

LẬP TRÌNH BACK-END WEB 1

| Back-end Web Development 1|

Bùi Thị Phương Thảo - Nguyễn Huy Hoàng
[09-2021]



FACULTY OF INFORMATION TECHNOLOGY
THU DUC COLLEGE OF TECHNOLOGY

Back-end Web Development 1

1



Chương 1

Cơ Bản Về Lập Trình PHP



FACULTY OF INFORMATION TECHNOLOGY
THU DUC COLLEGE OF TECHNOLOGY

Back-end Web Development 1

2



| Nội dung

1.1 Giới thiệu và cài đặt

1.2 Các kiểu dữ liệu

1.3 Hàm

1.4 Biểu thức điều kiện

1.5 Nhận dữ liệu từ Form



1.1 Giới thiệu và cài đặt

- Là ngôn ngữ lập trình scripting language ở phía server.
- "PHP: Hypertext Preprocessor", là ngôn ngữ mã nguồn mở, miễn phí download và sử dụng.
- File PHP (*.php) có thể bao gồm HTML, CSS, JavaScript, và code PHP.
- Các file PHP sẽ được chạy và xử lý ở server. Kết quả của quá trình xử lý được gửi trả về cho browser là những file HTML thuần túy



| 1.1 Giới thiệu và cài đặt

PHP có thể thực hiện các công việc sau:

- Tạo ra những trang web có nội dung động.
- Tạo, mở, đọc, ghi, xóa, đóng file trên server.
- Thu thập dữ liệu từ form.
- Gửi và nhận cookies.
- Thêm, xóa, sửa dữ liệu trong cơ sở dữ liệu.
- Quản lý quyền truy cập của người dùng.
- Mã hóa dữ liệu.



1.1 Giới thiệu và cài đặt

➤ Cài đặt LAMP stack



WampServer



XAMPP



AMPPS



https://www.w3schools.com/php/phptryit.asp?filename=tryphp_intro

| 1.1 Giới thiệu và cài đặt

➤ Cú pháp PHP cơ bản

- Code PHP có thể được đặt ở bất kỳ nơi đâu trong file *.php, và nằm trong cặp **<?php** //code php here **?>**
- Nếu file *.php chỉ chứa thuần túy code PHP mà không có bất kỳ gì khác, chúng ta nên bỏ ?> đi, chỉ giữ lại <?php ở đầu file
- Một câu lệnh PHP được kết thúc bằng dấu chấm phẩy (;)



| 1.1 Giới thiệu và cài đặt

Ví dụ:

```
<!DOCTYPE html>
<html>
<body>

<?php
echo "My first PHP script!";
?>

</body>
</html>
```

Nguồn: w3schools.com



| 1.1 Giới thiệu và cài đặt

➤ Comment trong PHP

- // Ghi chú một dòng đơn
- # Ghi chú một dòng đơn
- /* và */ Ghi chú một đoạn nhiều dòng. Để mở đầu đoạn ghi chú, ta dùng dấu /* và để kết thúc, ta dùng dấu */



1.1 Giới thiệu và cài đặt

➤ Biến trong PHP

Bắt đầu bằng dấu \$ theo sau đó là tên biến.

Ví dụ:

- `$txt = "Hello world!";`
- `$x = 5;`
- `$y = 10.5;`



Biến - Quy tắc đặt tên

- Một biến bắt đầu bằng dấu \$ và theo sau đó là tên biến
- Tên biến bắt buộc phải bắt đầu bằng ký tự hoặc dấu _
- Tên biến không được bắt đầu bằng ký số.
- Tên biến chỉ có thể chứa ký tự (A-z), ký số (0-9) và dấu _
- Tên biến phân biệt chữ hoa và chữ thường
- Thông thường tên biến được viết theo kiểu camelCase



Biến - Xuất giá trị

Cách 1:

```
$txt = "FIT-TDC";  
echo "I love $txt!";
```

Cách 2:

```
$txt = " FIT-TDC";  
echo "I love " . $txt . "!";
```



| Biến - Phạm vi của biến

Phạm vi của biến là tầm hoạt động của biến nơi biến có thể được sử dụng sau khi đã khai báo.



Biến - Phạm vi Global

Biến được khai báo ở bên ngoài một hàm và chỉ có thể được sử dụng ở bên ngoài hàm.

Ví dụ:

```
<?php
$x = 5; // global scope
function myTest() {
    // using x inside this function will generate an error
    echo $x;
}
myTest();
?>
```

Biến - Phạm vi Global

Sử dụng từ khóa “global” trước tên biến trong hàm sử dụng biến global

```
<?php
$x = 5; // global scope
function myTest() {
    // using global keyword to define a global variable
    global $x;
    echo $x;
}
myTest();
?>
```

Biến - Phạm vi Global

Ngoài ra, có thể sử dụng biến `$GLOBALS[index]` (trong đó index là tên của biến) để truy vấn giá trị của biến toàn cục như ví dụ bên dưới

```
<?php
$x = 5;
$y = 10;
function myTest() {
    $GLOBALS['y'] = $GLOBALS['x'] + $GLOBALS['y'];
}
myTest();
echo $y; // outputs 15
?>
```



Biến - Phạm vi local

- Biến được khai báo ở bên trong một hàm và chỉ có thể được sử dụng ở bên trong hàm
- Để các biến local không bị hủy đi sau khi kết thúc hàm, ta dùng từ khóa “static” lúc khai báo biến như sau:

```
<?php
function myTest() {
    static $x = 0;
    echo $x;
    $x++;
}

myTest(); // $x=0
myTest(); // $x=1
myTest(); // $x=2
?>
```

Biến - Superglobals

PHP hỗ trợ thêm các biến superglobals được khai báo sẵn (có thể được sử dụng ở bất kỳ phạm vi nào):

- `$GLOBALS`
- `$_SERVER`
- `$_REQUEST`
- `$_POST`
- `$_GET`
- `$_FILES`
- `$_ENV`
- `$_COOKIE`
- `$_SESSION`

https://www.w3schools.com/php/php_superglobals.asp



| Hãng - Quy tắc đặt tên

- Khi hãng được khai báo, giá trị của nó sẽ không bị thay đổi hoặc hủy.
- Mặc định, hãng sẽ có phạm vi global và có thể được sử dụng ở bất kỳ đâu sau khi khai báo.
- Tên hãng phải bắt đầu bằng ký tự hoặc dấu _ và không có dấu \$ như biến.
- Thông thường, tên hãng sẽ được viết IN toàn bộ, các từ sẽ cách nhau bằng dấu _.



| Hằng - Khai báo hằng

define(*name*, *value*)

- *name*: tên hằng
- *value*: giá trị của hằng



| Hằg - Khai báo hằg

Ví dụ:

```
<?php  
define("TDC", "Trường Cao Đằg Công Nghệ Thủ Đức");  
echo TDC; //case-sensitive  
?>
```



1.2 Các kiểu dữ liệu

- **String:** kiểu chuỗi ký tự. Một chuỗi phải được đặt trong dấu ' hoặc "
- **Integer:** số nguyên từ -2 147 483 648 đến 2 147 483 647.
- **Float:** số thực gồm hai phần, phần nguyên và phần thập phân, được ngăn cách nhau bằng dấu .
- **Boolean:** kiểu chân trị, gồm hai giá trị TRUE hay FALSE.
- **Object:** kiểu đối tượng
- **NULL:** chỉ có duy nhất một giá trị là NULL (không có giá trị).
- **Resource:** resource không phải là một kiểu dữ liệu thật sự. Nó được dùng để lưu trữ các tham chiếu đến các hàm hoặc cái tài nguyên bên ngoài.



1.3 Hàm

- Để tự định nghĩa một hàm trong PHP, ta sử dụng cú pháp sau:

```
function functionName()  
{  
    code to be executed;  
}
```

```
<?php  
function writeMsg() {  
    echo "Hello world!";  
}  
  
writeMsg(); // call the function  
?>
```

1.3 Hàm - Tham số

- Tham số có thể chứa giá trị mặc định.
- Nếu trong một hàm có lẫn lộn tham số có/không có giá trị mặc định thì các tham số không có giá trị mặc định sẽ được đặt bên trái, các tham số có giá trị mặc định sẽ được đặt bên phải.

```
function functionName(arg1,arg2,arg3=defaultvalue3,...) {  
  
    code to be executed;  
  
}
```


1.4 Biểu thức điều kiện

Cũng tương tự như các ngôn ngữ lập trình đã học, PHP hỗ trợ các biểu thức điều kiện sau:

- if
- if...else
- if...elseif...else
- switch

Cách sử dụng các biểu thức điều kiện này tương tự như các ngôn ngữ lập trình đã học



| Vòng lặp

Cũng tương tự như các ngôn ngữ lập trình đã học, PHP hỗ trợ các vòng lặp sau:

- while
- do...while
- for
- foreach - duyệt qua từng phần tử mảng

Cách sử dụng các vòng lặp này tương tự như các ngôn ngữ lập trình đã học.



| Các toán tử

https://www.w3schools.com/php/php_operators.asp



1.5 Nhận dữ liệu từ Form

- PHP sử dụng hai biến mảng superglobals là `$_GET` và `$_POST` để nhận dữ liệu từ form.
- Form có `method="get"`, ta dùng `$_GET`.
- Form có `method="post"`, ta dùng `$_POST`

`$_GET["tên của field cần lấy dữ liệu"]`

`$_POST["tên của field cần lấy dữ liệu"]`



1.5 Nhận dữ liệu từ Form

Ví dụ: Cho một form gửi 2 thông tin **name** và **email** qua trang welcome.php bằng phương thức post

```
<form action="welcome.php" method="post">  
Name: <input type="text" name="name"><br>  
E-mail: <input type="text" name="email"><br>  
<input type="submit">
```



1.5 Nhận dữ liệu từ Form

```
<form action="welcome.php" method="post">  
Name: <input type="text" name="name"><br>  
E-mail: <input type="text" name="email"><br>  
<input type="submit">
```

Tại trang welcome.php, sử dụng biến `$_POST['name']` và `$_POST['email']` để lấy giá trị form truyền qua

```
<?php  
echo $_POST['name'];  
echo $_POST['email'];  
?>
```



1.5 Nhận dữ liệu từ Form

- Việc sử dụng form để truyền dữ liệu tiềm ẩn nhiều nguy cơ bảo mật có thể gây nguy hiểm cho website và người dùng.
- Bất kỳ dữ liệu nào được nhận từ form đều phải trải qua quá trình kiểm tra và xử lý trước khi có thể được dùng



Thanks for your attention!



FACULTY OF
INFORMATION TECHNOLOGY
THU DUC COLLEGE OF TECHNOLOGY

Phone: (+848) 22 158 642

Email: fit@tdc.edu.vn

Website: fit.tdc.edu.vn

Facebook: facebook.com/tdc.fit

Youtube: youtube.com/fit-tdc



FACULTY OF INFORMATION TECHNOLOGY
THU DUC COLLEGE OF TECHNOLOGY

