Cách khai báo biến : bắt đầu bằng $

Tên biến không thể bắt đầu bằng các kí tự đặc biệt trừ \_

Tên biến số không được đặt đầu tiên

$color  
$operating\_system  
$\_some\_variable  
$model

Biến có 3 phạm vi :

Local( cục bộ )

Global( toàn cục)

Static(tĩnh)

Biến static

Thông thường, khi một hàm được hoàn thành / thực thi, tất cả các biến của nó sẽ bị xóa. Tuy nhiên, đôi khi chúng ta muốn có một biến địa phương không được xóa. Chúng tôi cần nó cho một công việc tiếp theo.

Biến tĩnh là biến sẽ được tạo ra duy nhất một lần khi gọi hàm lần đầu tiên và nó sẽ không bị xóa đi để tạo lại khi gọi lại hàm ấy. Đây là sự khác biệt giữa biến tĩnh và biến cục bộ.

Biến tĩnh là loại biến lưỡng tính, vừa có tính chất của 1 biến toàn cục, vừa mang tính chất của 1 biến cục bộ:

* Tính chất 1 biến toàn cục: biến không mất đi khi chương trình con kết thúc, nó vẫn nằm trong ô nhớ của chương trình và được tự động cập nhật khi chương trình con được gọi lại. Giống như 1 biến toàn cục vậy.
* Tính chất 1 biến cục bộ: biến chỉ có thể được sử dụng trong chương trình con mà nó được khai báo.

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

Result :0,1,2

Không có static result : 0, 0 , 0

Tham chiếu trong biến

<?php  
$value1 = "Hello";  
$value2 =& $value1; // $value1 and $value2 both equal "Hello"  
$value2 = "Goodbye"; // $value1 and $value2 both equal "Goodbye"  
?>

Phân biệt giữa gán theo giá trị và gán theo tham chiếu

Gán theo giá trị mỗi biến tương ứng với một vùng bộ nhớ

Gán theo tham chiếu các biến có thể sử dụng chung một vùng bộ nhớ

**Superglobal Variables**

là biến siêu toàn cầu

Một số biến được xác định trước trong PHP là "superglobals", có nghĩa là chúng luôn luôn tiếp cận được, bất kể phạm vi - và bạn có thể truy cập chúng từ bất kỳ chức năng, lớp học hoặc tập tin mà không cần phải làm bất cứ điều gì đặc biệt.

* $ GLOBALS
* $ \_SERVER
* $ \_REQUEST
* $ \_POST
* $ \_GET
* $ \_FILES
* $ \_ENV
* $ \_COOKIE
* $ \_SESSION

 $ GLOBALS

$ GLOBALS là một PHP biến siêu toàn cầu được sử dụng để truy cập các biến toàn cầu từ bất cứ nơi nào trong kịch bản PHP (cũng từ bên trong chức năng hoặc các phương pháp).

Cửa hàng PHP tất cả các biến toàn cầu trong một mảng gọi là $ GLOBALS [ *index* ]. Các chỉ số giữ tên của biến.

Ví dụ dưới đây cho thấy làm thế nào để sử dụng các siêu biến toàn cầu $ GLOBALS:

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

**$\_SERVER**

$ \_SERVER Là một PHP biến siêu toàn cầu nắm giữ thông tin về tiêu đề, đường dẫn, và các địa điểm kịch bản.

<?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'];  
?>

|  |  |
| --- | --- |
| **Element/Code** | **Description** |
| $\_SERVER['PHP\_SELF'] | Returns the filename of the currently executing script |
| $\_SERVER['GATEWAY\_INTERFACE'] | Returns the version of the Common Gateway Interface (CGI) the server is using |
| $\_SERVER['SERVER\_ADDR'] | Returns the IP address of the host server |
| $\_SERVER['SERVER\_NAME'] | Returns the name of the host server (such as www.w3schools.com) |
| $\_SERVER['SERVER\_SOFTWARE'] | Returns the server identification string (such as Apache/2.2.24) |
| $\_SERVER['SERVER\_PROTOCOL'] | Returns the name and revision of the information protocol (such as HTTP/1.1) |
| $\_SERVER['REQUEST\_METHOD'] | Returns the request method used to access the page (such as POST) |
| $\_SERVER['REQUEST\_TIME'] | Returns the timestamp of the start of the request (such as 1377687496) |
| $\_SERVER['QUERY\_STRING'] | Returns the query string if the page is accessed via a query string |
| $\_SERVER['HTTP\_ACCEPT'] | Returns the Accept header from the current request |
| $\_SERVER['HTTP\_ACCEPT\_CHARSET'] | Returns the Accept\_Charset header from the current request (such as utf-8,ISO-8859-1) |
| $\_SERVER['HTTP\_HOST'] | Returns the Host header from the current request |
| $\_SERVER['HTTP\_REFERER'] | Returns the complete URL of the current page (not reliable because not all user-agents support it) |
| $\_SERVER['HTTPS'] | Is the script queried through a secure HTTP protocol |
| $\_SERVER['REMOTE\_ADDR'] | Returns the IP address from where the user is viewing the current page |
| $\_SERVER['REMOTE\_HOST'] | Returns the Host name from where the user is viewing the current page |
| $\_SERVER['REMOTE\_PORT'] | Returns the port being used on the user's machine to communicate with the web server |
| $\_SERVER['SCRIPT\_FILENAME'] | Returns the absolute pathname of the currently executing script |
| $\_SERVER['SERVER\_ADMIN'] | Returns the value given to the SERVER\_ADMIN directive in the web server configuration file (if your script runs on a virtual host, it will be the value defined for that virtual host) (such as someone@w3schools.com) |
| $\_SERVER['SERVER\_PORT'] | Returns the port on the server machine being used by the web server for communication (such as 80) |
| $\_SERVER['SERVER\_SIGNATURE'] | Returns the server version and virtual host name which are added to server-generated pages |
| $\_SERVER['PATH\_TRANSLATED'] | Returns the file system based path to the current script |
| $\_SERVER['SCRIPT\_NAME'] | Returns the path of the current script |
| $\_SERVER['SCRIPT\_URI'] | Returns the URI of the current page |

**PHP $ \_REQUEST**

được sử dụng để thu thập dữ liệu sau khi nộp mẫu HTML.

Ví dụ dưới đây cho thấy một hình thức với một trường đầu vào và một nút gửi. Khi một người dùng gửi các dữ liệu bằng cách bấm vào "Gửi", các dữ liệu mẫu được gửi đến các tập tin chỉ định trong thuộc tính hành động của <form>. Trong ví dụ này, chúng tôi chỉ đến tập tin này tự cho dữ liệu hình thức xử lý. Nếu bạn muốn sử dụng một file PHP để xử lý dữ liệu hình thức, thay thế cho rằng với tên tập tin bạn đã chọn.Sau đó, chúng ta có thể sử dụng các siêu biến toàn cầu $ \_REQUEST để thu thập các giá trị của lĩnh vực đầu vào:

<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 = $\_REQUEST['fname'];  
    if (empty($name)) {  
        echo "Name is empty";  
    } else {  
        echo $name;  
    }  
}  
?>  
  
</body>  
</html>

## PHP $ \_POST

PHP $ \_POST được sử dụng rộng rãi để thu thập dữ liệu mẫu sau khi nộp một hình thức HTML với method = "post". $ \_POST Cũng được sử dụng rộng rãi để vượt qua các biến.

Ví dụ dưới đây cho thấy một hình thức với một trường đầu vào và một nút gửi. Khi một người dùng gửi các dữ liệu bằng cách bấm vào "Gửi", các dữ liệu mẫu được gửi đến các tập tin chỉ định trong thuộc tính hành động của <form>. Trong ví dụ này, chúng tôi chỉ đến tập tin riêng của mình cho các dữ liệu hình thức xử lý. Nếu bạn muốn sử dụng một file PHP để xử lý dữ liệu hình thức, thay thế cho rằng với tên tập tin bạn đã chọn. Sau đó, chúng ta có thể sử dụng các siêu biến toàn cầu $ \_POST để thu thập các giá trị của lĩnh vực đầu vào:

## PHP $ \_GET

PHP $ \_GET cũng có thể được sử dụng để thu thập dữ liệu mẫu sau khi nộp một hình thức HTML với method = "get".

$ \_GET Cũng có thể thu thập dữ liệu được gửi trong URL.