



Tổng quan PHP và cài đặt ứng dụng

Nội dung

Giới thiệu về PHP
Cài đặt môi trường
Nhúng mã PHP vào tài liệu HTML
Cú pháp của PHP
Biến trong PHP
Các toán tử trong PHP
Làm việc với FORM trong PHP

Giới thiệu PHP

- Là một ngôn ngữ kịch bản chạy trên server, được dùng để xây dựng các ứng dụng web động
- Được ra đời năm 1994 do Rasmus Lerdorf
- PHP được (công bố ngày 02/02/2012)
- Mã PHP trong một kịch bản có thể là các câu truy vấn csdl, tạo ảnh, đọc/ghi file...
- Kết quả sau khi thực hiện mã PHP được kết hợp với HTML và gửi đến cho trình duyệt, do đó người dùng không thể xem được mã của PHP từ trình duyệt.
- PHP được biên dịch như một module của Apache (Apache là một web-server được sử dụng phổ biến trên thế giới).

Giới thiệu PHP

- Mã nguồn mở (open source code).
- Miễn phí, dễ dàng download từ Internet.
- Ngôn ngữ rất dễ học, dễ viết.
- PHP có thể chạy trên các môi trường (platforms) khác nhau như (Windows, Linux, Unix, etc.)
- Có thể kết nối với nhiều DDMS một các đơn giản như MySQL, Microsoft SQL Server 2000, Oracle, PostgreSQL

Cài đặt môi trường

- Môi trường chạy PHP XAMPP, WAMP,...
- Xampp là chương trình tạo máy chủ Web (Web Server) trên máy tính cá nhân (Localhost)
- Được tích hợp sẵn Apache, PHP, MySQL, FTP Server
- Tải Xampp (<https://www.apachefriends.org>)
- Cài Đặt Xampp

- Khi cài đặt cài XAMPP, các chương trình sau cài được tự động
 - ✓ **Apache**: là một chương trình dành cho máy server đảm nhận việc giao tiếp bằng giao thức HTTP.
 - ✓ **PHP**: là một "plugin" của Apache, giúp Apache biết cách làm việc với các trang PHP.
 - ✓ **MySQL**: là hệ quản trị cơ sở dữ liệu
 - ✓ **PhpMyAdmin**: Là một tool giúp quản trị cơ sở dữ liệu trong MySQL

Ngôn ngữ kịch bản PHP

PHP gồm những gì?

- Các file PHP có thể chứa text, HTML tags và các đoạn scripts
- Các PHP sau khi Thông dịch trả về cho trình duyệt là các trang HTML
- File PHP có phần mở rộng là ".php", ".php3", hoặc ".phtml"

Nhúng mã PHP vào HTML

- Mã PHP vào vị trí bất kì trong trang HTML.
- Một khối mã lệnh PHP được đặt giữa `<?php` và `?>`
- Một cấu trúc lệnh thông thường của PHP có thể được tách làm nhiều phần, mỗi phần đặt giữa `<?php...?>`
- Kết quả do đoạn lệnh PHP thực thi được đưa vào vị trí mà đoạn lệnh PHP đang chiếm chỗ.

Cú pháp của PHP

- Sử dụng kí tự ; để kết thúc mỗi câu lệnh
- Chú thích trong PHP:
 - ✓ Dùng // để đánh dấu một dòng chú thích.
 - ✓ Dùng /* và */ để đánh dấu một khối chú thích trên nhiều dòng

```

<html>
  <body>
    <?php
      // Dòng chú thích

      /*
        Các dòng chú thích
      */
    ?>
  </body>
</html>

```

- Biến dùng để lưu trữ dữ liệu. Kiểu của dữ liệu mà biến lưu trữ có thể là `number`, `string`, `boolean`, `object`, hoặc `NULL`
- PHP tự động chuyển kiểu dữ liệu của biến phù hợp với kiểu của giá trị.
- Quy tắc đặt tên biến
 - ✓ Tên biến luôn bắt đầu bằng kí tự đô la \$.
 - ✓ Tên biến nên bao gồm chữ cái, chữ số, dấu gạch nối (_).
 - ✓ Không sử dụng dấu cách, tiếng việt có dấu cho tên biến
 - ✓ Có phân biệt chữ hoa, chữ thường.

Biến trong PHP

- Cú pháp khai báo và sử dụng một biến:

```
$tên_biến = giá_trị;
```

- Ví dụ

```
<?php
```

```
$txt = "Hello World!";
```

```
$number = 16;
```

```
?>
```

- Biến động:
 - ✓ Cho phép sử dụng một giá trị chuỗi của một biến để chỉ ra tên cho một biến khác.
- Ví Dụ :

<?php

```
$x = 'y';
```

```
$$x = 'z';
```

```
echo $x; // outputs 'y'
```

```
echo $y; // outputs 'z'
```

```
echo $$x; // outputs 'z'
```

?>

- Xác định kiểu của biến:

`gettype($tên_biến);`

Hàm `getType()` : trả về một trong các giá trị: "integer", "double", "string", "array", "object", "class", "unknown type"

- Ép kiểu

```
$a = 11.2 ;
```

```
$a = (int) $a ; //a = 11
```

```
$a = (double) $a ; //a = 11.0
```

```
$b = (string) $a ;
```

- Hàm chuyển đổi kiểu dữ liệu của biến

```
settype($tên_biến, "kiểu_dữ_liệu")
```

```
$a=7.5;
```

```
settype($a, "integer");
```

```
// $a=(int) $a;
```

- Hàm **unset(\$ten_bien)**

- Dùng huỷ bỏ biến *\$ten_bien*. Giải phóng bộ nhớ bị chiếm dụng bởi biến *\$ten_bien*

Sử dụng biến trong PHP

- Hàm kiểm tra giá trị của biến

- Hàm `isset($ten_bien)`

Dùng để kiểm tra xem biến `$ten_bien` có giá trị gì chưa.

Trả về TRUE nếu giá trị của biến khác NULL, ngược lại trả về FALSE

- Hàm `empty($ten_bien)`

Dùng để kiểm tra xem giá trị của biến `$ten_bien` có là `false`, `0`, chuỗi rỗng hoặc NULL không.

Trả về TRUE nếu có, ngược lại trả về FALSE

- Hàm `is_string($ten_bien)`

Dùng để kiểm tra xem kiểu của biến `$ten_bien` có phải là kiểu string không

- Không thể thay đổi giá trị sau khi gán.
- Có thể truy xuất hằng thông qua tên của nó
- Quy tắc về hằng:
 - Tên **không** bắt đầu bằng kí tự **(\$)**.
 - Sử dụng hàm **define()** để định nghĩa một hằng.
 - Được định nghĩa và truy xuất **toàn cục**

Ví Dụ :

```
define ("HELLO", "Xin chào Devmaster");  
echo HELLO;  
echo constant ("HELLO") ;
```


- Một số hằng xây dựng sẵn
 - **TRUE** có giá trị 1, **FALSE** có giá trị 0 (hay một chuỗi rỗng)
 - **__FILE__** : trả về tên của tập tin hiện đang được đọc bởi trình biên dịch
 - **__LINE__** : trả về số dòng hiện hành trong file
 - **__FUNCTION__** : trả về tên hàm đã được khai báo
 - **__CLASS__** : trả về tên lớp(class) đã được khai báo
 - **__METHOD__** : trả về tên phương thức của lớp đã được khai báo

Kiểu dữ liệu trong PHP

- Các kiểu dữ liệu được hỗ trợ trong PHP
 - Kiểu **Boolean** (có giá trị TRUE hoặc FALSE).
 - Kiểu số (int, float)
 - Kiểu string: Các giá trị được đặt trong nháy đơn hoặc kép
 - Mảng & đối tượng
 - Ngoài ra còn có kiểu "NULL", khi không có giá trị gán cho kiểu dữ liệu.

Chú ý : *PHP không yêu cầu phải khai báo kiểu dữ liệu cho biến trước khi dùng, và chỉ cần hiểu dữ liệu theo bối cảnh mà dữ liệu được sử dụng.*

- Có thể dùng hàm `gettype()` để biết kiểu dữ liệu của biến, và hàm `settype()` để thay đổi kiểu của biến

- Chỉ có một trong hai giá trị là TRUE (đúng) hoặc FALSE (sai)
- Khi thực hiện ép kiểu từ một kiểu dữ liệu khác sang Boolean:
 - Các giá trị như số nguyên 0, số thực 0.0, chuỗi rỗng "", hoặc chuỗi "0", mảng rỗng (không chứa phần tử nào) Array(), giá trị NULL sẽ được chuyển thành giá FALSE.
 - Các giá trị còn lại sẽ được chuyển đổi thành TRUE

Kiểu Integer - Số nguyên

- Có kích thước 32 bit (mang giá trị từ -2147483647 cho đến 2147483648)

- Số nguyên biểu diễn trong hệ thập phân(Decimal):

Ví dụ:

```
$a = 16;
```

- Số nguyên biểu diễn trong hệ 16 (Hexadecimal):

Ví dụ:

```
$a=0x10;
```

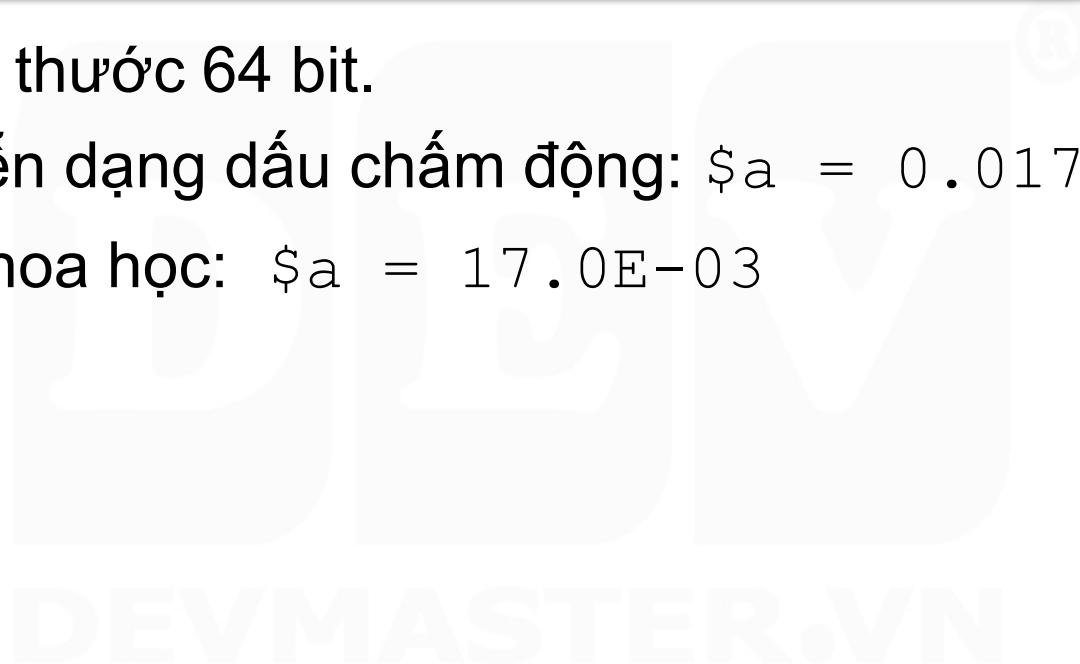
- Số nguyên biểu diễn trong hệ 8 (Octal):

Ví dụ:

```
$a=020;
```

Kiểu Float - Số thực

- Có kích thước 64 bit.
- Biểu diễn dạng dấu chấm động: $a = 0.017$
- Dạng khoa học: $a = 17.0E-03$



Kiểu String (chuỗi)

- Kiểu string lưu giữ 1 chuỗi ký tự, mỗi ký tự có kích thước 1 byte.
- Nội dung string được đặt giữa 2 dấu nháy, nháy đơn (') hoặc nháy kép (").
- Chuỗi đặt trong nháy kép bị thay thế và xử lý ký tự thoát. Trong nháy đơn thì không.

Ví dụ:

```
$a = "Hello";
$b = "$a world"; //tương đương $b="Hello world"
$c = '$a world'; //$c='$a world' (không thay đổi)
```

Kiểu String (chuỗi)

- Sử dụng ký tự thoát(escape): \
- Để viết các ký tự đặc biệt trong chuỗi.

Ví dụ:

Cần có chuỗi: Người ta nói "PHP rất tốt"

```
$a = "Người ta nói "PHP rất tốt""; //Sai
```

```
$a = "Người ta nói \"PHP rất tốt\""; //Đúng
```

Ngoài ra:

- \n: Ký tự xuống dòng
- \r: Ký tự trở về đầu dòng
- \t: dấu tab
- \\$ Ký tự \$

Kiểu String (chuỗi)

- Để làm rõ các biến trong chuỗi, cần bao biến vào giữa cặp ngoặc nhọn `{ }`

```
$a = "He";  
$b = "$aello"; //lỗi vì PHP hiểu là $aello  
$c = "{$a}ello"; //đúng ($c = "Hello")
```

- Biến mảng, biến biến, biến trong tên có dấu gạch nối hoặc dấu chấm khi viết trong chuỗi bắt buộc phải bao lại.

Kiểu String (chuỗi)

Ghép chuỗi

- Sử dụng toán tử . (dấu chấm) để ghép nối một hoặc nhiều chuỗi văn bản với nhau.
- Có thể dùng để nối chuỗi với giá trị kiểu dữ liệu khác, kết quả trả về là một chuỗi mới.

```
<?php
```

```
$my_string="Hello Devmaster. My name is: ";
```

```
$newline= "<br/>";
```

```
echo $my_string . "John" . $newline ;
```

```
?>
```

Kiểu String (chuỗi)

- Truy xuất một kí tự trong chuỗi
- ✓ Có thể truy xuất đến từng kí tự riêng biệt trong chuỗi bằng cú pháp sau:

`$tên_biến{vị_trí} // vị_trí bắt đầu bằng 0`

<?php

```
$st="P";  
$st[2]="H";  
$st[1]="P";  
echo $st;
```

?>

Toán tử trong PHP

- Các loại toán tử
 - Toán tử toán học
 - Toán tử tăng giảm
 - Toán tử so sánh
 - Toán tử điều kiện
 - Toán tử gán

Toán tử trong PHP

- Toán tử toán học

Toán tử	Tên	Ví dụ	Kết quả
+	Cộng	3+2	5
-	Trừ	3-2	1
*	Nhân	3*2	6
/	Chia	3/2	1.5
%	Chia dư	3%2	1

Toán tử trong PHP

- Toán tử tăng

```
$a = 1;  
$a++; // a = a + 1  
echo $a; // result 2
```

- Toán tử giảm

```
$a = 1;  
$a--; // a = a - 1  
echo $a; // result 0
```

- Toán tử so sánh

Toán tử	Mô tả	Ví dụ
==	Bằng	2=3 => false
>	Lớn hơn	2>3 => false
<	Nhỏ hơn	2< 3 => true
>=	Lớn hơn hoặc bằng	2>=3 =>false
<=	Nhỏ hơn hoặc bằng	2<=3 =>true
!=	Không bằng	2!=3 => true
===	Đồng nhất	

Toán tử trong PHP

- Toán tử điều kiện

`(điều_kiện) ? (biểu_thức_1) : (biểu_thức_2)`

- Ví dụ tìm giá trị lớn nhất của ba số

```
$a = 60;
```

```
$b = 30;
```

```
$c = 15;
```

```
$max = ($a < $b) ? $b : $a ;
```

```
$max = ($max < $c) ? $max : $c ;
```

```
echo $max;
```

- Toán tử gán

Toán tử	Ví dụ	Diễn giải
<code>+=</code>	<code>\$x += 3</code>	<code>\$x = \$x + 3</code>
<code>-=</code>	<code>\$x -= 3</code>	<code>\$x = \$x - 3</code>
<code>*=</code>	<code>\$x *= 3</code>	<code>\$x = \$x * 3</code>
<code>/=</code>	<code>\$x /= 3</code>	<code>\$x = \$x / 3</code>
<code>.=</code>	<code>\$x .= '3'</code>	<code>\$x = \$x . '3'</code>

Toán tử trong PHP

▪ Toán tử logic

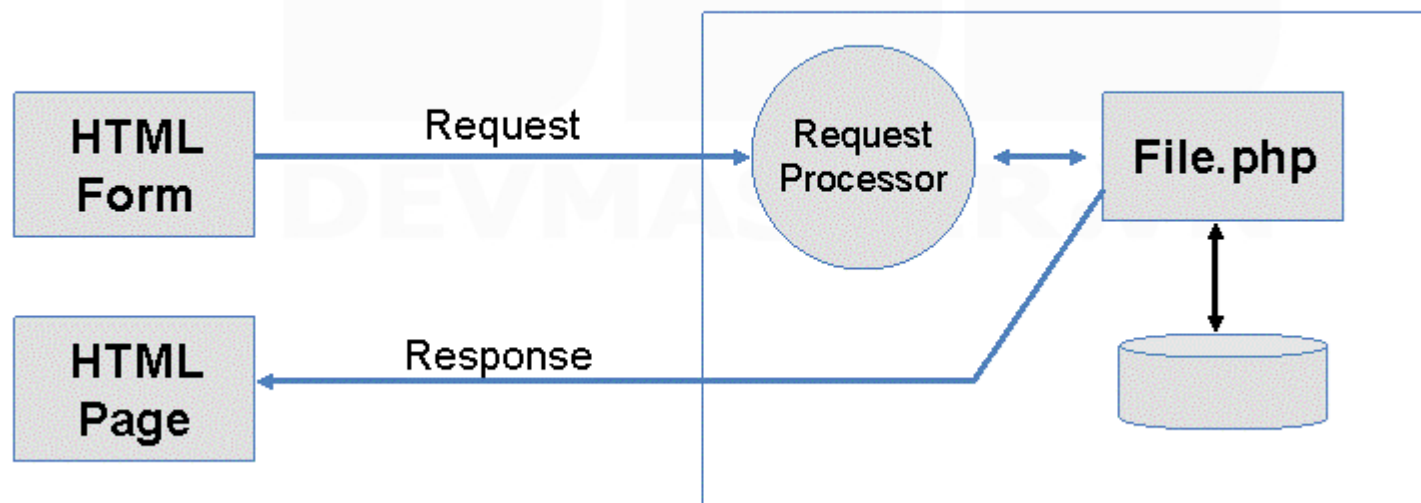
Toán tử	Mô tả	Ví dụ
&& , And	Và	<pre>x=60 y=30 (x < 100 && y > 10) trả lại true</pre>
 , OR	Hoặc	<pre>x=60 y=30 (x==50 y==50) trả lại false</pre>
! , NOT	Phủ định	<pre>x=60 y=30 !(x==y) trả lại true</pre>

Làm việc với HTML FORM

- HTML FORM

Được sử dụng để truyền dữ liệu do người dùng nhập từ client tới server

- Quá trình xử lý yêu cầu trên Web Server



Làm việc với HTML FORM

■ Khi tạo HTML form cần:

- ✓ Chỉ rõ đích xử lý dữ liệu trong thuộc tính `action`.
- ✓ Chỉ rõ phương thức gửi dữ liệu từ client tới server là `POST` hay `GET` trong thuộc tính `method`
- ✓ Dữ liệu trong form bao gồm các `INPUT`

```
<FORM name="dangnhap" action="Xuly.php" method="po
```

```
Tên đăng nhập: <br>
```

```
<INPUT type="text" name='txtTendangnhap'><br>
```

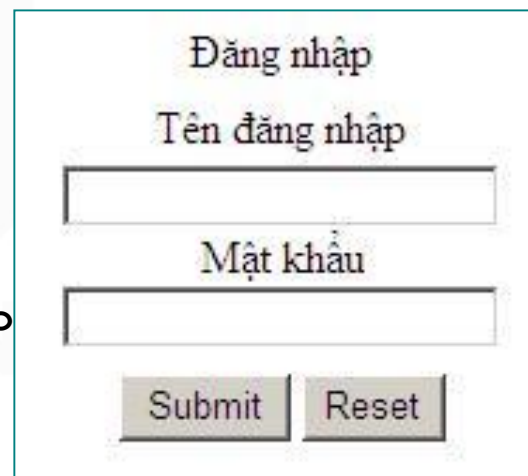
```
Mật khẩu:
```

```
<INPUT type="password" name='txtMatkhau'><br>
```

```
<INPUT type="submit" name='btubmit' value="Submit">
```

```
<INPUT type="reset" name='btReset' value="Submit">
```

```
</FROM>
```



Phương thức và biến `$_POST`

- Dữ liệu được điền trong form được gửi về server khi người click vào nút submit.
- Web Server lưu dữ liệu được gửi về từ phía client vào biến `$_POST` hoặc `$_GET` tùy theo phương thức (method) gửi dữ liệu của form.
- Web Server gọi file .php được chỉ ra trong thuộc tính action của form để xử lý dữ liệu của form.

Xuly.php

```
<?php
```

```
$tendangnhap=$_POST['txtTendangnhap'];
```

```
$matkhau=$_POST['txtMatkhau'];
```

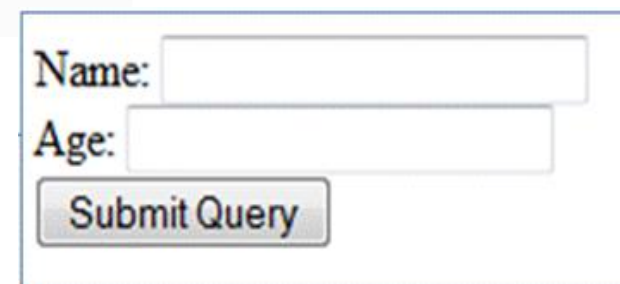
```
?>
```

Phương thức và biến \$_GET

- Biến \$_GET chứa dữ liệu gửi tới Web Server theo phương thức GET
- Khi form được gửi theo phương thức GET, các biến và giá trị được tổ chức thành chuỗi tham số đính kèm sau URL. Người dùng có thể thấy được

```
<form action="welcome.php" method="get">
  Name: <input type="text" name="fname" />
  Age: <input type="text" name="age" />
  <input type="submit" />
</form>.
```

get.php




Submit

```
http://localhost/welcome.php?fname=Tuan&age=20
```

Biến Request \$_REQUEST

- Biến \$_REQUEST chứa dữ liệu của biến \$_GET và \$_POST và \$_COOKIE,
- Có thể được dùng để lấy dữ liệu của phương thức POST, GET

```
Welcome <?php echo $_REQUEST["fname"]; ?>!  
You are <?php echo $_REQUEST["age"]; ?> years old.
```

welcome.php



HẾT
XIN CẢM ƠN!