

ÔN TẬP PHP

(Ty Tổ Trưởng)

1. Kiến thức cơ bản.

- Đoạn mã php có thể đặt bất kỳ nơi nào trong trang, kể cả HTML.
 - o Đoạn mã bắt đầu : `<?php //code php ?>`
- Trong một file.php có biên dịch được cả html, css, javascript, nhưng không được biên dịch được khi code trong Đoạn mã php mà chỉ lợi dụng câu lệnh echo để xuất.
- Một câu lệnh được kết thúc bởi dấu chấm phẩy.
- Các từ khóa như: `class`, `echo`, `function...` khi viết sẽ không phân biệt hoa thường.
- Các biến, tên phương thức (hàm) có phân biệt chữ hoa và thường.

2. Cú pháp .

- Xuất : `echo` "Hello World" hoặc `print`("Hello World");
- Ghi chú: `//`, `/**/`.
- Câu lệnh import mã nguồn từ file khác: ta có thể sử dụng mã nguồn của các file khác bằng câu lệnh `include`, `require`. Hai câu lệnh này tương đối giống nhau và đều là sao chép mã nguồn. Nhưng khi trường hợp import 2 lần `include` sẽ chỉ nhắc nhở bằng một cảnh báo, còn `require` sẽ thông báo lỗi.
- Cú pháp nối chuỗi: `echo` "Hello World"."!"
- Hàm `var_dump($para1, $para2,...)`
 - o Hàm này sẽ in ra thông tin cũng như kiểu dữ liệu của đối số:

```
VD<?php

$a = [
    "php",
    "css",
    "html"
];

$b = "hello.net";

$c = 4.58;
echo "<pre>";
var_dump($a, $b, $c);
```

```
echo "<pre>";
```

```
?>
```

Kết quả:

```
array(3) {  
    [0]=> string(3) "php"  
    [1]=> string(3) "css"  
    [2]=> string(4) "html"  
}  
string(12) "hello.net"  
float(4.58)
```

- Khai báo biến: Bắt đầu bằng dấu \$, và nó sẽ tự động ép kiểu dữ liệu .
 - o VD: \$variable = 5 (**integer**);
 - o Biến có 3 tầm vực là **local**, **global**, **static**.
 - **Local:**
 - Một biến được khai báo chỉ bên trong hàm thì phạm vi sử dụng của nó chỉ bên trong hàm.
 - **GloBal:**
 - Một biến được khai báo bên ngoài và bên trong hàm , biến được khai báo tầm vực global thì biến đó có thể truy xuất từ bên ngoài hàm.

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

Lúc này hàm MyTest sẽ hiểu \$x và \$y là biến toàn cục ở trên.

- **Static:** Thông thường khi chạy xong hàm thì các biến ở trong hàm được giải phóng. Tuy nhiên trong một số trường hợp chúng ta muốn dữ liệu giá trị của biến thì chúng ta sử dụng khai báo biến trong hàm dạng static.

```

VD: <?php
function myFunction() {
    static $x = 0;
    echo $x;
    $x++;
}

myFunction ();
myFunction ();

?>

```

- Như vậy sau khi gọi hàm thì giá trị của nó sẽ được bảo lưu lại cho lần sau khi chúng ta lại tiếp tục gọi hàm đó -> **Chú ý biến chỉ có giá trị trong hàm.**
- Hằng số: để khai báo một hằng số ta khai báo như sau.
 - **Define**(‘tên hằng’, ‘giá trị hằng’) -- > **Define**(“PI”, 3.14159);
 - **Echo** PI; -> khi gọi hằng sẽ không có dấu \$.
- **If else, for, while** tương tự c++.
- Cách hoạt động của mảng: Mảng hoạt động dựa vào **key** và **value**, khi khai báo mảng tạo giá trị trực tiếp thì tự động key sẽ là 0 -> n.
- Phương thức truyền dữ liệu **POST, GET**:
 - **POST:** Những thông tin truyền qua phương thức POST thì không mọi người khác không thể nhìn thấy và nó không giới hạn số lượng thông tin truyền lên server.
Ngoài ra POST hỗ trợ ở phương thức upload file lên server.
 - **GET:** Khi sử dụng phương thức GET dữ liệu truyền lên server được public mọi người đều nhìn thấy, những biến truyền lên server được hiển thị ngay trên url. Đối với phương thức GET số lượng thông tin gửi được được giới hạn(2000 ký tự). Tuy nhiên những thông tin được truyền trên Url có tác dụng cho bookmark, tốt cho Seo web.
- Các biến hệ thống trong PHP:
 - **\$_GLOBAL:**
 - Biến \$GLOBAL dùng để chứa các biến toàn cục từ bất kỳ nơi nào trong nội dung của php nó có thể là trong hàm hoặc phương thức.
 - Những biến toàn cục trong php được lưu trong một cái mảng gọi \$GLOBAL['index']. Index chính là tên của biến.

```

VD: <?php
$x= 10;
$y = 5;
function total(){
    $z = $GLOBALS['x'] + $GLOBALS['y'];
    echo $z;
}
total();
?>

```

- **\$_SERVER**: Đây là biến lưu trữ thông tin về server đang chạy file hiện hành. Để xem thông tin về các thông số SERVER bạn thực hiện thao tác in mảng `$_SERVER`.
- **\$_POST, \$_GET, \$_FILE**.
- **\$_SESSION**: Session là một khái niệm dùng để lưu lại thông tin của người dùng tác động đến từ đó sử dụng các công việc liên quan đến các hành vi người dùng như đăng nhập, mua hàng, và biến SESSION chỉ hết tác dụng khi tắt trình duyệt và bị hủy bằng hàm `unset()`. Tất cả các giá trị của SESSION được lưu trữ trong biến hệ thống `$_SESSION`.

```

VD <?php
session_start();
$_SESSION['status_login'] = true;
?>

```

- Lưu ý: khi làm việc với biến SESSION chúng ta cần khởi động thông qua hàm `session_start()` đặt ở đầu file đang làm việc.
- **\$_COOKIE**: Cookie là một file dung lượng nhỏ được server nhúng vào máy tính của người dùng. Với cookie ta có thể vừa thiết lập và gọi giá trị của cookie.
 - Cú pháp khai báo: `setcookie($name, $value, $expire, $path, $domain)`, hiện tại mình chỉ sử dụng 4 tham số.
 - \$name: Tên cookie.
 - \$value: giá trị cookie (mặc định chuỗi rỗng).
 - \$expire: Thời gian tồn tại, nếu thiết lập 0 thì sau khi tắt trình duyệt sẽ tự mất. Thời gian tồn tại tối đa 3 năm.
 - \$path: Đường dẫn của server (các file) mà cookie có hiệu lực. Nếu được thiết lập "/" thì có hiệu lực trên tất cả các file trên server.
 - \$domain: Tên miền cookie có hiệu lực.

```

- <?php
- setcookie('is_login', true, time() + 3600, '/');

```

```
- ?>
```

- Xóa COOKIE: Để xóa cookie thì phải thiết lập thời gian sống về quá khứ.

```
- <?php  
- setcookie('is_login', true, time() - 3600, '/');  
- ?>
```