệu: mysql_close(database_connection);
```

```
-Truy vấn cơ sở dữ liệu:
```

```
Int mysql_query(string sql);
```

```
Int mysql_query(string sql, [int db_connect]);
```

```
Int mysql_query(string database, string sql, [int db_connect]);
```

```
-Số lượng mẫu tin: mysql_num_rows(); Cho biết số lượng mẫu tin câu truy vấn trả về;
```

```
-Truy vấn dữ liệu:
```

```
Int mysql_fetch_array($result); Mysql_fetch_object($result);
```

```
-Dung lượng của tập mẫu tin: int mysql_free_result(int $result);
```

1.2. Giới thiệu MySQL.

1.2.1. Giới thiệu cơ sở dữ liệu:

MySQL là ứng dụng cơ sở dữ liệu mã nguồn mở phổ biến nhất hiện nay và được sử dụng phối hợp với PHP. Trước khi làm việc với MySQL cần xác định các nhu cầu cho ứng dụng.

MySQL là cơ sở dữ liệu có trình giao diện trên Windows hay Linux, cho phép người sử dụng có thể thao tác các hành động liên quan đến nó. Việc tìm hiểu từng công nghệ trước khi bắt tay vào việc viết mã kịch bản PHP, việc tích hợp hai công nghệ PHP và MySQL là một công việc cần thiết và quan trọng.

Chúng ta có thể download MySQL tại website www.mysql.com.

1.2.2. Mục đích sử dụng cơ sở dữ liệu:

Mục đích sử dụng cơ sở dữ liệu bao gồm các chức năng như: lưu trữ, truy cập, tổ chức và xử lý.

1.2.3. Các kiểu dữ liệu trong cơ sở dữ liệu MySQL.

a) Loại dữ liệu numeric

Loại	Range	Bytes	Diễn giải
Tinyint	-127 -> 128 hay 0..255	1	Số nguyên rất nhỏ
Smallint	-32768->32767 hay 0..65535	2	Số nguyên nhỏ
Mediumint	-8388608->8388607 hay 0..16777215	3	Số nguyên vừa
Int	-231 -> 231 hay 0..232-1	4	Số nguyên
Bigint	-263 -> 263 hay 0..264-1	8	Số nguyên lớn
Float	Phụ thuộc số thập phân		Số thập phân dạng Single hay Double
Float(M,D)	$\pm 1.175494351E-38$ ± 3.40282346638	4	Số thập phân dạng Single
Double(M,D)	$\pm 1.7976931348623157308$ $\pm 2.2250738585072014E-308$	8	Số thập phân dạng Double
Float(M[,D])			Số chấm động lưu dưới dạng char

b) Loại dữ liệu String

Loại	Range	Diễn giải
Char	1-255 ký tự	Chiều dài của chuỗi lớn nhất 255 ký tự
Varchar	1-255 ký tự	Chiều dài của chuỗi lớn nhất 255 ký tự

Tinyblob	28-1	Khai báo cho Field chứa kiểu đối tượng nhị phân cỡ 255 ký tự
Tinytext	28-1	Khai báo cho Field chứa kiểu chuỗi cỡ 255 ký tự
Blob	216-1	Khai báo cho Field chứa kiểu blob cỡ 65535 ký tự.
Text	216-1	Khai báo cho Field chứa kiểu chuỗi dạng văn bản cỡ 65535 ký tự.
Mediumblob	224-1	Khai báo cho Field chứa kiểu blob.
		vừa khoảng 16777215 ký tự.
Mediumtext	224-1	Khai báo cho Field chứa kiểu chuỗi dạng văn bản vừa khoảng 16777215 ký tự.
Longblob	232-1	Khai báo cho Field chứa kiểu blob lớn khoảng 4294967295 ký tự.
Longtext	232-1	Khai báo cho Field chứa kiểu blob vừa khoảng 4294967295 ký tự.

c) Loại dữ liệu Date and Time

Loại	Range	Diễn giải
Date	1000-01-01	Date trình bày dưới dạng YYYY-mm-dd
Time	00:00:00 23:59:59	Time trình bày dưới dạng hh:mm:ss

DateTime	1000-01-01 00:00:00 9999-12-31 23:59:59	Date và Time trình bày dưới dạng YYY-mm-dd hh:mm:ss
TimeStamp[(M)]	1970-01-01 00:00:00	TimeStamp trình bày dưới dạng YYYY-mm-dd hh:mm:ss
Year[(2 4)]	1970-2069 1901-2155	Year trình bày dưới dạng 2 số hay 4 số

d) Trình bày đại diện của TimeStamp

Loại	Hiển thị
TimeStamp	YYYYMMDDHHMMSS
TimeStamp(14)	YYYYMMDDHHMMSS
TimeStamp(12)	YYMMDDHHMMSS
TimeStamp(10)	YYMMDDHHMM
TimeStamp(8)	YYYYMMDD
TimeStamp(6)	YYMMDD
TimeStamp(4)	YYMM

(y = năm, m = tháng, d = ngày)

1.2.4. Phát biểu SQL.

Tạo cơ sở dữ liệu:

Create Database <database_name> Create <object type>

<object_name>

Phát biểu SQL thao tác dữ liệu

-SELECT (truy vấn)

-INSERT (Thêm)

-UPDATE (Cập nhật)

-DELETE (Xóa)

a) Phát biểu Select

Select <danh sách cột> From <danh sách bảng>

Where <các điều kiện ràng buộc>

[Group by <tên cột/biểu thức trong Select>]

[Having <điều kiện bắt buộc của Group by>]

[Order by <danh sách cột>]

[Limit fromNumber | To Number]

b) Phát biểu Insert

Insert into <table_name> [<column_name list>] Values (data_values)

c) Phát biểu Update

Update <table_name>

Set <column>=<value>, [<column>=<value>]

[where <restrictive conditions>]

d) Phát biểu Delete

Delete from <table_name> Where <condition>

Một số hàm thông dụng trong MySQL

+ Hàm AVG: Hàm trả về giá trị bình quân của cột hay trường trong câu truy vấn.

VD: select AVG(Amount) From tblOrder

+ Hàm MIN: Hàm trả về giá trị nhỏ nhất của cột hay trường trong câu truy vấn.

VD: select MIN(Amount) From tblOrder

+ Hàm MAX: Hàm trả về giá trị lớn nhất của cột hay trường trong câu truy vấn.

VD: select MAX(Amount) From tblOrder

+ Hàm Count: Hàm trả về số lượng mẫu tin trong câu truy vấn.

VD: select Count(*) From tblOrder

+ Hàm Sum: Hàm trả về tổng các giá trị của trường, cột trong câu truy vấn.

VD: select SUM(Amount) From tblOrder

+ Hàm CurDate(): hàm trả về ngày, tháng, năm hiện hành của hệ thống.

VD: select curdate() as 'Ngày hiện tại'

+Hàm CurTime(): Hàm trả về giờ phút giây hiện hành của hệ thống.

VD: select curtime() as 'Thời gian'

