List Articles CodeWr

# A.PHP

## 1.PHP là gì?

Tilte: PHP là gì?

Map: php-la-gi

Type: PHP

Meta Des: Định nghĩa PHP là gì ?

Meta Keys: PHP là gì , php la gi

Description: PHP là một ngôn ngữ kịch bản máy chủ, và một công cụ mạnh mẽ để làm cho các trang Web năng động và tương tác.Hiện được sử dụng rộng rãi, miễn phí và hiệu quả.

Content:

**PHP là 1 ngôn ngữ lập trình đơn giản và dễ sử dụng nhất**

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
echo "Welcome to PHP!";  
?>  
  
</body>  
</html>

## 2. Giới thiệu về PHP

Tilte: Giới thiệu về PHP

Map: gioi-thieu-ve-php

Type: PHP

Meta Des: Giới thiệu về PHP

Meta Keys: gioi thieu PHP, PHP có đặc điểm gì ?

Description: PHP là một ngôn ngữ được thực hiện trên server

Content:

**Các kiến thức cần biết trước khi học PHP:**

* HTML
* CSS
* JAVASCRIPT

**PHP là gì ?**

* PHP là viết tắt của từ “Hypertext Preprocessor”
* PHP là một ngôn ngữ kịch bản nguồn mở được sử dụng rộng rãi
* Các tập lệnh PHP được thực hiện trên máy chủ
* PHP là miễn phí để tải về và sử dụng

**Tại sao PHP là một ngôn ngữ tuyệt vời và phổ biến!**

* Vì nó đủ mạnh để trở thành cốt lõi của hệ thống viết blog lớn nhất trên web (WordPress)!
* Vì nó đủ chiều sâu để chạy mạng xã hội lớn nhất (Facebook)!
* Vì nó ngôn ngữ server dễ dàng học đối với người mới bắt đầu

**File PHP là gì ?**

* Các tệp PHP có thể chứa văn bản, HTML, CSS, JavaScript và mã PHP
* Mã PHP được thực hiện trên máy chủ, và kết quả được trả lại cho trình duyệt như là HTML đơn giản
* Các tệp PHP có phần mở rộng ".php"

**PHP có thể làm gì ?**

* PHP có thể tạo ra nội dung trang động
* PHP có thể tạo, mở, đọc, viết, xóa, và đóng các tập tin trên máy chủ
* PHP có thể thu thập dữ liệu biểu mẫu
* PHP có thể gửi và nhận cookie
* PHP có thể thêm, xóa, sửa đổi dữ liệu trong cơ sở dữ liệu của bạn
* PHP có thể được sử dụng để kiểm soát người dùng truy cập
* PHP có thể mã hóa dữ liệu

Với PHP bạn không bị giới hạn ở HTML đầu ra. Bạn có thể xuất hình ảnh, file PDF, và thậm chí cả phim Flash. Bạn cũng có thể xuất bất kỳ văn bản nào, chẳng hạn như XHTML và XM

**Tại sao lại chọn PHP?**

* PHP chạy trên các nền tảng khác nhau (Windows, Linux, Unix, Mac OS X, v.v ...)
* PHP tương thích với hầu hết các máy chủ được sử dụng ngày nay (Apache, IIS, ...)
* PHP hỗ trợ một loạt các cơ sở dữ liệu
* PHP là miễn phí. Tải nó từ tài nguyên PHP chính thức: www.php.net
* PHP là dễ dàng để tìm hiểu và chạy hiệu quả ở phía máy chủ

## 3.Cài đặt PHP

Tilte: Cài đặt PHP

Map: cat-dat-php

Type: PHP

Meta Des: Cài đặt PHP

Meta Keys: huong dan cai dat php , hướng dẫn cài đặt php, cai dat php, cài đặt php

Description: Bài viết này sẽ hướng dẫn cài đặt XAMPP để sử dụng PHP trên window

Content:

XAMPP là 1 công cụ để học PHP và Mysql rất tốt .

Để tải XAMPP các bạn vào địa chỉ :  <https://www.apachefriends.org/download.html> và chọn phiên bản phù hợp với window của bạn.

Sau khi cài đặt xong các bạn sẽ tiến hành start XAMPP :

- Vào thư mục C:\xampp\

- Click double vào xampp\_start.exe

- Xuất hiện panel chọn start ở module Apache

Bây giờ bạn có thể làm ví dụ đầu tiền như sau :

B1. Mở notepad tạo 1 file với nội dung sau :

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
echo "Welcome to PHP!";  
?>  
  
</body>  
</html>

B2. Save file bằng tổ hợp phím Ctrl+S or File\save . File Name đặt hello.php, Save as type : All File. Và lưu trong thư mục C:\xampp\htdocs\

B3. Mở trình duyệt lên và nhấn localhost/hello.php. và xem kết quả là Welcome to PHP!

Như vậy là bạn đã chạy được ví dụ đầu tiên của PHP rồi

## 4. Cú pháp PHP

Tilte: Cú pháp PHP

Map: cu-phap-php

Type: PHP

Meta Des: cú pháp cơ bản của php

Meta Keys: cu phap php, cú pháp cơ bản php, cú pháp php, cu php co ban php

Description: Một tập lệnh PHP được thực hiện trên máy chủ, và kết quả HTML được gửi lại cho trình duyệt.

Content:

**Cú pháp cơ bản PHP**

- Một tập lệnh PHP có thể được đặt ở bất cứ đâu trong tài liệu.

- Một lệnh PHP bắt đầu bằng **<?php** và kết thúc bằng **?>**:

**<? php**

/ / các lệnh PHP

**?>**

- Phần mở rộng file PHP là ".php".

- Một file PHP thường chứa các thẻ HTML, và một số mã lệnh PHP.

- Kết thúc mỗi câu lệnh trong PHP bằng dấu ;

Ví dụ cơ bản của PHP sử dụng lệnh echo để xuất ra Hello Wolrd:

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

**Chú thích trong PHP**

Một chú thích trong mã PHP là một dòng giải thích chứ không được thực thi trong chương trình. Nó chỉ giải thích cho người xem mã php.

Chú thích được sử dụng để:

* Chỉ cho người khác hiểu bạn đang làm gì .
* Ghi nhớ cho người lập trình hàm này làm gì ? lệnh này để làm gì?....

Ví dụ về các loại chú thích trong PHP:

<!DOCTYPE html>  
 <html>  
 <body>  
  
 <?php  
 // chú thích trên 1 dòng  
  
 # chú thích trên 1 dòng dùng tương tự //  
  
 /\*  
 chú thích trên nhiều dòng hay là 1 đoạn code

\*/  
  
 // chú thích ngay trên lệnh đang thực thi  
 $x = 5 /\* + 15 \*/ + 5;  
 echo $x;  
 ?>  
  
 </body>  
 </html>

**Một số chú ý cơ bản trong PHP**

Trong PHP tất cả các keyword (echo, if, else, while …..) không phân biệt hoa thường

Ví dụ:

<!DOCTYPE html>  
 <html>  
 <body>  
  
 <?php  
 ECHO "Hello World!<br>";  
 echo "Hello World!<br>";  
 EcHo "Hello World!<br>";  
 ?>  
  
 </body>  
</html>

Kết quả trả là:

Hello World!  
Hello World!  
Hello World!

Nhưng các biến trong PHP lại phân biệt chữ hoa chữ thường

Ví dụ:

<!DOCTYPE html>  
 <html>  
 <body>  
  
 <?php  
 $object = "you";  
 echo "I love " . $object. "<br>";  
 echo "he love " . $Object. "<br>";  
 echo "she love " . $OBJECT . "<br>";  
 ?>  
  
 </body>  
</html>

Kết quả trả ra:

I love you

He love

She love

## 5. Biến trong PHP

Tilte: Biến trong PHP

Map: bien-trong-php

Type:PHP

Meta Des: Quy định biến trong PHP

Meta Keys: biến trong php, bien trong php, quy tắc tên biến trong php, quy tac bien trong php

Description: Biến là một đối tượng dùng để lưu trữ thông tin

Content:

**Để tạo 1 biến trong PHP**

Cú pháp: $tenbien

Ví dụ:

<!DOCTYPE html>  
 <html>  
 <body>  
 <?php  
 $txt = "Hello world!";  
 $x = 5;  
 $y = 10.5;  
 echo $txt;  
 echo "<br>";  
 echo $x;  
 echo "<br>";  
 echo $y;  
 ?>  
  
 </body>  
</html>

Kết quả :

Hello world!  
5  
10.5

Các quy tắc đặt tên biến PHP

* Một biến bắt đầu với dấu $, theo sau là tên của biến
* Tên biến phải bắt đầu bằng một chữ cái hoặc ký tự gạch dưới
* Tên biến không thể bắt đầu bằng một số
* Tên biến chỉ có thể chứa các ký tự chữ và số và dấu gạch dưới (A-z, 0-9, và \_)
* Tên biến được phân biệt chữ hoa chữ thường ($age và $AGE là hai biến khác nhau)

**In giá trị của biến**

Lệnh echo trong PHP thường được sử dụng để xuất dữ liệu ra màn hình.

Ví dụ sau sẽ hiển thị cách xuất văn bản và biến:

<!DOCTYPE html>  
 <html>  
 <body>  
  
 <?php  
 $txt = "CodeWR.com";  
 echo "I love $txt!";  
 ?>  
  
 </body>  
</html>

Kết quả

I love CodeWR.com!

Ví dụ cho kết quả tương tự nhưng sử dụng nối chuỗi bằng dấu chấm (.)

<!DOCTYPE html>  
 <html>  
 <body>  
  
 <?php  
 $txt = "CodeWR.com";  
 echo "I love . $txt . "!";  
 ?>  
  
 </body>  
</html>

Kết quả

I love CodeWR.com!

Ví dụ về tổng 2 biến

<!DOCTYPE html>  
 <html>  
 <body>  
 <?php  
 $a = 5;  
 $b = 4;  
 echo $a + $b;  
 ?>  
</body>  
</html>

Kết quả

9

PHP là một ngôn ngữ mà không cần khai báo kiểu của biến. Trong PHP các biến sẽ tự động chuyển đổi kiểu dữ liệu phù hợp khi ta gắng giá trị cho nó. Đây là 1 đặc điểm khác với các loại ngôn ngữ khác như C , C++ , JAVA …

**Phạm vị biến trong PHP**

Có loại biến được định nghĩa khác nhau:

- global (biến toàn cục)

- local (biến địa phương)

-static (biến tĩnh)

Ví dụ 1:

<!DOCTYPE html>  
 <html>  
 <body>  
  
 <?php  
 $x = 5; //biến global   
   
 function test() {  
    // sử dụng x bên trong hàm này sẽ gây ra lỗi  
    echo "<p>Variable x inside function is: $x</p>";  
 }   
 test();  
  
 echo "<p>Variable x outside function is: $x</p>";  
?>  
  
</body>  
</html>

Kết quả

Variable x inside function is:

Variable x outside function is: 5

Ví dụ 2:

<!DOCTYPE html>  
<html>  
<body>  
<?php  
function test() {  
    $x = 5; // biến local  
    echo "<p>Variable x inside function is: $x</p>";  
}   
test();  
// sử dụng x bên ngoài function test sẽ gây lỗi  
echo "<p>Variable x outside function is: $x</p>";  
?>  
</body>  
</html>

Kết quả :

Variable x inside function is: 5

Variable x outside function is:

**Từ khóa global**

Từ khóa global được dùng để khai báo biến global để sử dụng trong các function

Ví dụ 3:

<!DOCTYPE html>  
 <html>  
 <body>  
  
 <?php  
 $x = 5;  
 $y = 10;  
  
 function test() {  
    global $x, $y;  
    $y = $x + $y;  
 }   
 test();  // gọi function  
 echo $y; // kết quả sẽ trả về giá trị mới của $y  
 ?>  
  
</body>  
</html>

Kết quả:

15

Mỗi biến gloal trong PHP sẽ được lưu trong một mảng được chỉ định theo cú pháp $GLOBAL[index]. Nên có thể sử dụng để gọi biến trong các hàm

Ví dụ 4 :

<!DOCTYPE html>  
 <html>  
 <body>  
 <?php  
 $x = 5;  
 $y = 10;  
 function test() {  
    $GLOBALS['y'] = $GLOBALS['x'] + $GLOBALS['y'];  
 }   
  
 test();  
 echo $y;  
 ?>  
  
 </body>  
</html>

Kết quả:

15

**Từ khóa Static**

Bình thường khi 1 function thực thị xong thì các biến trong function sẽ bị xóa. Để giữ lại biến đó không bị xóa thì ta sẽ sử dụng từ khóa static

Ví dụ

<!DOCTYPE html>  
 <html>  
 <body>  
 <?php  
 function test() {  
     static $x = 0;  
     echo $x;  
     $x++;  
 }  
  
 test();  
 echo "<br>";  
 test();  
 echo "<br>";  
 test();  
 ?>   
  
</body>  
</html>

Kết quả:

0

1

2

## 6. Lệnh echo và print trong PHP

Tilte: Lệnh echo và print trong PHP

Map: lenh-echo-va-print trong PHP

Type:PHP

Meta Des: Hướng dẫn sử dụng lệnh echo và print trong PHP

Meta Keys: lenh echo ,lenh print ,php,lệnh echo , lệnh print

Description: Trong php có 2 lệnh để in kết quả ra màn hình là echo và print

Content:

**So sánh echo và print**

Cả 2 lệnh này trong php đều trả kết quả ra màn hình nhưng có 1 vài điểm khác nhau 1 chút:

* Lệnh echo không trả về giá trị, còn print thì có thể trả về 1 để sử dụng trong biểu thức
* Lệnh echo có thể chứ nhiều tham số, còn print chỉ có 1 tham số
* Lệnh echo nhanh hơn 1 chút so với lệnh print

**Lệnh echo**

Cú pháp : echo hoặc echo()

Ví dụ 1 hiển thị text:

<!DOCTYPE html>  
 <html>  
 <body>  
  
 <?php  
 echo "<h2>PHP is Fun!</h2>";  
 echo "Hello world!<br>";  
 echo "I ", "love ", "PHP.";  
 ?>   
  
</body>  
</html>

Kết quả:

**PHP is Fun!**

Hello world!

I love PHP.

Ví dụ 2 hiển thị biến :

<!DOCTYPE html>  
 <html>  
 <body>  
  
 <?php  
 $txt1 = "Learn PHP";  
 $txt2 = "CodeWR.com";  
 $x = 5;  
 $y = 4;  
  
 echo "<h2>" . $txt1 . "</h2>";  
 echo "Study PHP at " . $txt2 . "<br>";  
 echo $x + $y;  
 ?>  
  
</body>  
</html>

Kết quả:

**Learn PHP**

Study PHP at CodeWR.com  
9

**Lệnh print**

Cú pháp : print hoặc print()

Ví dụ hiển thị text:

<!DOCTYPE html>  
 <html>  
 <body>  
 <?php  
 print "<h2>PHP is Fun!</h2>";  
 print "Hello world!<br>";  
 print "I'm about to learn PHP!";  
 ?>   
  
 </body>  
 </html>

Kết quả:

PHP is Fun!

Hello world!  
I'm about to learn PHP!

Ví vụ hiển thị biến :

<!DOCTYPE html>  
<html>  
 <body>  
  
 <?php  
 $txt1 = "Learn PHP";  
 $txt2 = "CodeWR.com";  
 $x = 5;  
 $y = 4;  
  
 print "<h2>" . $txt1 . "</h2>";  
 print "Study PHP at " . $txt2 . "<br>";  
 print $x + $y;  
?>  
  
</body>  
</html>

Kết quả:

**Learn PHP**

Study PHP at CodeWR.com  
9

## 7.Các kiểu dữ liệu trong PHP

Tilte: Các kiểu dữ liệu trong PHP

Map: cac-kieu-du-lieu-trong-php

Type:PHP

Meta Des: Các kiểu dữ liệu trong php

Meta Keys: kieu du lieu trong php, kiểu dữ liệu trong php ,integer ,string ,float,boolean, array, object, null, resource

Description: Các biến có thể lưu trữ trên các dữ liệu khác nhau. Mỗi kiểu dữ liệu khác nhau thì được xử lý khác nhau.

Content:

**Các kiểu dữ liệu trong PHP**

* String
* Integer
* Float (và Double được sử dụng giống nhau)
* Boolean
* Array
* Object
* NULL
* Resource

**Kiểu String**

Kiểu string là kiểu chuỗi kí tự được sử dụng trong dấu ngoặc dơn hoặc kép.

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php   
$x = "Hello world!";  
$y = 'Hello world!';  
  
echo $x;  
echo "<br>";   
echo $y;  
?>  
  
</body>  
</html>

Kết quả :

Hello world!

Hello world!

**Kiểu Integer**

Kiểu Integer là số không phải số thập phân có giá trị -2,1448,483,648 - 2,147,483,647.

Một số quy tắc của kiểu integer:

* Số nguyên phải có ít nhất một chữ số
* Một số nguyên không được có điểm thập phân
* Một số nguyên có thể là dương hoặc âm
* Các số nguyên có thể được xác định theo ba định dạng: thập phân ( 10), hệ thập lục phân (0-F) hoặc bát phân (0-7)

Trong ví dụ sau $ x là một số nguyên. Hàm var\_dump () của PHP trả về kiểu dữ liệu và giá trị:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php    
$x = 5985;  
var\_dump($x);  
?>    
  
</body>  
</html>

Kết quả:

int(5985)

**Kiểu Float**

Một float (floating point number) là một số với một số thập phân hoặc một số dạng số mũ.

Trong ví dụ sau, $ x là một float. Hàm var\_dump () của PHP trả về kiểu dữ liệu và giá trị:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php    
$x = 10.365;  
var\_dump($x);  
?>    
  
</body>  
</html>

Kết quả:

float(10.365)

**Kiểu Boolean**

Kiểu Boolean có 2 giá trị : true or false

$x = true;  
$y = false;

**Kiểu Array**

Kiểu Array lưu trữ nhiều giá trị vào 1 mảng

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php    
$cars = array("Volvo","BMW","Toyota");  
var\_dump($cars);  
?>    
  
</body>  
</html>

Kết quả :

array(3) { [0]=> string(5) "Volvo" [1]=> string(3) "BMW" [2]=> string(6) "Toyota" }

**Kiểu Object**

Kiểu Object là kiểu lưu trữ thông tin theo hướng đối tượng.

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
class Car {  
    function Car() {  
        $this->model = "VW";  
    }  
}  
// create an object  
$herbie = new Car();  
  
// show object properties  
echo $herbie->model;  
?>  
  
</body>  
</html>

Kết quả

VW

**Kiểu Null**

Null là một kiểu dữ liệu đặc biệt chỉ có thể có một giá trị: NULL.

Một biến kiểu dữ liệu NULL là một biến không có giá trị được gán cho nó.

Nếu một biến được tạo ra mà không có một giá trị, nó sẽ tự động được gán một giá trị NULL.

Các biến cũng có thể được làm trống bằng cách đặt giá trị NULL:

Ví dụ :

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$x = "Hello world!";  
$x = null;  
var\_dump($x);  
?>  
  
</body>  
</html>

Kết quả:

NULL

**Kiểu Resource**

Kiểu Resource là 1 kiểu đặc biệt. Liên quan đến database

## 8. Các hàm xử lý String phổ biến

Tilte: Các hàm xử lý String

Map: cac-ham-xu-ly-string

Type:PHP

Meta Des: Các hàm xử lý string trong php

Meta Keys: ham xu ly string trong php, hàm xử lý string php

Description: Phần này sẽ giới thiệu các hàm xử lý string của php hỗ trợ

Content:

**Hàm strlen() lấy length của string**

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
<?php  
echo strlen("Hello world!");  
?>   
</body>  
</html>

Kết quả:

12

**Hàm str\_word\_count() đếm số từ trong 1 câu**

Ví dụ :

<!DOCTYPE html>  
<html>  
<body>  
<?php  
echo str\_word\_count("Hello world!");  
?>   
</body>  
</html>

Kết quả:

2

**Hàm strrev() đảo ngược string**

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
<?php  
echo strrev("Hello world!");  
?>   
</body>  
</html>

**Hàm strops() lấy vị trí của từ trong chuỗi**

Ví dụ

<!DOCTYPE html>  
<html>  
<body>  
<?php  
echo strpos("Hello world!", "world");  
?>   
</body>  
</html>

Kết quả:

6

**Hàm str\_replace thay thế kí tự trong chuỗi**

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
<?php  
echo str\_replace("world", "Dolly", "Hello world!");  
?>   
</body>  
</html>

Kết quả :

Hello Dolly

## 9.Hằng số trong PHP

Tilte: Hằng số trong PHP

Map: hang-so-trong-php

Type:PHP

Meta Des: Hằng số trong PHP

Meta Keys: hằng số trong php, hang so trong php,

Description: Hằng số là biến có giá trị xác định và không thay đổi được

Content:

**Khai báo hằng**

Cú pháp : define(*name*, *value*, *case-insensitive*)

- name: tên hằng

- value : giá trị hằng

- case-insensitive: mặc định fasle là phân biệt hoa với thường.

Ví dụ :

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
// case-sensitive constant name  
define("Hi", "Welcome to CodeWR.com!");  
echo Hi;  
?>   
  
</body>  
</html>

Kết quả

Welcome to CodeWR.com!

Ví dụ 2:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
// case-insensitive constant name  
define("Hi", "Welcome to CodeWR.com!", true);  
echo hi;  
?>   
  
</body>  
</html>

Kết quả:

Welcome to CodeWR.com!

Hằng được xem là biến toàn cục.

Ví dụ 3:

<!DOCTYPE html>  
<html>  
<body>  
<?php  
define("Hi", "Welcome to CodeWR.com!");  
function test() {  
    echo Hi;  
}  
test();  
?>   
</body>  
</html>

Kết quả

Welcome to CodeWR.com!

## 10.Các toán tử trong PHP

Tilte: Các toán tử trong PHP

Map: cac-toan-tu-trong-php

Type:PHP

Meta Des: Các toán tử trong PHP

Meta Keys: toan tu trong php, toán tử trong php

Description: Toán tử được sử dụng để xử lý giá trị của biến

Content:

**PHP hỗ trợ các loại toán tử**

* Toán sử số học
* Toán tử gán
* Toán tử so sánh
* Toán tử tăng giảm
* Toán tử logic
* Toán tử chuỗi
* Toán tử mảng

**Toán tử số học**

PHP cũng cấp cái loại toán tử số học: +, - ,\* , / ,%

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
<?php  
$x = 10;    
$y = 6;  
echo $x + $y;

echo "<br>";

echo $x - $y;

echo "<br>";

echo $x \* $y;

echo "<br>";

echo $x / $y;

echo "<br>";

echo $x % $y;

echo "<br>";   
?>    
</body>  
</html>

Kết quả

16

4

60

1.666667

4

**Toán tử gán**

PHP hỗ trợ các toán tử gán : =, +=,…

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$x = 10;    
echo $x;

echo "<br>";

$x += 100;

echo $x;

echo "<br>";

$x -= 30;

echo $x;

echo "<br>";

$y=3;

echo $x \* $y;

echo "<br>";

$x /= 5;

echo $x;

echo "<br>";

$x %= 4;

echo $x;

?>    
  
</body>  
</html>

Kết quả :

10

110

80

240

16

0

**Toán tử so sánh**

PHP hỗ trợ các toán tử: ==, ===, != ….

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$x = 100;    
$y = "100";  
  
var\_dump($x == $y); // returns true because values are equal

var\_dump($x === $y);// returns false because types are not equal

var\_dump($x != $y); // returns false because values are equal

var\_dump($x <> $y); // returns false because values are equal

var\_dump($x !== $y); // returns true because types are not equal

$x1 = 100;  
$y1 = 50;  
  
var\_dump($x1 > $y1); // returns true because $x1 is greater than $y1

var\_dump($x1 < $y1); // returns false because $x is less than $y

var\_dump($x >= $y); // returns true because $x is greater than $y

var\_dump($x <= $y); // returns false because $x is less than $y  
?>    
  
</body>  
</html>

**Toán tử tăng giảm**

PHP hỗ trợ các loại : ++,--

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$x = 10;    
echo ++$x;

echo "<br>";

echo $x++;

echo "<br>";

echo --$x;

echo "<br>";

echo $x--;

?>    
  
</body>  
</html>

Kết quả:

11

10

9

10

**Toán tử logic**

PHP hỗ trợ các loại : and, or, xor , &&, || , !

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$x = 100;    
$y = 50;  
  
if ($x == 100 and $y == 50) {  
    echo "logic and<br>";  
}

if ($x == 100 or $y == 80) {  
    echo " logic or<br>";  
}

if ($x == 100 xor $y == 80) {  
    echo " logic xor<br>";  
}

if ($x == 100 && $y == 50) {  
    echo " logic &&<br>";  
}

if ($x == 100 || $y == 80) {  
    echo " logic ||<br>";  
}

if ($x !== 90) {  
    echo " logic ! <br>";  
}  
?>  
?>    
  
</body>  
</html>

Kết quả :

logic and

logic or

logic xor

logic &&

logic ||

logic !

**Toán tử String**

PHP hỗ trợ 2 loại cộng chuỗi: ., .=

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$txt1 = "Hello";  
$txt2 = " world!";

$txt3 = "Hi";  
echo $txt1 . $txt2;

echo "<br>";

$txt3 .= $txt2;  
echo $txt3;  
?>    
  
</body>  
</html>

Kết quả:

Hello world

Hi wordld

**Toán tử Array**

PHP hỗ trợ các loại : +, == , === , != , <>, !==

Ví dụ1 :

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$x = array("a" => "red", "b" => "green");    
$y = array("c" => "blue", "d" => "yellow");    
  
print\_r($x + $y); // union of $x and $y  
?>    
  
</body>  
</html>

Kết quả

Array ( [a] => red [b] => green [c] => blue [d] => yellow )

Ví dụ 2:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$x = array("a" => "red", "b" => "green");    
$y = array("c" => "blue", "d" => "yellow");    
  
var\_dump($x == $y);

var\_dump($x === $y);

var\_dump($x != $y);

var\_dump($x <> $y);

var\_dump($x !== $y);

?>    
  
</body>  
</html>

Kết quả:

bool(false)

bool(false)

bool(true)

bool(true)

bool(true)

## 11. Mệnh đề If…Else…Elseif

Tilte: Mệnh đề If…Else…Elseif

Map: menh-de-if-else-elseif

Type:PHP

Meta Des: Mệnh đề if…else…elseif trong php

Meta Keys: mệnh đề if…else, menh de if-else, menh de if, else trong php

Description: Mệnh đề điều kiện if..else được sử dụng phổ biến trong việc xử lý dữ liệu trong php

Content:

**Mệnh đề IF**

Cú pháp :

if (*điều kiện*) { *//các lệnh sẽ thực hiện nếu điều kiện true*

*lệnh 1;*

*lệnh 2;*

*……*  
}

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$t = date("H");  
  
if ($t < "12") {  
    echo "Good morning!";  
}  
?>  
   
</body>  
</html>

Kết quả:

Good morning! Nếu chơi tới 12 giờ trưa

**Mệnh đề IF-ELSE**

Cú pháp :

if (*điều kiện*) {

*//các lệnh sẽ thực hiện nếu điều kiện true*

*lệnh 1;*

*lệnh 2;*

*……*  
  
} else {  
  *//các lệnh sẽ thực hiện nếu điều kiện false*

*lệnh 3;*

*lệnh 4;*

*……*  
}

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$t = date("H");  
  
if ($t < "12") {  
    echo "Good morning!";  
} else {  
    echo "Good afternoon!";  
}  
?>  
   
</body>  
</html>

Kết quả

Good morning! Nếu là buổi sang

Good afternood! nếu là buổi chiều

**Mệnh đề IF-ELSEIF-ELSE**

Cú pháp:

if (*điều kiện 1*) {  
*//các lệnh sẽ thực hiện nếu điều kiện 1 true*

*lệnh 1;*

*lệnh 2;*

*……*  
} elseif (*điều kiện 2*) {  
*//các lệnh sẽ thực hiện nếu điều kiện 2 true*

*lệnh 1;*

*lệnh 2;*

*……*  
} else {  
    *//các lệnh sẽ thực hiện nếu điều kiện1, điều kiện 2 false*

*lệnh 1;*

*lệnh 2;*

*……*  
}

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$t = date("H");  
if ($t < "12") {  
    echo "Good morning!";  
} elseif ($t < "18") {  
    echo "Good afternoon!";  
} else {  
    echo "Good evening!";  
}  
?>  
   
</body>  
</html>

Kết quả:

Good morning ! Nếu buổi sáng

Good afternoon! Nếu buổi chiều

Good evening ! Nếu buổi tối

## 12.Mệnh đề Switch

Tilte: Mệnh đề Switch

Map: menh-de-switch-trong-php

Type:PHP

Meta Des: Mệnh đề switch trong php

Meta Keys: switch,php, mệnh đề switch , menh de switch

Description: Mệnh đề switch là một dạng của mệnh đề if…elseif….else

Content:

Mệnh đề Switch được sử dụng trong trường hợp có quá nhiều điều kiện

Cú pháp :

switch (*n*) {  
    case *label1:*  
  *//lệnh 1 sẽ được thực hiện nếu n=label1;*

*lệnh 1;*   
        break;  
    case *label2:*  
  *// lệnh 1 sẽ được thực hiện nếu n=label2;*

*lệnh 1;*   
        break;  
    case *label3:*  
  *// lệnh 1 sẽ được thực hiện nếu n=label3;*

*lệnh 1;*   
        break;  
    ...  
    default:

// *lệnh 1 sẽ được thực hiện nếu if n khác các label trên*

*lệnh 1;*   
}

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$favcolor = "red";  
  
switch ($favcolor) {  
    case "red":  
        echo "Your favorite color is red!";  
        break;  
    case "blue":  
        echo "Your favorite color is blue!";  
        break;  
    case "green":  
        echo "Your favorite color is green!";  
        break;  
    default:  
        echo "Your favorite color is white!";  
}  
?>  
   
</body>  
</html>

Kết quả:

Your favorite color is red

## 13. Mệnh đề While Loops

Tilte: Mệnh đề while Loops

Map: menh-de-while-loops

Type:PHP

Meta Des: Mệnh đề while-loops trong php

Meta Keys: while , loops, menh de while-loop, mệnh đề while-loops

Description: Mệnh đê while – loops sẽ thực hiện lặp lại lệnh nếu điều kiện đúng.

Content:

**Mệnh đề while – loops**

Cú pháp :

while (*điều kiện*) {

// khi điều kiện là true thì lệnh 1 sẽ thực hiện liên tục  
*lệnh 1;*  
}

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php    
$x = 1;  
   
while($x <= 5) {  
  echo "The number is: $x <br>";  
  $x++;  
}   
?>    
  
</body>  
</html>

Kết quả:

The number is: 1   
The number is: 2   
The number is: 3   
The number is: 4   
The number is: 5

**Mệnh đề do – while**

Cú pháp:

do {

// lệnh sẽ thực hiện rồi kiểm tra điều kiện còn là true thì tiếp tục lặp  
*lệnh;*} while (*điều kiện là true*);

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php   
$x = 1;  
  
do {  
    echo "The number is: $x <br>";  
    $x++;  
} while ($x <= 5);  
?>  
  
</body>  
</html>

Kết quả:

The number is: 1   
The number is: 2   
The number is: 3   
The number is: 4   
The number is: 5

Điểm khác nhau của while – loops và do – while -loop là while – loops sẽ kiểm tra điều kiện chắc chắn là true thì mới thực hiện lệnh, còn do-while –loops thì lệnh sẽ thực hiện ít nhất 1 lần rồi kiểm tra điều kiện xem là true hay không để thực hiện tiếp

Ví dụ

<!DOCTYPE html>  
<html>  
<body>  
  
<?php   
$x = 6;  
  
do {  
    echo "The number is: $x <br>";  
    $x++;  
} while ($x <= 5);  
?>  
  
</body>  
</html>

Kết quả:

The number is: 6

## 14. Mệnh đề For Loops

Tilte: Mệnh đề For Loops

Map: menh-de-for-loops

Type:PHP

Meta Des: Mệnh đề for- loops trong php

Meta Keys: for, loops, php, mệnh đề, menh de

Description: Mệnh đề For-loops dùng để thực hiện với số lần xác định

Content:

**Mệnh đề For Loops**

Cú pháp :

for (*giá trị đầu; điều kiện; giá trị cuối*) {  
  *lệnh;*  
}

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php    
for ($x = 0; $x <= 10; $x++) {  
  echo "The number is: $x <br>";  
}  
?>    
  
</body>  
</html>

Kết quả:

The number is: 0   
The number is: 1   
The number is: 2   
The number is: 3   
The number is: 4   
The number is: 5   
The number is: 6   
The number is: 7   
The number is: 8   
The number is: 9   
The number is: 10

**Mệnh đề Foreach Loop**

Cú pháp:

foreach ([mảng]as*[*giá trị]) {  
    *lệnh;*  
}

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php    
$colors = array("red", "green", "blue", "yellow");   
  
foreach ($colors as $value) {  
  echo "$value <br>";  
}  
?>    
  
</body>  
</html>

Kết quả:

red   
green   
blue   
yellow

## 15.Hàm trong PHP

Tilte: Hàm trong PHP

Map: ham-trong-php

Type:PHP

Meta Des: Hàm trong PHP

Meta Keys: hàm php, ham trong php

Description: Điểm mạnh của php là sử dụng hàm. Hiện có hơn 1000 hàm .

Content:

**Người dùng xác định hàm**

Bên cạnh hàm PHP đã xây dựng, người dùng có thể tạo ra các hàm riêng.

Một hàm là một khối các câu lệnh có thể được sử dụng lặp đi lặp lại trong một chương trình.

Một hàm sẽ không thực hiện ngay khi một trang tải.

Một hàm sẽ được thực hiện bằng một cuộc gọi đến hàm.

**Tạo 1 hàm**

Cú pháp

function *functionName*() {  
*lệnh*;  
}

Tên của hàm không phân biệt chữ hoa chữ thường.

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
function writeMsg() {  
    echo "Hello world!<br>";  
}  
  
writeMsg();

wriTeMsg();  
?>  
  
</body>  
</html>

Kết quả

Hello world!

Hello world!

**Hàm có tham số**

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
function myName($fname) {  
    echo "My name is $fname<br>";  
}  
  
myName("Ha");  
myName("Linh");  
myName("Sang");  
?>  
  
</body>  
</html>

Kết quả:

My name is Ha

My name is Linh

My name is Sang

**Hàm có tham số có giá trị ban đầu**

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
function setHeight($minheight = 50) {  
    echo "The height is : $minheight <br>";  
}  
  
setHeight(350);  
setHeight();  
setHeight(135);  
setHeight(80);  
?>  
  
</body>  
</html>

Kết quả

The height is : 350   
The height is : 50   
The height is : 135   
The height is : 80

**Hàm trả về giá trị**

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
function sum($x, $y) {  
    $z = $x + $y;  
    return $z;  
}  
  
echo "5 + 10 = " . sum(5,10) . "<br>";  
echo "7 + 13 = " . sum(7,13) . "<br>";  
echo "2 + 4 = " . sum(2,4);  
?>  
  
</body>  
</html>

Kết quả:

5 + 10 = 15  
7 + 13 = 20  
2 + 4 = 6

## 16.Mảng trong PHP

Tilte: Mảng trong PHP

Map: mang-trong-php

Type:PHP

Meta Des: Mảng trong PHP

Meta Keys: array, mang trong php, mảng trong php

Description: 1 mảng có thể lưu nhiều giá trị với 1 biến duy nhất

Content:

**Mảng là gì ?**

Mảng là 1 biến đặt biệt có thể chứa nhiều hơn 1 giá trị tại 1 thời điểm.

Cú pháp: array();

**Mảng index**

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$cars = array("Volvo", "BMW", "Toyota");   
echo "I like " . $cars[0] . ", " . $cars[1] .".";  
?>  
  
</body>  
</html>

Kết quả :

I like Volvo, BMW.

**Hàm đếm số phẩn tử của mảng**

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$cars = array("Volvo", "BMW", "Toyota");  
echo count($cars);  
?>  
  
</body>  
</html>

Kết quả:

3

Ví dụ lấy các phần tử của mảng bằng for và index

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$cars = array("Volvo", "BMW", "Toyota");  
$arrlength = count($cars);  
  
for($x = 0; $x < $arrlength; $x++) {  
    echo $cars[$x];  
    echo "<br>";  
}  
?>  
  
</body>  
</html>

Kết quả:

Volvo  
BMW  
Toyota

**Mảng Object**

Mảng Object dùng theo kiểu key-value.

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");  
echo "Peter is " . $age['Peter'] . " years old.";  
?>  
  
</body>  
</html>

Kết quả:

Peter is 35 years old.

Ví dụ lấy các key value của 1 mảng Object:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");  
  
foreach($age as $x => $x\_value) {  
    echo "Key=" . $x . ", Value=" . $x\_value;  
    echo "<br>";  
}  
?>  
  
</body>  
</html>

Kết quả:

Key=Peter, Value=35  
Key=Ben, Value=37  
Key=Joe, Value=43

## 17. Sắp xếp thứ tự trong array

Tilte: Sắp xếp thứ tự trong array

Map: sap-xep-thu-tu-trong-array

Type:PHP

Meta Des: sắp xếp thứ tự trong array

Meta Keys: sắp xếp thứ tự trong array, thứ tự array, sort array

Description: Các phần tử trong một mảng có thể được sắp xếp theo thứ tự chữ cái hoặc số, giảm dần hoặc tăng dần.

Content:

**PHP hỗ trợ các loại sort**

* sort() - sắp xếp mảng theo thứ tự tăng dần
* rsort () - sắp xếp các mảng theo thứ tự giảm dần
* asort () - sắp xếp mảng kết hợp theo thứ tự tăng dần, theo giá trị
* ksort () - sắp xếp các mảng kết hợp theo thứ tự tăng dần, theo khóa
* arsort () - sắp xếp mảng kết hợp theo thứ tự giảm dần theo giá trị
* krsort () - sắp xếp các mảng kết hợp theo thứ tự giảm dần theo khoá

**Sắp xếp tăng dần sort()**

Ví dụ1:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$cars = array("Volvo", "BMW", "Toyota");  
sort($cars);  
  
$clength = count($cars);  
for($x = 0; $x < $clength; $x++) {  
    echo $cars[$x];  
    echo "<br>";  
}  
?>  
  
</body>  
</html>

Kết quả

BMW  
Toyota  
Volvo

Ví dụ2 :

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$numbers = array(4, 6, 2, 22, 11);  
sort($numbers);  
  
$arrlength = count($numbers);  
for($x = 0; $x < $arrlength; $x++) {  
    echo $numbers[$x];  
    echo "<br>";  
}  
?>  
  
</body>  
</html>

Kết quả

2  
4  
6  
11  
22

**Sắp xếp giảm dần rsort()**

Ví dụ1:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$cars = array("Volvo", "BMW", "Toyota");  
rsort($cars);  
  
$clength = count($cars);  
for($x = 0; $x < $clength; $x++) {  
    echo $cars[$x];  
    echo "<br>";  
}  
?>  
  
</body>  
</html>

Kết quả

Volvo  
Toyota  
BMW

Ví dụ2:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$numbers = array(4, 6, 2, 22, 11);  
rsort($numbers);  
  
$arrlength = count($numbers);  
for($x = 0; $x < $arrlength; $x++) {  
    echo $numbers[$x];  
    echo "<br>";  
}  
?>  
  
</body>  
</html>

Kết quả

22  
11  
6  
4  
2

**Sắp xếp tăng dần theo value asort()**

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");  
asort($age);  
  
foreach($age as $x => $x\_value) {  
    echo "Key=" . $x . ", Value=" . $x\_value;  
    echo "<br>";  
}  
?>  
  
</body>  
</html>

Kết quả

Key=Peter, Value=35  
Key=Ben, Value=37  
Key=Joe, Value=43

**Sắp xếp tăng dần theo key ksort()**

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");  
ksort($age);  
  
foreach($age as $x => $x\_value) {  
    echo "Key=" . $x . ", Value=" . $x\_value;  
    echo "<br>";  
}  
?>  
  
</body>  
</html>

Kết quả

Key=Ben, Value=37  
Key=Joe, Value=43  
Key=Peter, Value=35

**Sắp xếp giảm dần theo value arsort()**

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");  
arsort($age);  
  
foreach($age as $x => $x\_value) {  
    echo "Key=" . $x . ", Value=" . $x\_value;  
    echo "<br>";  
}  
?>  
  
</body>  
</html>

Kết quả

Key=Joe, Value=43  
Key=Ben, Value=37  
Key=Peter, Value=35

**Sắp xếp giảm dần theo key krsort()**

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$age = array("Peter"=>"35", "Ben"=>"37", "Joe"=>"43");  
krsort($age);  
  
foreach($age as $x => $x\_value) {  
    echo "Key=" . $x . ", Value=" . $x\_value;  
    echo "<br>";  
}  
?>  
  
</body>  
</html>

Kết quả

Key=Peter, Value=35  
Key=Joe, Value=43  
Key=Ben, Value=37

## 18.Các biến toàn cục phổ biến

Tilte: Các biến toàn cục phổ biến trong php

Map: cac-bien-toan-cuc-pho-bien

Type:PHP

Meta Des: Các biến toàn cục phổ biến trong php

Meta Keys: biến toàn cục phổ biến , biến toàn cuc, bien toan cuc, bien toan cuc php

Description: PHP có hỗ trợ vài biến gọi là “superglobal”. Các biến này bạn có thể gọi ở bất cứ đâu.

Content:

**Biến $GLOBAL**

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php   
$x = 75;  
$y = 25;   
  
function addition() {  
    $GLOBALS['z'] = $GLOBALS['x'] + $GLOBALS['y'];  
}  
  
addition();  
echo $z;  
?>  
  
</body>  
</html>

Kết quả:

100

**Biến $\_SERVER**

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php   
echo $\_SERVER['PHP\_SELF'];  
echo "<br>";  
echo $\_SERVER['SERVER\_NAME'];  
echo "<br>";  
echo $\_SERVER['HTTP\_HOST'];  
echo "<br>";  
echo $\_SERVER['HTTP\_REFERER'];  
echo "<br>";  
echo $\_SERVER['HTTP\_USER\_AGENT'];  
echo "<br>";  
echo $\_SERVER['SCRIPT\_NAME'];  
?>  
  
</body>  
</html>

Kết quả:

/php/demo\_global\_server.php  
www.codewr.com  
www.codewr.com  
https://www.codewr.com/php/showphp.asp?filename=demo\_global\_server  
Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/61.0.3163.100 Safari/537.36  
/php/demo\_global\_server.php

**Biến $\_REQUEST**

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<form method="post" action="<?php echo $\_SERVER['PHP\_SELF'];?>">  
  Name: <input type="text" name="fname">  
  <input type="submit">  
</form>  
  
<?php  
if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  
    // collect value of input field  
    $name = htmlspecialchars($\_REQUEST['fname']);   
    if (empty($name)) {  
        echo "Name is empty";  
    } else {  
        echo $name;  
    }  
}  
?>  
  
</body>  
</html>

Kết quả:

Nhận vào input và cho output

**Biến $\_POST**

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<form method="post" action="<?php echo $\_SERVER['PHP\_SELF'];?>">  
  Name: <input type="text" name="fname">  
  <input type="submit">  
</form>  
  
<?php  
if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  
    // collect value of input field  
    $name = $\_POST['fname'];   
    if (empty($name)) {  
        echo "Name is empty";  
    } else {  
        echo $name;  
    }  
}  
?>  
  
</body>  
</html>

Kết quả:

Nhận vào input và cho output

**Biến $\_GET**

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<a href="test\_get.php?subject=PHP&web=codewr.com">Test $GET</a>  
  
</body>  
</html>

Kết quả:

Study PHP at codewr.com

## 19. Form trong PHP

Tilte: Form trong PHP

Map: form-trong-php

Type:PHP

Meta Des: Form trong php

Meta Keys: xử lý form trong php, xu ly form trong php

Description: Form dữ liệu là 1 kiểu phổ biến được dùng trong php

Content:

**Form PHP đơn giản dùng POST**

Ví dụ:

<!DOCTYPE HTML>  
<html>    
<body>  
  
<form action="welcome.php" method="post">  
Name: <input type="text" name="name"><br>  
E-mail: <input type="text" name="email"><br>  
<input type="submit">  
</form>  
  
</body>  
</html>

Nội dụng của file welcome.php như sau

<html>  
<body>  
  
Welcome <?php echo $\_POST["name"]; ?><br>  
Your email address is: <?php echo $\_POST["email"]; ?>  
  
</body>  
</html>

Kết quả khi bạn nhập name:ABC và email: [ABC@gmail.com](mailto:ABC@gmail.com):

Welcome ABC  
Your email address is: [ABC@gmail.com](mailto:ABC@gmail.com)

**Form PHP đơn giản dùng GET**

Ví dụ :

<!DOCTYPE HTML>  
<html>    
<body>  
  
<form action="welcome\_get.php" method="get">  
Name: <input type="text" name="name"><br>  
E-mail: <input type="text" name="email"><br>  
<input type="submit">  
</form>  
  
</body>  
</html>

Nội dụng file welcome\_get.php

<html>  
<body>  
  
Welcome <?php echo $\_GET["name"]; ?><br>  
Your email address is: <?php echo $\_GET["email"]; ?>  
  
</body>  
</html>

Kết quả khi bạn nhập name:ABC và email: [ABC@gmail.com](mailto:ABC@gmail.com):

Welcome ABC  
Your email address is: [ABC@gmail.com](mailto:ABC@gmail.com)

**GET vs POST**

* Cả GET và POST đều tạo một mảng (ví dụ: mảng (key => value, key2 => value2, key3 => value3, ...)). Mảng này giữ các cặp khóa / giá trị, trong đó các khóa là tên của điều khiển biểu mẫu và giá trị là dữ liệu đầu vào từ người dùng.
* Cả GET và POST được coi là $ \_GET và $ \_POST. Đây là những superglobals, có nghĩa là chúng luôn có thể truy cập, bất kể phạm vi - và bạn có thể truy cập chúng từ bất kỳ hàm, lớp hoặc tệp nào mà không cần phải làm gì đặc biệt.
* $ \_GET là một mảng các biến được chuyển đến kịch bản hiện tại thông qua các tham số URL.
* $ \_POST là một mảng các biến được truyền đến kịch bản hiện tại thông qua phương thức HTTP POST.

**Khi nào dùng GET**

Thông tin được gửi từ một mẫu với phương thức GET là hiển thị cho tất cả mọi người (tất cả các tên biến và giá trị được hiển thị trong URL). GET cũng có giới hạn về số lượng thông tin để gửi. Giới hạn là khoảng 2000 ký tự. Tuy nhiên, vì các biến được hiển thị trong URL, có thể đánh dấu trang. Điều này có thể hữu ích trong một số trường hợp.

GET có thể được sử dụng để gửi dữ liệu không nhạy cảm.

Lưu ý: GET KHÔNG BAO GIỜ được sử dụng để gửi mật khẩu hoặc các thông tin nhạy cảm khác!

**Khi nào dùng POST**

Thông tin được gửi từ một mẫu với phương thức POST là không thể thấy được đối với những người khác (tất cả các tên / giá trị được nhúng trong phần yêu cầu HTTP) và không có giới hạn về lượng thông tin cần gửi.

Hơn nữa POST hỗ trợ các chức năng tiên tiến như hỗ trợ đầu vào nhị phân nhiều phần trong khi tải tệp lên máy chủ.

Tuy nhiên, vì các biến không được hiển thị trong URL, bạn không thể đánh dấu trang.

Nhà phát triển thích POST để gửi dữ liệu biểu mẫu.

Tiếp theo, hãy xem làm thế nào chúng ta có thể xử lý PHP một cách an toàn!

## 20. Form validation trong PHP

Tilte: Form validation trong php

Map: form-validation-trong-php

Type:PHP

Meta Des: Form validation trong php

Meta Keys: validation form , php, form validation

Description: Form validation là 1 hình thức xác thực đúng dữ liệu tránh nhập sai dữ liệu hoặc spam.

Content:

Ví dụ về 1 dạng form mà chưa validation

<!DOCTYPE HTML>    
<html>  
<head>  
</head>  
<body>    
  
<?php  
// define variables and set to empty values  
$name = $email = $gender = $comment = $website = "";  
  
if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  
  $name = test\_input($\_POST["name"]);  
  $email = test\_input($\_POST["email"]);  
  $website = test\_input($\_POST["website"]);  
  $comment = test\_input($\_POST["comment"]);  
  $gender = test\_input($\_POST["gender"]);  
}  
  
function test\_input($data) {  
  $data = trim($data);  
  $data = stripslashes($data);  
  $data = htmlspecialchars($data);  
  return $data;  
}  
?>  
  
<h2>PHP Form Validation Example</h2>  
<form method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]);?>">    
  Name: <input type="text" name="name">  
  <br><br>  
  E-mail: <input type="text" name="email">  
  <br><br>  
  Website: <input type="text" name="website">  
  <br><br>  
  Comment: <textarea name="comment" rows="5" cols="40"></textarea>  
  <br><br>  
  Gender:  
  <input type="radio" name="gender" value="female">Female  
  <input type="radio" name="gender" value="male">Male  
  <br><br>  
  <input type="submit" name="submit" value="Submit">    
</form>  
  
<?php  
echo "<h2>Your Input:</h2>";  
echo $name;  
echo "<br>";  
echo $email;  
echo "<br>";  
echo $website;  
echo "<br>";  
echo $comment;  
echo "<br>";  
echo $gender;  
?>  
  
</body>  
</html>

Đối với ví dụ tao có các dạng dữ liệu cần nhập

Name: dữ liệu nhập không có chữ cái nếu có required

Email: dữ liệu nhập phải có @ và dấu chấm (.) nếu có required

Website: Dữ liệu nhập vào phải là 1 url nếu có required

Comment: dữ liệu nhập vào phải có kí tự nếu có required

Gender: phải chọn 1 trong 2 option nếu có required

**$ \_SERVER ["PHP\_SELF"]** là một biến siêu toàn cầu trả về tên tập tin của tập lệnh đang thực hiện.

Vì vậy, **$ \_SERVER ["PHP\_SELF"]** gửi dữ liệu biểu mẫu được gửi đến trang đó, thay vì nhảy sang một trang khác. Bằng cách này, người dùng sẽ nhận được thông báo lỗi trên cùng một trang với mẫu.

Hàm **htmlspecialchars ()** chuyển các ký tự đặc biệt sang các thực thể HTML. Điều này có nghĩa là nó sẽ thay thế các ký tự HTML như <và> bằng & lt; và & gt ;. Điều này ngăn cản kẻ tấn công khai thác mã bằng cách tiêm chích mã HTML hoặc Javascript (các cuộc tấn công Cross-site Scripting) dưới các hình thức.

**Required Form**

Required là bắt buộc nhập trong form dùng để cho người dùng biết các vị trí cần có dữ liệu

Ví dụ:

<!DOCTYPE HTML>    
<html>  
<head>  
<style>  
.error {color: #FF0000;}  
</style>  
</head>  
<body>    
  
<?php  
// define variables and set to empty values  
$nameErr = $emailErr = $genderErr = $websiteErr = "";  
$name = $email = $gender = $comment = $website = "";  
  
if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  
  if (empty($\_POST["name"])) {  
    $nameErr = "Name is required";  
  } else {  
    $name = test\_input($\_POST["name"]);  
  }  
    
  if (empty($\_POST["email"])) {  
    $emailErr = "Email is required";  
  } else {  
    $email = test\_input($\_POST["email"]);  
  }  
      
  if (empty($\_POST["website"])) {  
    $website = "";  
  } else {  
    $website = test\_input($\_POST["website"]);  
  }  
  
  if (empty($\_POST["comment"])) {  
    $comment = "";  
  } else {  
    $comment = test\_input($\_POST["comment"]);  
  }  
  
  if (empty($\_POST["gender"])) {  
    $genderErr = "Gender is required";  
  } else {  
    $gender = test\_input($\_POST["gender"]);  
  }  
}  
  
function test\_input($data) {  
  $data = trim($data);  
  $data = stripslashes($data);  
  $data = htmlspecialchars($data);  
  return $data;  
}  
?>  
  
<h2>PHP Form Validation Example</h2>  
<p><span class="error">\* required field.</span></p>  
<form method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]);?>">    
  Name: <input type="text" name="name">  
  <span class="error">\* <?php echo $nameErr;?></span>  
  <br><br>  
  E-mail: <input type="text" name="email">  
  <span class="error">\* <?php echo $emailErr;?></span>  
  <br><br>  
  Website: <input type="text" name="website">  
  <span class="error"><?php echo $websiteErr;?></span>  
  <br><br>  
  Comment: <textarea name="comment" rows="5" cols="40"></textarea>  
  <br><br>  
  Gender:  
  <input type="radio" name="gender" value="female">Female  
  <input type="radio" name="gender" value="male">Male  
  <span class="error">\* <?php echo $genderErr;?></span>  
  <br><br>  
  <input type="submit" name="submit" value="Submit">    
</form>  
  
<?php  
echo "<h2>Your Input:</h2>";  
echo $name;  
echo "<br>";  
echo $email;  
echo "<br>";  
echo $website;  
echo "<br>";  
echo $comment;  
echo "<br>";  
echo $gender;  
?>  
  
</body>  
</html>

Ở ví dụ này có 3 bắt buộc nhập dữ liệu name, email, gender

Nếu bạn để 3 ô này trống hoặc không select thì error sẽ hiện thị ra.

**Validation cho Email và URL**

<!DOCTYPE HTML>    
<html>  
<head>  
<style>  
.error {color: #FF0000;}  
</style>  
</head>  
<body>    
  
<?php  
// define variables and set to empty values  
$nameErr = $emailErr = $genderErr = $websiteErr = "";  
$name = $email = $gender = $comment = $website = "";  
  
if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  
  if (empty($\_POST["name"])) {  
    $nameErr = "Name is required";  
  } else {  
    $name = test\_input($\_POST["name"]);  
    // check if name only contains letters and whitespace  
    if (!preg\_match("/^[a-zA-Z ]\*$/",$name)) {  
      $nameErr = "Only letters and white space allowed";   
    }  
  }  
    
  if (empty($\_POST["email"])) {  
    $emailErr = "Email is required";  
  } else {  
    $email = test\_input($\_POST["email"]);  
    // check if e-mail address is well-formed  
    if (!filter\_var($email, FILTER\_VALIDATE\_EMAIL)) {  
      $emailErr = "Invalid email format";   
    }  
  }  
      
  if (empty($\_POST["website"])) {  
    $website = "";  
  } else {  
    $website = test\_input($\_POST["website"]);  
    // check if URL address syntax is valid  
    if (!preg\_match("/\b(?:(?:https?|ftp):\/\/|www\.)[-a-z0-9+&@#\/%?=~\_|!:,.;]\*[-a-z0-9+&@#\/%=~\_|]/i",$website)) {  
      $websiteErr = "Invalid URL";   
    }      
  }  
  
  if (empty($\_POST["comment"])) {  
    $comment = "";  
  } else {  
    $comment = test\_input($\_POST["comment"]);  
  }  
  
  if (empty($\_POST["gender"])) {  
    $genderErr = "Gender is required";  
  } else {  
    $gender = test\_input($\_POST["gender"]);  
  }  
}  
  
function test\_input($data) {  
  $data = trim($data);  
  $data = stripslashes($data);  
  $data = htmlspecialchars($data);  
  return $data;  
}  
?>  
  
<h2>PHP Form Validation Example</h2>  
<p><span class="error">\* required field.</span></p>  
<form method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]);?>">    
  Name: <input type="text" name="name">  
  <span class="error">\* <?php echo $nameErr;?></span>  
  <br><br>  
  E-mail: <input type="text" name="email">  
  <span class="error">\* <?php echo $emailErr;?></span>  
  <br><br>  
  Website: <input type="text" name="website">  
  <span class="error"><?php echo $websiteErr;?></span>  
  <br><br>  
  Comment: <textarea name="comment" rows="5" cols="40"></textarea>  
  <br><br>  
  Gender:  
  <input type="radio" name="gender" value="female">Female  
  <input type="radio" name="gender" value="male">Male  
  <span class="error">\* <?php echo $genderErr;?></span>  
  <br><br>  
  <input type="submit" name="submit" value="Submit">    
</form>  
  
<?php  
echo "<h2>Your Input:</h2>";  
echo $name;  
echo "<br>";  
echo $email;  
echo "<br>";  
echo $website;  
echo "<br>";  
echo $comment;  
echo "<br>";  
echo $gender;  
?>  
  
</body>  
</html>

Khi bạn nhập sai format email hoặc url thì sẽ báo lỗi

Validation đầy đủ cho form :

<!DOCTYPE HTML>    
<html>  
<head>  
<style>  
.error {color: #FF0000;}  
</style>  
</head>  
<body>    
  
<?php  
// define variables and set to empty values  
$nameErr = $emailErr = $genderErr = $websiteErr = "";  
$name = $email = $gender = $comment = $website = "";  
  
if ($\_SERVER["REQUEST\_METHOD"] == "POST") {  
  if (empty($\_POST["name"])) {  
    $nameErr = "Name is required";  
  } else {  
    $name = test\_input($\_POST["name"]);  
    // check if name only contains letters and whitespace  
    if (!preg\_match("/^[a-zA-Z ]\*$/",$name)) {  
      $nameErr = "Only letters and white space allowed";   
    }  
  }  
    
  if (empty($\_POST["email"])) {  
    $emailErr = "Email is required";  
  } else {  
    $email = test\_input($\_POST["email"]);  
    // check if e-mail address is well-formed  
    if (!filter\_var($email, FILTER\_VALIDATE\_EMAIL)) {  
      $emailErr = "Invalid email format";   
    }  
  }  
      
  if (empty($\_POST["website"])) {  
    $website = "";  
  } else {  
    $website = test\_input($\_POST["website"]);  
    // check if URL address syntax is valid (this regular expression also allows dashes in the URL)  
    if (!preg\_match("/\b(?:(?:https?|ftp):\/\/|www\.)[-a-z0-9+&@#\/%?=~\_|!:,.;]\*[-a-z0-9+&@#\/%=~\_|]/i",$website)) {  
      $websiteErr = "Invalid URL";   
    }  
  }  
  
  if (empty($\_POST["comment"])) {  
    $comment = "";  
  } else {  
    $comment = test\_input($\_POST["comment"]);  
  }  
  
  if (empty($\_POST["gender"])) {  
    $genderErr = "Gender is required";  
  } else {  
    $gender = test\_input($\_POST["gender"]);  
  }  
}  
  
function test\_input($data) {  
  $data = trim($data);  
  $data = stripslashes($data);  
  $data = htmlspecialchars($data);  
  return $data;  
}  
?>  
  
<h2>PHP Form Validation Example</h2>  
<p><span class="error">\* required field.</span></p>  
<form method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]);?>">    
  Name: <input type="text" name="name" value="<?php echo $name;?>">  
  <span class="error">\* <?php echo $nameErr;?></span>  
  <br><br>  
  E-mail: <input type="text" name="email" value="<?php echo $email;?>">  
  <span class="error">\* <?php echo $emailErr;?></span>  
  <br><br>  
  Website: <input type="text" name="website" value="<?php echo $website;?>">  
  <span class="error"><?php echo $websiteErr;?></span>  
  <br><br>  
  Comment: <textarea name="comment" rows="5" cols="40"><?php echo $comment;?></textarea>  
  <br><br>  
  Gender:  
  <input type="radio" name="gender" <?php if (isset($gender) && $gender=="female") echo"checked";?> value="female">Female  
  <input type="radio" name="gender" <?php if (isset($gender) && $gender=="male") echo"checked";?> value="male">Male  
  <span class="error">\* <?php echo $genderErr;?></span>  
  <br><br>  
  <input type="submit" name="submit" value="Submit">    
</form>  
  
<?php  
echo "<h2>Your Input:</h2>";  
echo $name;  
echo "<br>";  
echo $email;  
echo "<br>";  
echo $website;  
echo "<br>";  
echo $comment;  
echo "<br>";  
echo $gender;  
?>  
  
</body>  
</html>

## 21. Mảng đa chiều trong PHP

Tilte: Mảng đa chiều trong PHP

Map: mang-da-chieu-trong-php

Type:PHPAD

Meta Des: Mảng đa chiều trong PHP

Meta Keys: nhieu mang trong php, mang da chieu trong php,nhiều mảng trong php, mảng đã chiều trong php

Description: Mảng đa chiều là một mảng chứa nhiều hơn 2 mảng con

Content:

**Mảng 2 chiều**

Ví dụ :

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$cars = array  
  (  
  array("Volvo",22,18),  
  array("BMW",15,13),  
  array("Saab",5,2),  
  array("Land Rover",17,15)  
  );  
    
echo $cars[0][0].": In stock: ".$cars[0][1].", sold: ".$cars[0][2].".<br>";  
echo $cars[1][0].": In stock: ".$cars[1][1].", sold: ".$cars[1][2].".<br>";  
echo $cars[2][0].": In stock: ".$cars[2][1].", sold: ".$cars[2][2].".<br>";  
echo $cars[3][0].": In stock: ".$cars[3][1].", sold: ".$cars[3][2].".<br>";  
?>  
  
</body>  
</html>

Kết quả:

Volvo: In stock: 22, sold: 18.  
BMW: In stock: 15, sold: 13.  
Saab: In stock: 5, sold: 2.  
Land Rover: In stock: 17, sold: 15.

Ví dụ cũng mảng 2 chiều bạn có thể in ra với for:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$cars = array  
  (  
  array("Volvo",22,18),  
  array("BMW",15,13),  
  array("Saab",5,2),  
  array("Land Rover",17,15)  
  );  
      
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>";  
}  
?>  
  
</body>  
</html>

Kết quả:

**Row number 0**

* Volvo
* 22
* 18

**Row number 1**

* BMW
* 15
* 13

**Row number 2**

* Saab
* 5
* 2

**Row number 3**

* Land Rover
* 17
* 15

## 22. Thời gian và ngày tháng trong PHP

Tilte: Thời gian và ngày tháng trong PHP

Map: thoi-gian-va-ngay-thang-trong-php

Type:PHPAD

Meta Des: Ngày tháng và thời gian trong php

Meta Keys: Tháng ngày trong php, thời gian trong php, thoi gian trong php

Description: Hàm date() trong php được sử dụng để định dạng ngày tháng và thời gian

Content:

**Hàm date()**

Cú pháp:

date(*format*,*timestamp*)

- format : là bắt buộc định dạng cho thời gian

- timestamp: mặc định là thời gian hiện tại

Format trong date như sau:

- d: đại diện cho ngày

- m: đại diện cho tháng

- Y: đại diện cho Năm

- l: đại điện cho ngày trong tuần

- h: đại diện cho giờ

- i: đại diện cho phút

- s: đại diện cho giây

- a: đại diện cho buổi trong ngày AM or PM

- Có 3 kí để định dạng để ngăn các ngày tháng năm như sau : / , . , -

Ví dụ 1 cho format ngày :

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
echo "Today is " . date("Y/m/d") . "<br>";  
echo "Today is " . date("Y.m.d") . "<br>";  
echo "Today is " . date("Y-m-d") . "<br>";  
echo "Today is " . date("l");  
?>  
  
</body>  
</html>

Kết quả :

Today is 2017/11/03  
Today is 2017.11.03  
Today is 2017-11-03  
Today is Friday

Ví dụ 2 về định dạng thời gian:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
echo "The time is " . date("h:i:sa");  
?>  
  
</body>  
</html>

Kết quả:

The time is 02:37:56am

Ví dụ 3 muốn lấy thời gian hiện tại ở 1 khu vực nào đó:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
date\_default\_timezone\_set("America/New\_York");  
echo "The time is " . date("h:i:sa");  
?>  
  
</body>  
</html>

Kết quả :

The time is 02:39:29am

**Hàm tạo 1 ngày tháng cụ thể mktime()**

Cú pháp:

mktime(hour,minute,second,month,day,year)

Ví dụ 4:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$d=mktime(11, 14, 54, 8, 12, 2017);  
echo "Created date is " . date("Y-m-d h:i:sa", $d);  
?>  
  
</body>  
</html>

Kết quả:

Created date is 2014-08-12 11:14:54am

**Hàm tạo 1 ngày tháng cụ thể từ string strtotime()**

Cú pháp 5:

strtotime(time,now)

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$d=strtotime("10:30pm April 15 2017");  
echo "Created date is " . date("Y-m-d h:i:sa", $d). "<br>";

$d=strtotime("tomorrow");  
echo date("Y-m-d h:i:sa", $d) . "<br>";  
  
$d=strtotime("next Saturday");  
echo date("Y-m-d h:i:sa", $d) . "<br>";  
  
$d=strtotime("+3 Months");  
echo date("Y-m-d h:i:sa", $d) . "<br>";  
?>  
  
</body>  
</html>

Kết quả:

Created date is 2017-04-15 10:30:00pm

2017-11-04 12:00:00am  
2017-11-04 12:00:00am  
2018-02-03 03:08:23am

Ví dụ 6:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$startdate=strtotime("Saturday");  
$enddate=strtotime("+6 weeks", $startdate);  
  
while ($startdate < $enddate) {  
  echo date("M d", $startdate) . "<br>";  
  $startdate = strtotime("+1 week", $startdate);  
}  
?>  
  
</body>  
</html>

Kết quả:

Nov 04  
Nov 11  
Nov 18  
Nov 25  
Dec 02  
Dec 09

Ví dụ 7:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$d1=strtotime("Nov 04");  
$d2=ceil(($d1-time())/60/60/24);  
echo "There are " . $d2 ." days until 4th of July.";  
?>  
  
</body>  
</html>

Kết quả:

There are -122 days until 4th of July.

## 23.Chèn file php trong 1 file view

Tilte: Chèn file php vào 1 file php view

Map: chen-file-trong-file-khac-php

Type:PHPAD

Meta Des: Chèn 1 file php vào 1 file php khác

Meta Keys: chèn file php vào 1 file php khác, chèn file php, chen file php

Description: Include và require là 2 từ khóa để chuyển 1 file php vào 1 file php khác

Content:

**Include và Require**

PHP hỗ trợ cho chèn nội dụng từ 1 file php này sang 1 file php khác. Việc này rất phổ biến khi bạn có nhiều trang nhưng có 1 phần nội dung nhỏ lặp lại ở các trang ví dụ như header, footer của web. PHP cũng cấp 2 từ khóa là include và require để thực hiện việc gọi file.

Điểm khác nhau giữa include và require là : require nếu file không tồn tại thì sẽ tạo ra 1 lỗi và dùng các lệnh tiếp theo, còn include thì sẽ tiếp tục thực hiện các lệnh tiếp theo và không có báo lỗi hoặc cảnh báo.

Cú pháp:

include '*filename*';  
  
or  
  
require '*filename*';

Ví dụ 1:

Tạo 1 file footer.php nội dụng sau:

<?php  
echo "<p>Copyright &copy; 2016-" . date("Y") . " CodeWR.com</p>";  
?>

File chạy main php có nội dung

<!DOCTYPE html>  
<html>  
<body>  
  
<h1>Welcome to my home page!</h1>  
<p>Some text.</p>  
<p>Some more text.</p>  
<?php include 'footer.php';?>  
  
</body>  
</html>

Kết quả:

**Welcome to my home page!**

Some text.

Some more text.

Copyright © 2016-2017 CodeWR.com

Ví dụ 2 include menu.php

File menu.php có nội dụng

<?php  
echo '<a href="/default.asp">Home</a> -  
<a href="/html/default.asp">HTML </a> -  
<a href="/css/default.asp">CSS </a> -  
<a href="/js/default.asp">JavaScript </a> -  
<a href="default.asp">PHP </a>';  
?>

File main.php

<html>  
<body>  
  
<div class="menu">  
<?php include 'menu.php';?>  
</div>  
  
<h1>Welcome to my home page!</h1>  
<p>Some text.</p>  
<p>Some more text.</p>  
  
</body>  
</html>

Kết quả

[Home](https://www.w3schools.com/default.asp) - [HTML](https://www.w3schools.com/html/default.asp) - [CSS](https://www.w3schools.com/css/default.asp) - [JavaScript](https://www.w3schools.com/js/default.asp) - [PHP](https://www.w3schools.com/php/default.asp)

**Welcome to my home page!**

Some text.

Some more text.

Ví dụ 3: lấy nội dung biến từ file php:

File vars.php có nội dung như sau:

<?php  
$color='red';  
$car='BMW';  
?>

File main.php có nội dung :

<!DOCTYPE html>  
<html>  
<body>  
  
<h1>Welcome to my home page!</h1>  
<?php include 'vars.php';  
echo "I have a $color $car.";  
?>  
  
</body>  
</html>

Kết quả:

**Welcome to my home page**!

I have a red BMW.

**2 ví dụ xem sự khác khau của include và require**

Ví dụ 4: include 1 file không tồn tại

<!DOCTYPE html>  
<html>  
<body>  
<?php include 'noFileExists.php';  
echo "I have a $color $car.";  
?>  
  
</body>  
</html>

Kết quả:

I have a .

Do không có file tồn tại nên biến $color và $car không có

Ví dụ 5: require 1 file không tồn tại

<!DOCTYPE html>  
<html>  
<body>  
  
<b>Welcome to my home page!</b>  
<?php require 'noFileExists.php';  
echo "I have a $color $car.";  
?>  
  
</body>  
</html>

Kết quả:

Welcome to my home page!

Do file không tồn tại mà dùng require nên lệnh echo không thực hiện được

## 24. Xử lý file trong php

Tilte: Xử lý file trong php

Map: xu-ly-file-trong-php

Type:PHPAD

Meta Des: Xử lý file trong php

Meta Keys: Đọc file trong php

Description: Xử lý file là một trong những ứng dụng web rất quan trọng như đọc dữ liệu và viết dữ liệu từ file

Content:

PHP hỗ trợ các function để tạo, đọc, viết file

Ví dụ về đọc file

Tạo 1 file text.txt với nội dung sau:

AJAX = Asynchronous JavaScript and XML  
CSS = Cascading Style Sheets  
HTML = Hyper Text Markup Language  
PHP = PHP Hypertext Preprocessor  
SQL = Structured Query Language  
SVG = Scalable Vector Graphics  
XML = EXtensible Markup Language

File main.php:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
echo readfile("text.txt");  
?>  
  
</body>  
</html>

Kết quả

AJAX = Asynchronous JavaScript and XML CSS = Cascading Style Sheets HTML = Hyper Text Markup Language PHP = PHP Hypertext Preprocessor SQL = Structured Query Language SVG = Scalable Vector Graphics XML = EXtensible Markup Language236

## 25. Mở và đọc file trong php

Tilte: Mở và đọc file trong php

Map: mo-va-doc-file-trong-php

Type:PHPAD

Meta Des: Mở và đọc file trong php

Meta Keys: Mở file trong php, read dữ liệu từ file trong php, doc du lieu tu file trong php

Description: PHP hỗ trợ các hàm mở file, đọc file và đóng file trong server

Content:

**Hàm mở file fopen()**

Ví dụ:

Tạo 1 file text.txt có nội dung sau:

AJAX = Asynchronous JavaScript and XML  
CSS = Cascading Style Sheets  
HTML = Hyper Text Markup Language  
PHP = PHP Hypertext Preprocessor  
SQL = Structured Query Language  
SVG = Scalable Vector Graphics  
XML = EXtensible Markup Language

File main.php có nội dung:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$myfile = fopen("text.txt", "r") or die("Unable to open file!");  
echo fread($myfile,filesize("text.txt"));  
fclose($myfile);  
?>  
  
</body>  
</html>

Kết quả:

AJAX = Asynchronous JavaScript and XML CSS = Cascading Style Sheets HTML = Hyper Text Markup Language PHP = PHP Hypertext Preprocessor SQL = Structured Query Language SVG = Scalable Vector Graphics XML = EXtensible Markup Language

Các param truyền vào hàm fopen:

- r: Mở tập tin để chỉ đọc. Con trỏ tập tin bắt đầu ở đầu tập tin

- w: Chỉ mở tập tin để ghi. Xóa nội dung của tệp hoặc tạo một tệp mới nếu tệp không tồn tại. Con trỏ tập tin bắt đầu ở đầu tập tin

- a: Mở một tệp để ghi. Dữ liệu hiện có trong tệp được bảo toàn. Con trỏ tập tin bắt đầu ở cuối của tập tin. Tạo một tệp mới nếu tệp không tồn tại

- x: Tạo ra một tập tin mới chỉ để ghi. Trả về FALSE và lỗi nếu tệp đã tồn tại

- r+: Mở một tập tin để đọc / ghi. Con trỏ tập tin bắt đầu ở đầu tập tin

- w +: Mở một tập tin để đọc / ghi. Xóa nội dung của tệp hoặc tạo một tệp mới nếu tệp không tồn tại. Con trỏ tập tin bắt đầu ở đầu tập tin

- a + Mở một tập tin để đọc / ghi. Dữ liệu hiện có trong tệp được bảo toàn. Con trỏ tập tin bắt đầu ở cuối của tập tin. Tạo một tệp mới nếu tệp không tồn tại

- x + Tạo một tập tin mới để đọc / ghi. Trả về FALSE và lỗi nếu tệp đã tồn tại

**Hàm đọc file fread()**

Cú pháp: fread(filename,filesize)

* filename: file cần đọc
* filesize: số byte tối đa để đọc

**Hàm đóng file fclose()**

Cú pháp : fclose(filename)

* filename: tên file cần đóng

**Hàm được file theo dòng fgets()**

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$myfile = fopen("text.txt", "r") or die("Unable to open file!");  
echo fgets($myfile);  
fclose($myfile);  
?>  
  
</body>  
</html>

Kết quả:

AJAX = Asynchronous JavaScript and XML

**Hàm kiểm tra tới nội dung cuối của file chưa feof()**

Ví dụ đọc hết nội dung file :

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$myfile = fopen("text.txt", "r") or die("Unable to open file!");  
// Output one line until end-of-file  
while(!feof($myfile)) {  
  echo fgets($myfile) . "<br>";  
}  
fclose($myfile);  
?>  
  
</body>  
</html>

Kết quả:

AJAX = Asynchronous JavaScript and XML   
CSS = Cascading Style Sheets   
HTML = Hyper Text Markup Language   
PHP = PHP Hypertext Preprocessor   
SQL = Structured Query Language   
SVG = Scalable Vector Graphics   
XML = EXtensible Markup Language

**Hàm đọc 1 ký từ fgetc()**

Ví dụ:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$myfile = fopen("text.txt", "r") or die("Unable to open file!");  
// Output one character until end-of-file  
while(!feof($myfile)) {  
  echo fgetc($myfile);  
}  
fclose($myfile);  
?>  
  
</body>  
</html>

Kết quả:

AJAX = Asynchronous JavaScript and XML CSS = Cascading Style Sheets HTML = Hyper Text Markup Language PHP = PHP Hypertext Preprocessor SQL = Structured Query Language SVG = Scalable Vector Graphics XML = EXtensible Markup Language

## 26. Tạo và ghi file trong php

Tilte: Tạo và ghi file trong php

Map: tao-va-ghi-file-trong-php

Type:PHPAD

Meta Des: Tạo và ghi file trong php

Meta Keys: tạo file, tao file, ghi file,ghi file trong php

Description: PHP cũng cấp các hàm tạo file và ghi file trong php

Content:

**Hàm tạo file fopen()**

Hàm fopen cũng là hàm tạo file nếu nó không tồn tại. khi param truyền vào là w thì file đã tồn tại, còn a thì nếu file chưa tồn tại nó sẽ tự động tạo 1 file.

**Hàm ghi file fwrite()**  
<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$myfile = fopen("newText.txt", "w") or die("Unable to open file!");  
$txt = "John Doe\n";  
fwrite($myfile, $txt);  
$txt = "Jane Doe\n";  
fwrite($myfile, $txt);  
fclose($myfile);  
?>  
</body>  
</html>

Kết quả:

Mickey Mouse  
Minnie Mouse

## 27. Upload file trong php

Tilte: Upload file trong php

Map: upload-file-trong-php

Type:PHPAD

Meta Des: Upload file trong php

Meta Keys: upload file, form upload file php,tải file lên server php

Description: PHP hỗ trợ upload file lên server rất dễ dàng

Content:

Để upload được file bạn phải chỉnh config trong file php.init ở thư mục C:\\xampp\php:

Search và set is on

file\_uploads = On

Ví dụ tạo 1 form upload file như sau:

Tạo file upload.php có nội dung

<?php  
$target\_dir = "uploads/";  
$target\_file = $target\_dir . basename($\_FILES["fileToUpload"]["name"]);  
$uploadOk = 1;  
$imageFileType = pathinfo($target\_file,PATHINFO\_EXTENSION);  
// Check if image file is a actual image or fake image  
if(isset($\_POST["submit"])) {  
    $check = getimagesize($\_FILES["fileToUpload"]["tmp\_name"]);  
    if($check !== false) {  
        echo "File is an image - " . $check["mime"] . ".";  
        $uploadOk = 1;  
    } else {  
        echo "File is not an image.";  
        $uploadOk = 0;  
    }  
}  
// Check if file already exists  
if (file\_exists($target\_file)) {  
    echo "Sorry, file already exists.";  
    $uploadOk = 0;  
}  
// Check file size  
if ($\_FILES["fileToUpload"]["size"] > 500000) {  
    echo "Sorry, your file is too large.";  
    $uploadOk = 0;  
}  
// Allow certain file formats  
if($imageFileType != "jpg" && $imageFileType != "png" && $imageFileType != "jpeg"  
&& $imageFileType != "gif" ) {  
    echo "Sorry, only JPG, JPEG, PNG & GIF files are allowed.";  
    $uploadOk = 0;  
}  
// Check if $uploadOk is set to 0 by an error  
if ($uploadOk == 0) {  
    echo "Sorry, your file was not uploaded.";  
// if everything is ok, try to upload file  
} else {  
    if (move\_uploaded\_file($\_FILES["fileToUpload"]["tmp\_name"], $target\_file)) {  
        echo "The file ". basename( $\_FILES["fileToUpload"]["name"]). " has been uploaded.";  
    } else {  
        echo "Sorry, there was an error uploading your file.";  
    }  
}  
?>

File main.php:

<!DOCTYPE html>  
<html>  
<body>  
  
<form action="upload.php" method="post" enctype="multipart/form-data">  
    Select image to upload:  
    <input type="file" name="fileToUpload" id="fileToUpload">  
    <input type="submit" value="Upload Image" name="submit">  
</form>  
  
</body>  
</html>

Note: chú ý bạn phải tạo trước thư mục uploads

## 28. Cookie trong php

Tilte: Cookie trong php

Map: cookie-trong-php

Type:PHPAD

Meta Des: Cookie trong php

Meta Keys: cookie trong php

Description: Cookie thường được sử dụng để xác định người dùng

Content:

**Cookie là gì?**

Cookie thường được sử dụng để xác định người dùng. Cookie là một tệp tin nhỏ mà máy chủ được nhúng vào máy tính của người dùng. Mỗi khi cùng một máy tính yêu cầu một trang với trình duyệt, nó cũng sẽ gửi cookie. Với PHP, bạn có thể tạo và lấy các giá trị cookie.

**Tạo cookie với php**

Cú pháp:

setcookie(name, value, expire, path, domain, secure, httponly);

Ví dụ tạo 1 cookie:

<!DOCTYPE html>  
<?php  
$cookie\_name = "user";  
$cookie\_value = "John Doe";  
setcookie($cookie\_name, $cookie\_value, time() + (86400 \* 30), "/"); // 86400 = 1 day  
?>  
<html>  
<body>  
  
<?php  
if(!isset($\_COOKIE[$cookie\_name])) {  
     echo "Cookie named '" . $cookie\_name . "' is not set!";  
} else {  
     echo "Cookie '" . $cookie\_name . "' is set!<br>";  
     echo "Value is: " . $\_COOKIE[$cookie\_name];  
}  
?>  
  
<p><strong>Note:</strong> You might have to reload the page to see the value of the cookie.</p>  
  
</body>  
</html>

Kết quả :

Cookie named 'user' is not set!

**Note:** You might have to reload the page to see the value of the cookie.

**Thay đổi cookie**

Vi dụ:

<!DOCTYPE html>  
<?php  
$cookie\_name = "user";  
$cookie\_value = "Alex Porter";  
setcookie($cookie\_name, $cookie\_value, time() + (86400 \* 30), "/");  
?>  
<html>  
<body>  
  
<?php  
if(!isset($\_COOKIE[$cookie\_name])) {  
     echo "Cookie named '" . $cookie\_name . "' is not set!";  
} else {  
     echo "Cookie '" . $cookie\_name . "' is set!<br>";  
     echo "Value is: " . $\_COOKIE[$cookie\_name];  
}  
?>  
  
<p><strong>Note:</strong> You might have to reload the page to see the new value of the cookie.</p>  
  
</body>  
</html>

Kết quả:

Cookie 'user' is set!  
Value is: John Doe

**Note:** You might have to reload the page to see the new value of the cookie.

**Xóa cookie**

Ví dụ:

<!DOCTYPE html>  
<?php  
// set the expiration date to one hour ago  
setcookie("user", "", time() - 3600);  
?>  
<html>  
<body>  
  
<?php  
echo "Cookie 'user' is deleted.";  
?>  
  
</body>  
</html>

Kết quả:

Cookie 'user' is deleted.

**Check cookie có tồn tại**

Ví dụ:

<!DOCTYPE html>  
<?php  
setcookie("test\_cookie", "test", time() + 3600, '/');  
?>  
<html>  
<body>  
  
<?php  
if(count($\_COOKIE) > 0) {  
    echo "Cookies are enabled.";  
} else {  
    echo "Cookies are disabled.";  
}  
?>  
  
</body>  
</html>

Kết quả:

Cookies are enabled.

## 29. Session trong php

Tilte: Session trong php

Map: session-trong-php

Type:PHPAD

Meta Des: Session trong php

Meta Keys: session trong php, set session trong php

Description: Session là cách lưu thông tin để sử dụng trên nhiều trang

Content:

**Session là gì?**

Session là 1 phiên làm việc tính từ lúc open ứng dụng tới khi đóng ứng dụng. Session thì chứa thông tin về một người dùng duy nhất . Nên sẽ được sử dụng ở nhiều trang khác nhau.

Bắt đầu session trong php

Ví dụ:

<?php  
// Start the session  
session\_start();  
?>  
<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
// Set session variables  
$\_SESSION["favcolor"] = "green";  
$\_SESSION["favanimal"] = "cat";  
echo "Session variables are set.";  
?>  
  
</body>  
</html>

Kết quả:

Session variables are set.

Lấy giá trị của session

Ví dụ:

<?php  
session\_start();  
?>  
<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
// Echo session variables that were set on previous page  
echo "Favorite color is " . $\_SESSION["favcolor"] . ".<br>";  
echo "Favorite animal is " . $\_SESSION["favanimal"] . ".";  
?>  
  
</body>  
</html>

Kết quả:

Favorite color is green.  
Favorite animal is cat.

Ví dụ lấy tất cả các session

<?php  
session\_start();  
?>  
<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
print\_r($\_SESSION);  
?>  
  
</body>  
</html>

Kết quả:

Array ( [favcolor] => green [favanimal] => cat )

**Thay đổi session**

Ví dụ:

<?php  
session\_start();  
?>  
<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
// to change a session variable, just overwrite it   
$\_SESSION["favcolor"] = "yellow";  
print\_r($\_SESSION);  
?>  
  
</body>  
</html>

Kết quả:

Array ( [favcolor] => yellow [favanimal] => cat )

**Xóa tất cả session**

Ví dụ:

<?php  
session\_start();  
?>  
<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
// remove all session variables  
session\_unset();   
  
// destroy the session   
session\_destroy();  
  
echo "All session variables are now removed, and the session is destroyed."   
?>  
  
</body>  
</html>

Kết quả:

All session variables are now removed, and the session is destroyed.

## 30. Bộ lộc dữ liệu trong php

Tilte: Bộ lọc dữ liệu trong php

Map: bo-loc-du-lieu-trong-php

Type:PHPAD

Meta Des: Bộ lọc dữ liệu trong php

Meta Keys: Bo loc du lieu, bộ lọc dữ liệu trong php

Description: PHP hỗ trợ các hàm để validation dữ liệu đầu vào

Content:

**Bộ lọc php**

Php hỗ trợ nhiều hàm xác minh dữ liệu .

Ví dụ lấy toàn bộ các filter trong php:

<!DOCTYPE html>  
<html>  
<head>  
<style>  
table, th, td {  
    border: 1px solid black;  
    border-collapse: collapse;  
}  
th, td {  
    padding: 5px;  
}  
</style>  
</head>  
<body>  
  
<table>  
  <tr>  
    <td>Filter Name</td>  
    <td>Filter ID</td>  
  </tr>  
  <?php  
  foreach (filter\_list() as $id =>$filter) {  
      echo '<tr><td>' . $filter . '</td><td>' . filter\_id($filter) . '</td></tr>';  
  }  
  ?>  
</table>  
  
</body>  
</html>

Kết quả có các bộ lọc xác minh như sau:

|  |  |
| --- | --- |
| Filter name | Filter ID |
| int | 257 |
| boolean | 258 |
| float | 259 |
| validate\_regexp | 272 |
| validate\_url | 273 |
| validate\_email | 274 |
| validate\_ip | 275 |
| string | 513 |
| stripped | 513 |
| encoded | 514 |
| special\_chars | 515 |
| full\_special\_chars | 522 |
| unsafe\_raw | 516 |
| email | 517 |
| url | 518 |
| number\_int | 519 |
| number\_float | 520 |
| magic\_quotes | 521 |
| callback | 1024 |

**Tại sao lại sử dụng bộ lọc**

Hiện có nhiều form mẫu nên việc sử dụng bộ lọc để tránh spam. Ví dụ ô nhập số điện thoại mà nhập chữ sẽ báo lỗi.

Ví du String:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$str = "<h1>Hello World!</h1>";  
$newstr = filter\_var($str, FILTER\_SANITIZE\_STRING);  
echo $newstr;  
?>  
  
</body>  
</html>

Kết quả:

Hello World!

Ví dụ 2 Integer:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$int = 100;  
  
if (!filter\_var($int, FILTER\_VALIDATE\_INT) === false) {  
    echo("Integer is valid");  
} else {  
    echo("Integer is not valid");  
}  
?>  
  
</body>  
</html>

Kết quả:

Integer is valid

Ví dụ 3 Đối với int mà biến =0 thì sẽ báo invalid:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$int = 0;  
  
if (filter\_var($int, FILTER\_VALIDATE\_INT) === 0 || !filter\_var($int, FILTER\_VALIDATE\_INT) === false) {  
    echo("Integer is valid");  
} else {  
    echo("Integer is not valid");  
}  
?>  
  
</body>  
</html>

Kết quả:

Integer is valid

Ví dụ 4 validate hàm ip address:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$ip = "127.0.0.1";  
  
if (!filter\_var($ip, FILTER\_VALIDATE\_IP) === false) {  
    echo("$ip is a valid IP address");  
} else {  
    echo("$ip is not a valid IP address");  
}  
?>  
  
</body>  
</html>

Kết quả:

127.0.0.1 is a valid IP address

Ví dụ 5 bộ lọc cho địa chỉ email:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$email = "john.doe@example.com";  
  
// Remove all illegal characters from email  
$email = filter\_var($email, FILTER\_SANITIZE\_EMAIL);  
  
// Validate e-mail  
if (!filter\_var($email, FILTER\_VALIDATE\_EMAIL) === false) {  
    echo("$email is a valid email address");  
} else {  
    echo("$email is not a valid email address");  
}  
?>  
  
</body>  
</html>

Kết quả:

john.doe@example.com is a valid email address

Ví dụ 6 validate for url:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
$url = "https://www.CodeWR.com";  
  
// Remove all illegal characters from a url  
$url = filter\_var($url, FILTER\_SANITIZE\_URL);  
  
// Validate url  
if (!filter\_var($url, FILTER\_VALIDATE\_URL) === false) {  
    echo("$url is a valid URL");  
} else {  
    echo("$url is not a valid URL");  
}  
?>  
  
</body>  
</html>

Kết quả:

https://www.CodeWR.com is a valid URL

## 31.Bộ loc dữ liệu nâng cao trong php

Tilte: Bộ lọc dữ liệu nâng cao trong php

Map: bo-loc-du-lieu-nang-cao-trong-php

Type:PHPAD

Meta Des: Bộ lọc dữ liệu nâng cao trong php

Meta Keys: kiểm tra biến thuộc trong mang integer

Description: Bộ lọc dữ liệu nâng cao sử dụng để kiểm tra xem giá trị có thuộc khoảng giá trị nhất định

Content:

Ví dụ 1 : Kiểm tra giá trị có thuộc 1 khoảng giá trị

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
/\* variable to check \*/  
$int = 122;  
  
/\* min value \*/  
$min = 1;  
/\* max value \*/  
$max = 200;  
  
if (filter\_var($int, FILTER\_VALIDATE\_INT, array("options" => array("min\_range"=>$min, "max\_range"=>$max))) === false) {  
    echo("Variable value is not within the legal range");  
} else {  
    echo("Variable value is within the legal range");  
}  
?>  
  
</body>  
</html>

Kết quả

Variable value is within the legal range

Ví dụ 2 : Kiểm tra địa chỉ ip có phải là ipv6

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
// Variable to check  
$ip = "2001:0db8:85a3:08d3:1319:8a2e:0370:7334";  
  
// Validate ip as IPv6  
if (!filter\_var($ip, FILTER\_VALIDATE\_IP, FILTER\_FLAG\_IPV6) === false) {  
    echo("$ip is a valid IPv6 address");  
} else {  
    echo("$ip is not a valid IPv6 address");  
}  
?>  
  
</body>  
</html>

Kết quả:

2001:0db8:85a3:08d3:1319:8a2e:0370:7334 is a valid IPv6 address

Ví dụ 3:

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
// Variable to check  
$url = "https://www.CodeWR.com";  
  
// Validate url  
if (!filter\_var($url, FILTER\_VALIDATE\_URL, FILTER\_FLAG\_QUERY\_REQUIRED) === false) {  
    echo("$url is a valid URL");  
} else {  
    echo("$url is not a valid URL");  
}  
?>  
  
</body>  
</html>

Kết quả:

https://www.CodeWR.com is not a valid URL

Ví dụ 4: Xóa các kí tự đặc biết của bảng mã ASCII

<!DOCTYPE html>  
<html>  
<body>  
  
<?php  
// Variable to check  
$str = "<h1>Hello WorldÆØÅ!</h1>";  
  
// Remove HTML tags and all characters with ASCII value > 127  
$newstr = filter\_var($str, FILTER\_SANITIZE\_STRING, FILTER\_FLAG\_STRIP\_HIGH);  
echo $newstr;  
?>  
  
</body>  
</html>

Kết quả:

Hello World!

## 32. Xử lý lỗi trong php

Tilte: Xử lý lỗi trong php

Map: xu-ly-loi-trong-php

Type:PHPAD

Meta Des: Xử lý lỗi trong php

Meta Keys: xu ly loi trong php, xử lý lỗi trong php

Description: Việc hiện thì lỗi trong php rất phổ biến như thông báo email không hợp lệ….

Content:

**Lỗi xử lý PHP**

Khi tạo các ứng dụng web, xử lý lỗi là một phần quan trọng. Khi xử lý code bị thiếu thì chương trình sẽ thông báo lỗi thì web của bạn trông không chuyên nghiệp và không được bảo mật.

Chúng tôi sẽ hiển thị các phương pháp xử lý lỗi khác nhau:

* Thông báo lỗi với lệnh die()
* Custom error và error triggers
* Error reporting

**Hàm die()**

Ví dụ : Lỗi khi mở 1 file  
<?php  
$myfile = fopen("text.txt", "r");  
?>

Trong trường hợp không có file thì sẽ gây ra lỗi như sau:

**Warning**: fopen(text.txt) [function.fopen]: failed to open stream:  
No such file or directory in **C:\tutorialPHP\test.php** on line **2**

Để thông báo lỗi thì t có thể dùng thay đổi như sau:  
<?php

if(!file\_exists("text.txt")) {  
  die("File not found");  
} else {  
  $file=fopen("text.txt","r");  
}  
  
?>

Kết quả:

File not found

**Lệnh thông báo lỗi chỉnh được**

Cú pháp: error\_function(error\_level,error\_message,error\_file,error\_line,error\_context)

* error\_level(Bắt buộc) Chỉ định mức báo cáo lỗi cho lỗi do người dùng xác định. Phải là một số giá trị.
* error\_message(bắt buộc). Chỉ định thông báo lỗi cho lỗi do người dùng xác định
* error\_file. Chỉ định tên tập tin mà lỗi xảy ra
* error\_line. Chỉ định số dòng trong đó xảy ra lỗi
* error\_context. Chỉ định một mảng chứa mọi biến, và các giá trị của chúng, được sử dụng khi lỗi xảy ra

Các mức báo lỗi gồm:

2 - E\_WARNING : Lỗi cảnh báo chứ không có dừng chương trình.

8 - E\_NOTICE : Lỗi thông báo nếu có lỗi nhưng chương trình vẫn chạy

256 - E\_USER\_ERROR :Lỗi do người dùng tạo ra nghiêm trọng. Điều này giống như một E\_ERROR thiết lập bởi các lập trình viên bằng cách sử dụng PHP trigger\_error chức năng ()

512 - E\_USER\_WARNING : Cảnh báo do người dùng tạo nhưng không dừng chương trình. Điều này giống như một E\_WARNING được lập trình bằng cách sử dụng hàm PHP trigger\_error ()

1024 - E\_USER\_NOTICE :Thông báo do người dùng tạo. Điều này giống như một E\_NOTICE được lập trình bằng cách sử dụng hàm PHP trigger\_error ()

4096 - E\_RECOVERABLE\_ERROR : Lỗi này làm chết chương trình. Điều này giống như một E\_ERROR nhưng có thể bị bắt bởi một tay người dùng xác định (xem set\_error\_handler ())

8191 - E\_ALL Tất cả các lỗi và cảnh báo (E\_STRICT đã trở thành một phần của E\_ALL trong PHP 5.4)

Ví dụ tạo 1 hàm xử lý lỗi:

function customError($errno, $errstr) {  
  echo "<b>Error:<b/>[$errno] $errstr <br>";  
  echo "Ending Script";  
  die();  
}

Sẽ dùng dùm hàm **set\_error\_handler**("customError"); để tham chiếu tới hàm xử lý lỗi

Ví dụ:

<?php  
//error handler function  
function customError($errno, $errstr) {  
  echo "<b>Error:</b> [$errno] $errstr";  
}  
  
//set error handler  
set\_error\_handler("customError");  
  
//trigger error  
echo($test);  
?>

Kết quả:

**Error:** [8] Undefined variable: test

**Thông báo lỗi với trigger**

Ví dụ:

<?php  
$test=2;  
if ($test>=1) {  
  trigger\_error("Value must be 1 or below");  
}  
?>

Kết quả:

**Notice**: Value must be 1 or below  
in **C:\webfolder\test.php** on line **6**

Hàm **trigger\_error** Có thể có cái tham số sau:

* E\_USER\_ERROR - Fatal lỗi do chạy do người dùng tạo ra. Lỗi không thể khôi phục được. Thi hành kịch bản bị tạm dừng
* E\_USER\_WARNING - Cảnh báo thời gian chạy do người dùng tạo không gây tử vong. Thực hiện kịch bản không bị dừng
* E\_USER\_NOTICE - Mặc định. Thông báo chạy do người dùng tạo. Tập lệnh tìm thấy một cái gì đó có thể là lỗi, nhưng cũng có thể xảy ra khi chạy một tập lệnh bình thường

Ví dụ có tham số thông báo:

<?php  
//error handler function  
function customError($errno, $errstr) {  
  echo "<b>Error:</b> [$errno] $errstr<br>";  
  echo "Ending Script";  
  die();  
}  
  
//set error handler  
set\_error\_handler("customError",E\_USER\_WARNING);  
  
//trigger error  
$test=2;  
if ($test>=1) {  
  trigger\_error("Value must be 1 or below",E\_USER\_WARNING);  
}  
?>

Kết quả:

**Error:** [512] Value must be 1 or below  
Ending Script

Thông báo lỗi với logging

Ví dụ sẽ gửi mail thông báo lỗi:

<?php  
//error handler function  
function customError($errno, $errstr) {  
  echo "<b>Error:</b> [$errno] $errstr<br>";  
  echo "Webmaster has been notified";  
  error\_log("Error: [$errno] $errstr",1,  
  "someone@example.com","From: webmaster@example.com");  
}  
  
//set error handler  
set\_error\_handler("customError",E\_USER\_WARNING);  
  
//trigger error  
$test=2;  
if ($test>=1) {  
  trigger\_error("Value must be 1 or below",E\_USER\_WARNING);  
}  
?>

Kết quả:

**Error:** [512] Value must be 1 or below  
Webmaster has been notified

Còn mail sẽ nhận được thông tin

Error: [512] Value must be 1 or below

## 33 Ngoại lệ trong php.

Tilte: Ngoại lệ trong php

Map: ngoai-le-trong-php

Type:PHPAD

Meta Des: Ngoại lệ trong php

Meta Keys: Ngoại lệ, Exception trong php

Description: Các ngoại lệ được sử dụng để thay đổi dòng lệnh bình thường của một file nếu xảy ra lỗi xác định

Content:

**Exception cơ bản**

Ví dụ:

<?php  
//create function with an exception  
function checkNum($number) {  
  if($number>1) {  
    throw new Exception("Value must be 1 or below");  
  }  
  return true;  
}  
  
//trigger exception  
checkNum(2);  
?>

Kết quả:

**Fatal error**: Uncaught exception 'Exception'  
with message 'Value must be 1 or below' in C:\webfolder\test.php:6  
Stack trace: #0 C:\webfolder\test.php(12):  
checkNum(28) #1 {main} thrown in **C:\webfolder\test.php** on line **6**

**Sử dụng try catch**

Ví dụ:

<?php  
//create function with an exception  
function checkNum($number) {  
  if($number>1) {  
    throw new Exception("Value must be 1 or below");  
  }  
  return true;  
}  
  
//trigger exception in a "try" block  
try {  
  checkNum(2);  
  //If the exception is thrown, this text will not be shown  
  echo 'If you see this, the number is 1 or below';  
}  
  
//catch exception  
catch(Exception $e) {  
  echo 'Message: ' .$e->getMessage();  
}  
?>

Kết quá:

Message: Value must be 1 or below

Giải thích các hàm:

1. Hàm checkNum () được tạo ra. Nó kiểm tra nếu một số lớn hơn 1. Nếu đó là, một ngoại lệ được ném ra
2. Các checkNum () chức năng được gọi là trong một khối "try"
3. Trường hợp ngoại lệ trong hàm checkNum () được ném ra
4. Khối "catch" lấy ra ngoại lệ và tạo ra một đối tượng ($ e) chứa thông tin ngoại lệ
5. Thông báo lỗi từ ngoại lệ được lặp lại bằng cách gọi $ e-> getMessage () từ đối tượng ngoại lệ

**Tạo một class Exception**

Ví dụ:

<?php  
class customException extends Exception {  
  public function errorMessage() {  
    //error message  
    $errorMsg = 'Error on line '.$this->getLine().' in '.$this->getFile()  
    .': <b>'.$this->getMessage().'</b> is not a valid E-Mail address';  
    return $errorMsg;  
  }  
}  
  
$email = "someone@example...com";  
  
try {  
  //check if  
  if(filter\_var($email, FILTER\_VALIDATE\_EMAIL) === FALSE) {  
    //throw exception if email is not valid  
    throw new customException($email);  
  }  
}  
  
catch (customException $e) {  
  //display custom message  
  echo $e->errorMessage();  
}  
?>

Giải thích các hàm:

1. Lớp customException () được tạo ra như một phần mở rộng của lớp ngoại lệ cũ. Bằng cách này nó thừa hưởng tất cả các phương pháp và thuộc tính từ lớp ngoại lệ cũ
2. Hàm errorMessage () được tạo ra. Hàm này trả lại thông báo lỗi nếu địa chỉ e-mail không hợp lệ
3. Biến email $ được đặt thành chuỗi không phải là địa chỉ e-mail hợp lệ
4. Khối "try" được thực hiện và một ngoại lệ được ném ra vì địa chỉ e-mail không hợp lệ
5. Khối "catch" nắm bắt ngoại lệ và hiển thị thông báo lỗi

**Multi Exception**

Ví dụ:

<?php  
class customException extends Exception {  
  public function errorMessage() {  
    //error message  
    $errorMsg = 'Error on line '.$this->getLine().' in '.$this->getFile()  
    .': <b>'.$this->getMessage().'</b> is not a valid E-Mail address';  
    return $errorMsg;  
  }  
}  
  
$email = "someone@example.com";  
  
try {  
  //check if  
  if(filter\_var($email, FILTER\_VALIDATE\_EMAIL) === FALSE) {  
    //throw exception if email is not valid  
    throw new customException($email);  
  }  
  //check for "example" in mail address  
  if(strpos($email, "example") !== FALSE) {  
    throw new Exception("$email is an example e-mail");  
  }  
}  
  
catch (customException $e) {  
  echo $e->errorMessage();  
}  
  
catch(Exception $e) {  
  echo $e->getMessage();  
}  
?>

Giải thích:

1. Lớp customException () được tạo ra như một phần mở rộng của lớp ngoại lệ cũ. Bằng cách này nó thừa hưởng tất cả các phương pháp và thuộc tính từ lớp ngoại lệ cũ
2. Hàm errorMessage () được tạo ra. Hàm này trả lại thông báo lỗi nếu địa chỉ e-mail không hợp lệ
3. Biến email $ được đặt thành một chuỗi là một địa chỉ e-mail hợp lệ, nhưng chứa chuỗi "example"
4. Khối "try" được thực hiện và một ngoại lệ không bị ném vào điều kiện đầu tiên
5. Các điều kiện thứ hai gây nên một ngoại lệ vì e-mail chứa chuỗi "ví dụ"
6. Khối "catch" nắm bắt ngoại lệ và hiển thị thông báo lỗi chính xác

**Re-throwing exception**

Ví dụ:

<?php  
class customException extends Exception {  
  public function errorMessage() {  
    //error message  
    $errorMsg = $this->getMessage().' is not a valid E-Mail address.';  
    return $errorMsg;  
  }  
}  
  
$email = "someone@example.com";  
  
try {  
  try {  
    //check for "example" in mail address  
    if(strpos($email, "example") !== FALSE) {  
      //throw exception if email is not valid  
      throw new Exception($email);  
    }  
  }  
  catch(Exception $e) {  
    //re-throw exception  
    throw new customException($email);  
  }  
}  
  
catch (customException $e) {  
  //display custom message  
  echo $e->errorMessage();  
}  
?>

Giải thích

1. Lớp customException () được tạo ra như một phần mở rộng của lớp ngoại lệ cũ. Bằng cách này nó thừa hưởng tất cả các phương pháp và thuộc tính từ lớp ngoại lệ cũ
2. Hàm errorMessage () được tạo ra. Hàm này trả lại thông báo lỗi nếu địa chỉ e-mail không hợp lệ
3. Biến email $ được đặt thành một chuỗi là một địa chỉ e-mail hợp lệ, nhưng chứa chuỗi "example"
4. Khối "try" chứa một khối "try" khác để làm cho nó có thể ném lại ngoại lệ
5. Ngoại lệ được kích hoạt vì e-mail chứa chuỗi "example"
6. Khối "catch" nắm bắt ngoại lệ và lại ném ra một "customException"
7. Các "customException" bị bắt và hiển thị một thông báo lỗi

Exception nâng cao:

Ví dụ:

<?php  
function myException($exception) {  
  echo "<b>Exception:</b> " . $exception->getMessage();  
}  
  
set\_exception\_handler('myException');  
  
throw new Exception('Uncaught Exception occurred');  
?>

Kết quả:

**Exception:** Uncaught Exception occurred

## 34.

Tilte:

Map:

Type:

Meta Des:

Meta Keys:

Description:

Content:

## 21.

Tilte:

Map:

Type:

Meta Des:

Meta Keys:

Description:

Content:

## 21.

Tilte:

Map:

Type:

Meta Des:

Meta Keys:

Description:

Content:

# 21.

Tilte:

Map:

Type:

Meta Des:

Meta Keys:

Description:

Content:

# 21.

Tilte:

Map:

Type:

Meta Des:

Meta Keys:

Description:

Content:

# 21.

Tilte:

Map:

Type:

Meta Des:

Meta Keys:

Description:

Content:

# Sample.

Tilte:

Map:

Type:

Meta Des:

Meta Keys:

Description:

Content: