

LẬP TRÌNH WEB BACK-END 1

| Back-end Web Development 1 |

Bùi Thị Phương Thảo - Nguyễn Huy Hoàng
[09 . 2019]

Dùng kèm Giáo trình Lập trình web back-end 1, Khoa Công nghệ thông tin, Cao đẳng Công nghệ Thủ Đức



Bài 1.

CƠ BẢN VỀ LẬP TRÌNH PHP

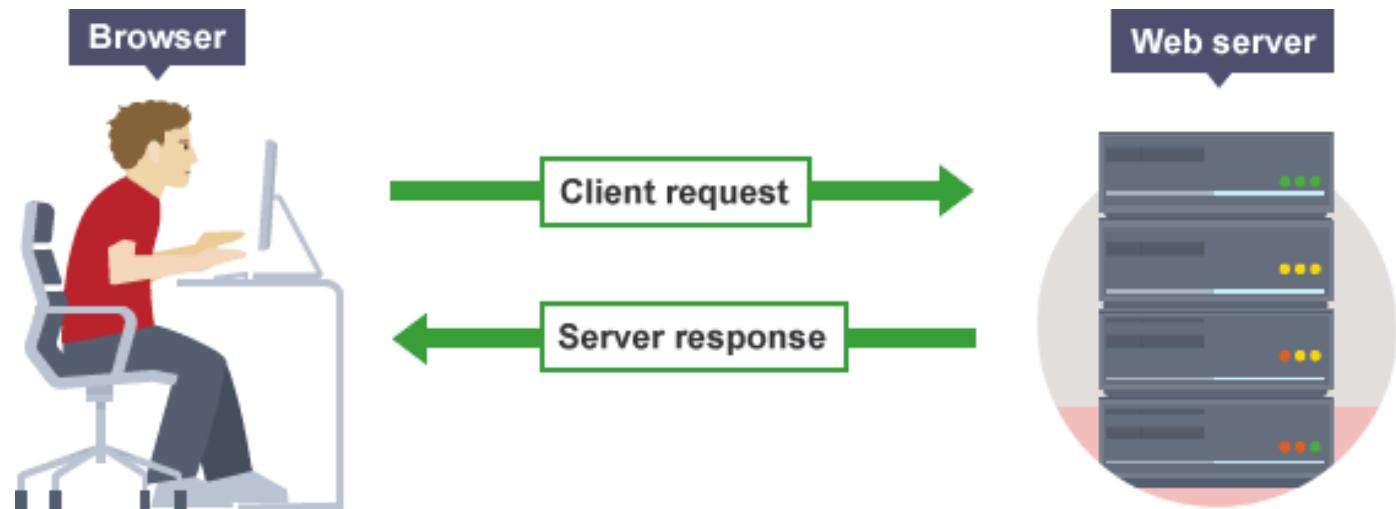
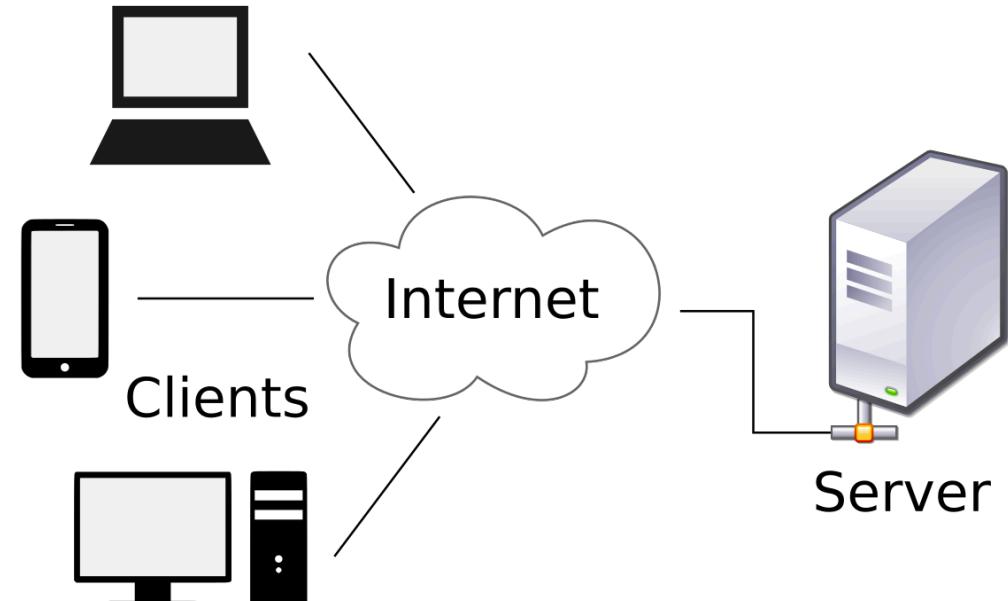


1.1

MÔ HÌNH CLIENT – SERVER

Mô hình Client – Server

Câu hỏi: Trình bày mô hình Client-Server đã học?



Mô hình Client – Server



- Bắt cặp 2 sinh viên và xem video sau:
<https://www.youtube.com/watch?v=WY8jwTNYTfg>
- Mỗi cặp chuẩn bị trong vòng 15ph và trình bày lại nội dung video.



1.2 **VÍ DỤ MỞ ĐẦU**

Ví dụ mở đầu

Khởi tạo trang <http://localhost>

- Khởi động Wamp
- Khởi động Sublime
- Khởi động Chrome

Các lỗi có thể xảy ra

- Proxy => localhost*
- Port => 82

Ví dụ mở đầu

Tạo project tdc_web_1 tại thư mục (*/www/)

Tạo file index.php

```
<html>
<head>
<title>My first PHP page</title>
</head>
<body>
<h1>Hello World!</h1>
</body>
</html>
```



Hello World!

||| Cấu trúc folders lưu trữ website

Câu hỏi: Nêu cách tổ chức folders/files lưu trữ website ở môn thiết kế web.

||| Cấu trúc folders lưu trữ website

Ở môn Lập trình web 1:

- Cấu trúc lưu trữ kế thừa từ môn Thiết kế web
- Các file có đuôi *.html được đổi thành đuôi *.php
- Tùy theo mô hình phát triển website được sử dụng sẽ có cách tổ chức lưu trữ khác nhau. Tuy nhiên, để thống nhất cho các học phần sau và phù hợp với thực tế, sinh viên phải tổ chức lưu trữ theo các bài được học (tùy theo bài học sẽ thêm vào folders/files).
- Khi học đến bài cuối cùng sẽ được một cấu trúc lưu trữ hoàn chỉnh.

Ví dụ mở đầu: index.php

Tích hợp PHP

```
1 <html>
2 <head>
3   <title>PHP Test</title>
4 </head>
5 <body>
6
7   <?php echo '<p>Hello World</p>'; ?>
8
9 </body>
10 </html>|
```



Hello World!

||| Ví dụ mở đầu

- Copy file index.php đến một thư mục khác bên ngoài */www/
- Chạy lại file index.php và nhận xét?



1.3

PHP CĂN BẢN



1.3.1

GIỚI THIỆU PHP

||| Giới thiệu PHP

- Là ngôn ngữ lập trình scripting language ở phía server rất phổ biến.
- Được viết tắt của "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.

||| Giới thiệu PHP

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.

||| Giới thiệu PHP

PHP được dùng phổ biến hiện nay bởi các lý do sau:

- PHP có thể chạy trên các nền tảng khác nhau (Windows, Linux, Unix, Mac OS X, v.v...)
- PHP tương thích với hầu hết tất cả server hiện nay (Apache, IIS, v.v...)
- PHP hỗ trợ nhiều loại cơ sở dữ liệu khác nhau.
- PHP miễn phí, có thể được download tại website chính thức của PHP là: www.php.net
- PHP dễ học và hoạt động hiệu quả ở phía server.



1.3.2 **CÀI ĐẶT**

Cài đặt LAMP stack



WampServer



XAMPP



AMPPS



1.3.3

CÚ PHÁP PHP CƠ BẢN

Cú pháp PHP cơ bản

- Code PHP có thể được đặt ở bất kỳ nơi đâu trong file *.php
- Một đoạn code PHP bắt đầu bằng **<?php** và kết thúc bằng **?>**
- (short open tag **<?** không được khuyến khích dùng)
- 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 (**;**)

Ví dụ cơ bản

```
<!DOCTYPE html>
<html>
<body>
    <h1>My first PHP page</h1>
    <?php
        echo "Hello World!";
    ?>
</body>
</html>
```

||| 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 */



1.3.4 BIẾN (VARIABLES)

||| Biến

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

Ví dụ:

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

||| Biến

Các lưu ý khi làm việc với biến:

- Một biến bắt đầu bằng dấu \$ và 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 – In giá trị của biến

- Cách 1:

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

- Cách 2:

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

Hãy thử thay dấu " ở cách 1 bằng dấu ' và cho nhận xét.

|| Phạm vi của biển

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

Phạm vi global

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

```
$x = 5; // global scope
function myTest() {
    // Sử dụng biến x trong hàm sẽ gây ra lỗi
    echo "<p>Giá trị biến x là: $x</p>";
}
myTest();
echo "<p>Giá trị biến x là: $x</p>";
```

Phạm vi global

- Để có thể sử dụng được biến global bên trong một hàm, ta dùng từ khóa “**global**” trước tên biến khi sử dụng biến bên trong hàm

```
$x = 5;  
$y = 10;  
function myTest() {  
    global $x, $y;  
    $y = $x + $y;  
}  
myTest();  
echo $y; // outputs 15
```

|| Phạm vi global

- PHP lưu trữ giá trị của tất cả các biến global trong một mảng là **\$GLOBALS[index]**, trong đó **index** là tên của biến.
- Mảng này có thể được truy xuất bên trong hàm và có thể được dùng để cập nhật lại giá trị cho biến global một cách trực tiếp.

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

Phạm vi local

- Biến được khai báo ở **bên trong** một hàm và chỉ có thể được sử dụng ở bên trong hàm.

```
function myTest() {  
    $x = 5; // local scope  
    echo "<p>Giá trị biến x là: $x</p>";  
}  
myTest();  
// Sử dụng biến x bên ngoài hàm sẽ gây ra lỗi  
echo "<p>Giá trị biến x là: $x</p>";
```

Phạm vi local

- Khi muốn các biến local không bị hủy đi sau khi kết thúc hàm, ta dùng từ khóa “static” lúc khai báo biến.

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

|| 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



1.3.5 HÀNG (CONSTANTS)

Hằng

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

Khai báo hằng

define(name, value, case-insensitive)

- name: tên hằng
- value: giá trị của hằng
- case-insensitive: mang giá trị true/false, dùng để xác định xem tên hằng có cần phân biệt chữ hoa hay thường hay không. Giá trị mặc định là false nghĩa là có phân biệt chữ hoa và chữ thường. Nếu giá trị là true nghĩa là không phân biệt chữ hoa và chữ thường.

Khai báo hằng

- Ví dụ:

```
define("GREETING", "Welcome to FIT-TDC!", true);
function myTest() {
    echo GREETING;
}
myTest();
```

1.3.6 **KIỂU DỮ LIỆU**

||| Kiểu dữ liệu trong PHP

- **String**: kiểu chuỗi ký tự. Một chuỗi phải được đặt trong dấu ' hoặc " (Xem thêm ở bài 3)
- **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 (xem thêm ở bài 2)
- **NULL**: kiểu NULL chỉ có duy nhất một giá trị là NULL, mang ý nghĩa là giá trị rỗng (không có giá trị). Nếu một biến được tạo ra mà không được gán giá trị, nó sẽ mang giá trị NULL. Giá trị của biến có thể được xóa bằng cách gán giá trị NULL cho biến đó.
- **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.

1.3.7 **HÀM**

Hàm

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

```
function functionName($arg1, $arg2 = defaultValue) {  
    //Nội dung hàm  
}
```

- Tên hàm chỉ được bắt đầu bằng ký tự hoặc dấu _ (không được bắt đầu bằng số). Thông thường tên hàm được viết theo kiểu camelCase.

Hàm – Tham số

- Tên hàm chỉ được bắt đầu bằng ký tự hoặc dấu _ (không được bắt đầu bằng số). Thông thường tên hàm được viết theo kiểu camelCase.
- Các tham số cách nhau bằng dấu phẩy, được viết giống như quy tắc viết biến trong PHP (Nhắc lại quy tắc viết tên biến?)
- Để truyền tham chiếu, ta đặt thêm dấu & trước tham số.

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 bên trái, các tham số có giá trị mặc định sẽ được đặt bên phải.

```
function functionName($arg1, $arg2, $arg3 = defaultValue, $arg4 = defaultValue) {  
    //Nội dung hàm  
}
```



1.3.8 **BIỂU THỨC ĐIỀU KIỆN**

||| 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.



1.3.9 **VÒNG LẶP**

||| Vòng lặp

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

- while
- do...while
- for

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

||| Vòng lặp foreach

Foreach là vòng lặp chuyên sử dụng cho mảng.

- Vòng lặp foreach cho mảng thường:

```
foreach ($array as $value) {  
    //Nội dung lặp  
}
```

- Vòng lặp foreach cho mảng associative:
(xem bài 3)

```
foreach ($array as $key => $value) {  
    //Nội dung lặp  
}
```



1.3.10
TOÁN TỬ

Toán tử toán học

Operator	Name	Example	Result
+	Addition	$$x + y	Sum of \$x and \$y
-	Subtraction	$$x - y	Difference of \$x and \$y
*	Multiplication	$$x * y	Product of \$x and \$y
/	Division	$$x / y	Quotient of \$x and \$y
%	Modulus	$$x \% y	Remainder of \$x divided by \$y
**	Exponentiation	$$x ** y	Result of raising \$x to the \$y'th power (Introduced in PHP 5.6)

||| Toán tử gán

Assignment	Same as...	Description
$x = y$	$x = y$	The left operand gets set to the value of the expression on the right
$x += y$	$x = x + y$	Addition
$x -= y$	$x = x - y$	Subtraction
$x *= y$	$x = x * y$	Multiplication
$x /= y$	$x = x / y$	Division
$x \%= y$	$x = x \% y$	Modulus

||| Toán tử chuỗi

Operator Name	Example	Result
.	Concatenation <code>\$txt1 = "Hello" \$txt2 = \$txt1 . " world!"</code>	Now \$txt2 contains "Hello world!"
.=	Concatenation assignment <code>\$txt1 = "Hello" \$txt1 .= " world!"</code>	Now \$txt1 contains "Hello world!"

Toán tử tăng/giảm

Operator	Name	Description
<code>++\$x</code>	Pre-increment	Increments \$x by one, then returns \$x
<code>\$x++</code>	Post-increment	Returns \$x, then increments \$x by one
<code>--\$x</code>	Pre-decrement	Decrements \$x by one, then returns \$x
<code>\$x--</code>	Post-decrement	Returns \$x, then decrements \$x by one

Toán tử so sánh

Operator	Name	Example	Result
<code>==</code>	Equal	<code>\$x == \$y</code>	True if <code>\$x</code> is equal to <code>\$y</code>
<code>=====</code>	Identical	<code>\$x ===== \$y</code>	True if <code>\$x</code> is equal to <code>\$y</code> , and they are of the same type
<code>!=</code>	Not equal	<code>\$x != \$y</code>	True if <code>\$x</code> is not equal to <code>\$y</code>
<code><></code>	Not equal	<code>\$x <> \$y</code>	True if <code>\$x</code> is not equal to <code>\$y</code>
<code>!==</code>	Not identical	<code>\$x !== \$y</code>	True if <code>\$x</code> is not equal to <code>\$y</code> , or they are not of the same type
<code>></code>	Greater than	<code>\$x > \$y</code>	True if <code>\$x</code> is greater than <code>\$y</code>
<code><</code>	Less than	<code>\$x < \$y</code>	True if <code>\$x</code> is less than <code>\$y</code>
<code>>=</code>	Greater than or equal to	<code>\$x >= \$y</code>	True if <code>\$x</code> is greater than or equal to <code>\$y</code>
<code><=</code>	Less than or equal to	<code>\$x <= \$y</code>	True if <code>\$x</code> is less than or equal to <code>\$y</code>

Toán tử logic

Operator	Name	Example	Result
and	And	<code>\$x and \$y</code>	True if both \$x and \$y are true
or	Or	<code>\$x or \$y</code>	True if either \$x or \$y is true
xor	Xor	<code>\$x xor \$y</code>	True if either \$x or \$y is true, but not both
&&	And	<code>\$x && \$y</code>	True if both \$x and \$y are true
	Or	<code>\$x \$y</code>	True if either \$x or \$y is true
!	Not	<code>!\$x</code>	True if \$x is not true

Toán tử mảng

Operator	Name	Example	Result
+	Union	<code>\$x + \$y</code>	Union of <code>\$x</code> and <code>\$y</code> (but duplicate keys are not overwritten)
<code>==</code>	Equality	<code>\$x == \$y</code>	True if <code>\$x</code> and <code>\$y</code> have the same key/value pairs
<code>====</code>	Identity	<code>\$x === \$y</code>	True if <code>\$x</code> and <code>\$y</code> have the same key/value pairs in the same order and of the same types
<code>!=</code>	Inequality	<code>\$x != \$y</code>	True if <code>\$x</code> is not equal to <code>\$y</code>
<code><></code>	Inequality	<code>\$x <> \$y</code>	True if <code>\$x</code> is not equal to <code>\$y</code>
<code>!==</code>	Non-identity	<code>\$x !== \$y</code>	True if <code>\$x</code> is not identical to <code>\$y</code>



1.3.11

NHẬN DỮ LIỆU TỪ FORM

||| 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"]`

||| Nhận dữ liệu từ form

- Ví dụ, lấy dữ liệu của `<input type="text" name="email">` nằm trong form có `method="get"` và in ra màn hình:

```
echo $_GET["email"];
```

- Tương tự ví dụ trên, nếu form có `method="post"`:

```
echo $_POST["email"];
```

- Thông thường, các developer thường dùng post để truyền dữ liệu từ form.

||| 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