# PHP CƠ BẢN

## 1. Chuỗi (String)

### 1.1. Cấu trúc chuỗi trong php

Chuỗi trong php thường sử dụng dấu nháy kép “”. Ngoài ra trong php còn sử dụng dấu nháy đơn ‘’ để thể hiện chuỗi. Sự khác nhau cơ bản của chuỗi chứa dấu nháy đơn và dấu nháy kép.

- Chuỗi dấu nháy kép xem biến là một biến bình thường và in ra màn hình

- Chuỗi dấu nháy đơn xem biến là một hằng và không in ra màn hình.

<?php

//Sự khác nhau chuỗi dấu nháy kép và nháy đơn

$bien\_chuoi = "name";

$bien\_chuoi\_hang = 'Chuỗi $bien\_chuoi sẽ không được in!\\n';

print($bien\_chuoi\_hang);

print "<br />";

$bien\_chuoi\_hang = "Chuỗi $bien\_chuoi sẽ được in!\\n";

print($bien\_chuoi\_hang);

?>

Chuỗi $bien\_chuoi sẽ không được in!\n

Chuỗi name sẽ được in!\n

### 1.2. Toán tử nối chuỗi

Để nối hai chuỗi trong php sử dụng **toán tử . (dấu chấm)**

<?php

$string1="VietJack";

$string2=" Team";

echo $string1 . " " . $string2;

?>

### 1.3. Hàm tìm độ dài chuỗi

Hàm strlen() trong php được sử dụng để tìm độ dài của chuỗi.

<?php

echo strlen("VietJack Team!");

?>

## 2. Toán tử và biểu thức

### 2.1. Toán tử quan hệ

Toán tử quan hệ là đn trong toán tử và biểu thức được dùng để kiểm tra mối quan hệ giữa hai biến hay giữ một biến và một hằng số.

Bảng mô tả các toán tử quan hệ:

|  |  |
| --- | --- |
| **Toán tử** | **Ý nghĩa** |
| > | Lớn hơn |
| >= | Lớn hơn bằng |
| < | Bé hơn |
| <= | Nhỏ hơn hoặc bằng |
| == | Bằng |
| != | Không bằng |

Toán tử quan hệ === dùng để so sánh giá trị giữa các biến và hằng đúng theo giá trị và kiểu dữ liệu của nó.

Ví dụ: giả sử $a = ‘123’, $b = 123 thì phép $a == $b cho kết quả là true, còn phép $a===$b cho kết quả là false.

## 3. Mảng

Trong PHP mảng được tạo ra bằng từ khóa: array();

Trong đó có 3 kiểu mảng trong php:

- Mảng chỉ số

- Mảng liên hợp

- Mảng nhiều chiều

### 3.1. Độ dài của mảng

Để xác định độ dài của mảng sử dụng hàm count()

<?php  
$cars = array("Volvo", "BMW", "Toyota");

echo count($cars);  
?>

### 3.2. Mảng chỉ số

Có 2 phương pháp tạo mảng chỉ số:

Phương pháp 1 tạo mảng tự động:

$cars = array("Volvo", "BMW", "Toyota");

Phương pháp 2 gán chỉ số:

$cars[0] = "Volvo";

$cars[1] = "BMW";

$cars[2] = "Toyota";

In mảng theo vòng lặp:

<?php  
$cars = array("Volvo", "BMW", "Toyota");

$arrlength = count($cars);

for($x = 0; $x < $arrlength; $x++) {

echo $cars[$x];

echo "<br>";  
}  
?>

### 3.3. Mảng liên hợp

Mảng liên hợp sử dụng cặp key và value

Để tạo mảng liên hợp có 2 phương pháp sau đây:

Phương pháp 1:

$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");

Phương pháp 2:

$age['Peter'] = "35";

$age['Ben'] = "37";

$age['Joe'] = "43";

In mảng theo vòng lặp sử dụng foreach:

<?php  
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");

foreach($age as $x => $x\_value) {

echo "Key=" . $x . ", Value=" . $x\_value;

echo "<br>";  
}  
?>

### 3.4. Mảng nhiều chiều

a) Mảng 2 chiều

Mảng 2 chiều là mảng của mảng

$cars = array (

array("Volvo",22,18),

array("BMW",15,13),

array("Saab",5,2),

array("Land Rover",17,15)

);

Sử dụng vòng lặp For để lấy phần tử mảng $cars

<?php  
for ($row = 0; $row < 4; $row++) {

echo "<p><b>Row number $row</b></p>";

echo "<ul>";

for ($col = 0; $col < 3; $col++) {

echo "<li>".$cars[$row][$col]."</li>";

}

echo "</ul>";  
}  
?>

### 3.5. Xóa một phần tử trong mảng

Để xóa một phần tử trong mảng sử dụng từ khóa unset($array[3])

<?php

//Xóa một phần tử trong mảng

$arr = array(1, 2, 3, 4, 5);

unset($arr[2]);

foreach($arr as $a){

    echo $a." ";

}

?>

## 4. Bài Tập Date Time

### 4.1. Chuyển đổi kiểu dữ liệu từ chuỗi sang date time

<html>

<head>

<title>Chuyển đổi định dạng ngày tháng trong PHP</title>

</head>

<body>

<?php

$dinh\_dang\_cu = "2016-09-29";

$dinh\_dang\_moi = date("d-m-Y", strtotime($dinh\_dang\_cu));

echo "Định dạng cũ: " .$dinh\_dang\_cu."<br>";

echo "Định dạng mới: " .$dinh\_dang\_moi."<br>";

?>

</body>

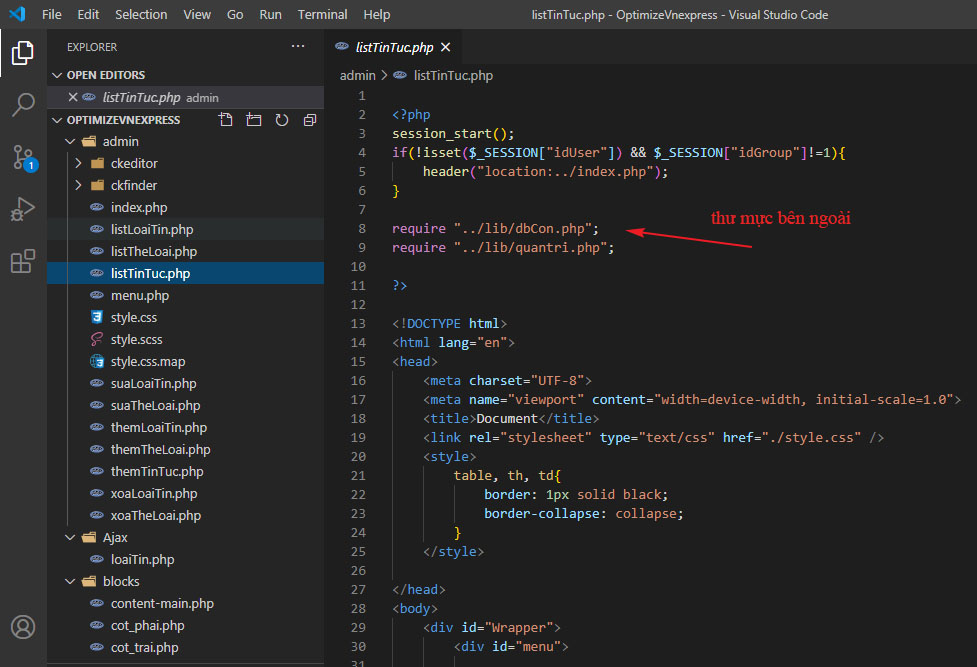
</html>

Định dạng cũ: 2016-09-29

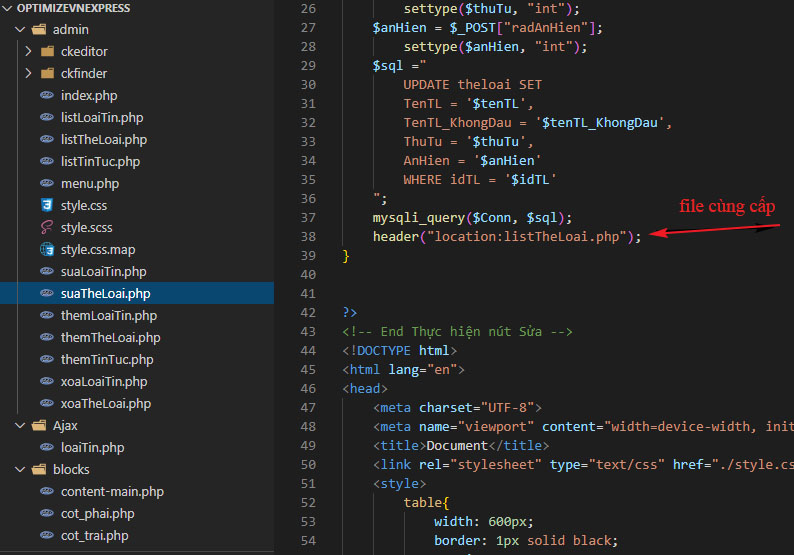
Định dạng mới: 29-09-2016

## 5. Đường dẫn thư mục

Đối với các thư mục ở ngoài cùng khi truy cập vào đường dẫn sử dụng



Đường dẫn thư mục cùng cấp



# PHP NÂNG CAO

## 1. Lập trình hướng đối tượng

### 1.1. Đặc điểm lập trình hướng đối tượng

Tập trung vào dữ liệu thay cho các hàm

Chương trình được chia thành các đối tượng độc lập

Cấu trúc dữ liệu đưuọc thiết kế sao cho đặc tả được các đối tượng

Dữ liệu được che dấu, bao bọc

Các đối tượng trao đổi với nhau thông qua các hàm

Chương trình được thiết kế theo hướng tiếp cận từ dưới lên

### 1.2. Ưu điểm

Loại bỏ được những dư thừa, trùng lặp trong việc xây dựng ứng dụng

Giảm kích thước, thời gian xử lý, thời gian phát triển hệ thống, tăng năng suất lao động

### 1.3. Các tính chất cơ bản của lập trình hướng đối tượng

a) Trừ tượng hóa

Là quá trình đơn giản hóa một đối tượng, là việc xác định những thuộc tính, phương thức cần thiết cho một chương trình.

b) Thừa kế

Là kỹ thuật cho phép lớp này có thể thừa kế các phương thức và thuộc tính của lớp khác

c) Đa hình

Là kỹ thuật cho phép lớp này có thể viết lại các thuộc tính hay phương thức của lớp khác.

d) đóng gói

là tích chất không cho phép người dùng hay đối tượng khác thay đổi dữ liệu thành viên của đối tượng nội tại. Chỉ có các thành viên trong đối tượng đó mới được phép thay đổi.

### 1.4. Khai báo và sử dụng class trong PHP

Khai báo|:

class ten\_class{

    //Các thuộc tính, phương thức

    //Sử dụng từ khóa $this->thuộc tính, phương thức

}

Sử dụng: $đối\_tuong = new ten\_class;

        //Sử dụng $đối\_tuong->thuộc tính, phương thức trong class

Các cơ chế:

- public: các đối tượng từ bên ngoài class có thể truy cập, hỗ trợ sự thừa kế.

- protected: các đối tượng từ bên ngoài class không thể truy cập, hỗ trợ sự thừa kế

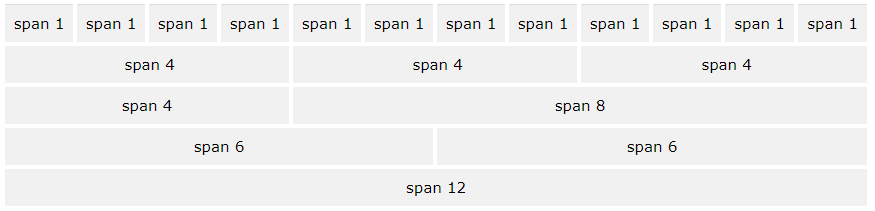
- private: các đối tượng từ bên ngoài class không thể truy cập, không hỗ trợ sự thừa kế.

- set, get: cho phép các đối tượng bên ngoài class truy cập, xử lý các cơ chế private và protected.

# BOOTSTRAP

## 1. Cấu trúc grid

Hệ thống lưới của bootstrap chia trang web ra thành 12 cột. Tùy vào mục đích sử dụng mà có thể chia theo ý tưởng của lập trình viên.



Lớp grid chia thành 4 lớp sau:

- xs (cho điện thoại – độ rộng phân giải màn hình nhỏ hơn 768px)

- sm (cho tablet - >=768px)

- md (cho laptop >=992px)

- lg (cho laptop >=1200px)

Cấu trúc bootstrap grid:

<div class="row">  
  <div class="col-\*-\*"></div>  
  <div class="col-\*-\*"></div>  
</div>  
<div class="row">  
  <div class="col-\*-\*"></div>  
  <div class="col-\*-\*"></div>  
  <div class="col-\*-\*"></div>  
</div>  
<div class="row">  
  ...  
</div>

# JAVASCRIPT

## 1. Vòng lặp FOR…IN

Vòng lặp For…in được sử dụng để lặp qua các thuộc tính của một đối tượng.

for (variablename in object){

statement or block to execute

}

VD: sử dụng vòng lặp FOR…IN để in đối tượng Navigator của trình duyệt Web.

   <script type="text/javascript">

        var pro;

        for(pro in navigator){

            document.write(pro+"<br>");

        }

    </script>

## 2. Hàm

Cú pháp của hàm:

<script type="text/javascript">

<!--

function functionname(parameter-list)

{

statements

}

//-->

</script

### 2.1. Hàm không có tham số

Ví dụ: định nghĩa hàm sayHello không có tham số

<script type="text/javascript">

<!--

function sayHello()

{

alert("Hello there");

}

//-->

</script>

Ví dụ 2: Gọi một hàm sau khi định nghĩa

<html>

<head>

<script type="text/javascript">

function sayHello()

{

document.write ("Hello there!");

}

</script>

</head>

<body>

<p>Click the following button to call the function</p>

<form>

<input type="button" onclick="sayHello()" value="Xin chao!">

</form>

<p>Use different text in write method and then try...</p>

</body>

</html>

### 2.2. Hàm có tham số

Hàm được truyền 2 tham số.

<html>

<head>

<script type="text/javascript">

function sayHello(name, age)

{

document.write (name + " is " + age + " years old.");

}

</script>

</head>

<body>

<p>Click the following button to call the function</p>

<form>

<input type="button" onclick="sayHello('Zara', 7)" value="Xin chao!">

</form>

<p>Use different parameters inside the function and then try...</p>

</body>

</html>

## 3. Từ khóa Void

Void là một từ khóa quan trọng trong javascript mà nó có thể sử dụng như một toán tử một ngôi xuất hiện trước toán hạng đơn của nó. Toán tử này không trả về một giá trị nào.

### 3.1. Cú pháp của void

<head>

<script type="text/javascript">

void func()

javascript:void func()

or:

void(func())

javascript:void(func())

</script>

</head>