

BÀI 2:

Cú pháp lập trình PHP



Mục tiêu bài học

- Cấu trúc lặp trong PHP
- Các lệnh điều khiển vòng lặp trong PHP
- Giới thiệu mảng trong PHP
- Chuỗi trong PHP

➤ Cấu trúc lặp (vòng lặp)

- Là những khối lệnh được thực hiện lặp đi lặp lại nhiều lần chừng nào điều kiện còn thỏa mãn. Các vòng lặp sẽ kết thúc khi điều kiện không còn thỏa mãn.
- Vòng lặp được sử dụng khi ta muốn thực hiện những tính toán, hay những tác vụ lặp lại nhiều lần.
- Khi sử dụng vòng lặp, ta sẽ khai báo các biến đếm để kiểm soát số lần lặp. Biến đếm là những biến được khai báo ở bên trong vòng lặp, và giá trị của nó sẽ thay đổi sau mỗi lần lặp, để có thể kết thúc vòng lặp.
- Biến đếm chỉ tồn tại khi vòng lặp đang hoạt động. Khi vòng lặp kết thúc thì biến đếm cũng bị hủy.

Cấu trúc lặp trong PHP

- PHP hỗ trợ 4 loại cấu trúc lặp sau:
 - Vòng lặp for
 - Vòng lặp while
 - Vòng lặp do while
 - Vòng lặp foreach

➤ Vòng lặp for

- Được sử dụng khi ta đã biết được số lần lặp. Vòng lặp for sẽ được thực hiện chừng nào điều kiện còn thỏa mãn.

➤ Cú pháp của vòng lặp for:

```
for (initialization; condition; increment){  
    code to be executed;  
}
```

➤ Vòng lặp for bao gồm 3 câu lệnh như sau:

- Lệnh khởi tạo: Được dùng để khai báo và khởi tạo biến đếm. Lệnh này được thực hiện một lần duy nhất trước khi vòng lặp được thực thi. Vòng lặp for có thể có một hoặc nhiều biến đếm.
- Lệnh điều kiện: Dùng để thể hiện điều kiện của vòng lặp. Lệnh này sẽ được kiểm tra trước mỗi lần lặp.
- Lệnh tăng/giảm: Dùng để tăng hoặc giảm giá trị của biến đếm. Lệnh này sẽ được thực hiện sau mỗi lần lặp.

➤ Chú ý: Ba lệnh này là tùy chọn, vòng lặp for có thể khuyết cả 3 câu lệnh này.

Cấu trúc lặp trong PHP

➡ Ví dụ:

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<title>Minh họa for</title>
</head>
<body>
<?php
//vong lap for duyet qua 10 so nguyen dau tien
for($i = 1; $i <= 10; $i++) {
    echo '<BR>i = ' . $i;
}
?>
</body>
</html>
```

➡ Kết quả:



```
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9
i = 10
```


➤ Vòng lặp while

- Được sử dụng khi ta không biết trước số lần lặp. Vòng lặp while sẽ được thực hiện chừng nào điều kiện còn thỏa mãn.
- Chú ý: Vòng lặp while sẽ kiểm tra điều kiện trước, sau đó mới thực hiện vòng lặp. Vì vậy, nếu điều kiện sai ngay từ đầu, vòng lặp while sẽ không thực hiện được lần nào.
- Cú pháp:

```
while (condition) {  
    code to be executed;  
}
```

Cấu trúc lặp trong PHP

➡ Ví dụ:

```
<?php
$i = 1;
//vong lap while cho phép duyệt qua 10 số nguyên đầu tiên
while($i <= 10) {
    echo '<BR>i = ' . $i;
    $i++;
}
?>
```

➡ Kết quả:

localhost:8888/demophp2018/DemoWhile2018.php

```
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9
i = 10
```


➤ Vòng lặp do while

- Cũng được sử dụng khi ta chưa biết trước số lần lặp. Về cơ bản, vòng lặp do while cũng giống như vòng lặp while.
- Khác biệt ở chỗ: vòng lặp while kiểm tra điều kiện trước rồi mới thực hiện thân của vòng lặp, nên nếu điều kiện sai ngay từ ban đầu, vòng lặp while có thể không được thực hiện lần nào.
- Còn vòng lặp do while thì thực hiện thân của vòng lặp trước rồi mới kiểm tra điều kiện. Do đó nếu điều kiện bị sai ngay từ ban đầu, vòng lặp do while vẫn được thực thi ít nhất một lần.

➤ Cú pháp:

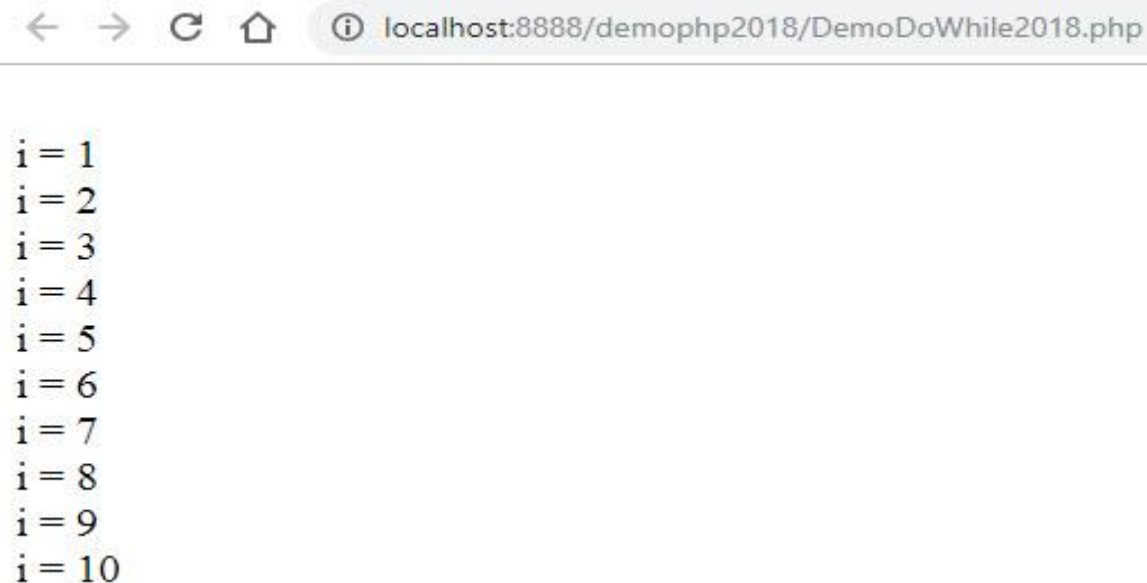
```
do {  
    code to be executed;  
}  
while (condition);
```

Cấu trúc lặp trong PHP

➔ Ví dụ:

```
<?php
$i = 1;
//vong lap do while duyet qua 10 so nguyen dau tien
do {
    echo '<BR>i = ' . $i;
    $i++;
} while($i <= 10);
?>
```

➔ Kết quả:



```
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7
i = 8
i = 9
i = 10
```

➤ Vòng lặp foreach

- Vòng lặp foreach được dùng để duyệt qua các mảng. Vòng lặp này ngắn gọn hơn so với vòng lặp for, trong đó ta chỉ cần khai báo một tên biến, rồi sử dụng tên biến đó để duyệt qua mảng.

➤ Cú pháp:

```
foreach (array as value) {  
    code to be executed;  
}
```

Cấu trúc lặp trong PHP

➡ Ví dụ:

```
<?php
//khai bao mang
$seafood = array('Tôm hùm bò lò phomai',
'Mực ống xào cần tỏi tây',
'Cua bể hấp sả',
'Lẩu cá chình tôm càng xanh bào ngư');

//duyet mang su dung foreach
foreach($seafood as $haisan) {
    echo '<BR>Món ăn: ' . $haisan;
}
?>
```

➡ Kết quả:

← → ↺ ⬆ ⓘ localhost:8888/demophp2018/DemoForEach2018.php

Món ăn: Tôm hùm bò lò phomai
Món ăn: Mực ống xào cần tỏi tây
Món ăn: Cua bể hấp sả
Món ăn: Lẩu cá chình tôm càng xanh bào ngư

Demo

- Giảng viên demo minh họa cấu trúc lặp trong PHP

- Lệnh nhảy (jump statement): Là những câu lệnh được dùng để kiểm soát luồng thực thi trong các chương trình. Lệnh nhảy được sử dụng khi ta làm việc với các vòng lặp và hàm.
- Có 3 câu lệnh nhảy như sau:
 - Lệnh break
 - Lệnh continue
 - Lệnh return

➤ Lệnh break

- Dùng để kết thúc vòng lặp. Khi được thực thi, lệnh break sẽ dừng vòng lặp, và chương trình sẽ thực thi câu lệnh ở phía sau vòng lặp.
- Lệnh break còn được sử dụng để kết thúc một case trong cấu trúc switch-case.

Lệnh nhảy

➡ Ví dụ:

```
<html>
  <body>

    <?php
      $i = 0;

      while( $i < 10) {
        $i++;
        if( $i == 3 )break;
      }
      echo ("Loop stopped at i = $i" );
    ?>

  </body>
</html>
```

➡ Kết quả:

```
Loop stopped at i = 3
```

➡ Lệnh continue

- ➡ Khi được thực thi sẽ kết thúc lần lặp hiện tại và nhảy đến lần lặp tiếp theo trong vòng lặp.
- ➡ Khi chương trình thực thi lệnh continue, các câu lệnh của lần lặp hiện tại sẽ được bỏ qua, và chương trình sẽ chuyển đến lần lặp tiếp theo.

➡ Ví dụ:

```
<html>
  <body>

    <?php
      $array = array( 1, 2, 3, 4, 5);

      foreach( $array as $value ) {
        if( $value == 3 )continue;
        echo "Value is $value <br />";
      }
    ?>

  </body>
</html>
```

➡ Kết quả:

```
Value is 1
Value is 2
Value is 4
Value is 5
```

➤ Lệnh return

- Dùng để dừng việc thực thi và kết thúc một hàm, đồng thời trả lại kết quả từ hàm.
- Điều khiển của chương trình sẽ được quay trở lại nơi mà hàm được gọi.

➡ Ví dụ:

```
<html>

<head>
  <title>Writing PHP Function which returns value</title>
</head>

<body>

  <?php
    function addFunction($num1, $num2) {
      $sum = $num1 + $num2;
      return $sum;
    }
    $return_value = addFunction(10, 20);

    echo "Returned value from the function : $return_value";
  ?>

</body>
</html>
```

➡ Kết quả:

```
Returned value from the function : 30
```


- Mảng là một cấu trúc dữ liệu dùng để lưu trữ nhiều giá trị, nhiều phần tử. Mỗi phần tử trong mảng là một biến, và các phần tử của mảng sẽ có cùng kiểu dữ liệu.
- Các phần tử của mảng sẽ được lưu ở trong các ô nhớ liên tiếp nhau trong bộ nhớ, vì vậy sử dụng mảng cho phép ta thực hiện các thao tác duyệt, sắp xếp, tìm kiếm một cách dễ dàng.
- Các phần tử của mảng sẽ được truy cập theo tên mảng, sau tên là chỉ số mảng. Chỉ số mảng là một số nguyên, và thường bắt đầu bằng 0.
- Trong PHP, để có thể tạo mảng, ta gọi hàm `array()`. Để lấy về tổng số phần tử chứa trong mảng, ta gọi hàm `count()`.

➤ PHP hỗ trợ 3 loại mảng như sau:

- **Numeric array:** Là loại mảng trong đó các phần tử được đặt các chỉ số là các số nguyên. Mặc định chỉ số của numeric array sẽ bắt đầu bằng 0.
- **Associate array (mảng kết hợp):** Cũng tương tự như numeric array nhưng khác ở chỗ chỉ số của associate array sẽ là các chuỗi.
 - Associate array sẽ lưu các phần tử dưới dạng các cặp key-value. Mỗi key sẽ đóng vai trò là các chỉ số, và key được dùng để truy cập đến các phần tử trong mảng kết hợp. Mỗi value chính là giá trị của một phần tử trong mảng kết hợp.
 - Để duyệt qua mảng kết hợp, ta sử dụng vòng lặp foreach, trong đó mỗi lần lặp ta sẽ lấy ra từng cặp key-value.

- PHP hỗ trợ 3 loại mảng như sau:
 - Multidimensional array (mảng đa chiều)
 - Thông thường ta hay sử dụng loại đơn giản nhất của mảng đa chiều là mảng 2 chiều. Mảng 2 chiều là một mảng, trong đó mỗi phần tử của nó là một mảng một chiều.
 - Mảng 2 chiều còn được gọi là mảng của các mảng. Mỗi phần tử của mảng 2 chiều sẽ được truy cập bằng 2 chỉ số, gồm chỉ số dòng và chỉ số cột.
 - Có thể tưởng tượng các phần tử của mảng 2 chiều có thể được bố trí giống như một bảng tính, bao gồm các dòng và các cột.

Mảng trong PHP

➤ Ví dụ về mảng số:

```
<body>
<?php
//khai bao mang
$drinks = array('Sinh tố dừa mè đen',
'Nước ép dưa',
'Sinh tố măng cầu',
'Nước cam nguyên chất');

//lay ve do dai mang
$length = count($drinks);

//su dung vong lap de duyet qua mang
for($i = 0; $i < $length; $i++) {
    echo '<BR>' . $drinks[$i];
}
?>
```

➤ Kết quả:

← → ↻ ⬆ ⓘ localhost:8888/demophp2018/demo%20array/DemoNumericArray.php

Sinh tố dừa mè đen
Nước ép dưa
Sinh tố măng cầu
Nước cam nguyên chất

➔ Ví dụ về mảng kết hợp:

```
<?php
//khai bao mang ket hop
$listseafoods = array('gh'=>'Ghẹ hấp sả',
    'mu'=>'Mực ống nhồi thịt',
    'lu'=>'Lẩu hải sản tôm cua bào ngư ngao sò',
    'th'=>'Tôm hùm bỏ lò phomai');

//duyet qua mang ket hop
foreach($listseafoods as $key=>$value) {
    echo '<BR>Key = ' . $key . '. Value = ' . $value;
}
?>
```

➔ Kết quả:

← → ↺ ⬆ ⓘ localhost:8888/demophp2018/demo%20array/DemoAssociateArray.php

Key = gh. Value = Ghẹ hấp sả
Key = mu. Value = Mực ống nhồi thịt
Key = lu. Value = Lẩu hải sản tôm cua bào ngư ngao sò
Key = th. Value = Tôm hùm bỏ lò phomai

➤ Ví dụ về mảng hai chiều:

```
<?php
//khai bao va khoi tao mang 2 chieu
$fruits = array(array('Durian', 'Sầu riêng', 200),
array('Pineapple', 'Dứa', 10),
array('Orange', 'Cam', 30),
array('Apple', 'Táo', 50));

//su dung vong lap for long nhau
//de duyet qua mang 2 chieu
for($row = 0; $row < 4; $row++) {
    echo '<BR>';
    for($col = 0; $col < 3; $col++) {
        echo $fruits[$row][$col] . ' ';
    }
}
?>
```

➤ Kết quả:

Durian Sầu riêng 200
Pineapple Dứa 10
Orange Cam 30
Apple Táo 50

➤ Chuỗi (String)

- Là một tập hợp các ký tự, dùng để thể hiện các xâu văn bản trong chương trình.
- PHP cung cấp nhiều hàm dựng sẵn để thao tác với chuỗi.
- Ví dụ:

```
$string_1 = "This is a string in double quotes";  
$string_2 = "This is a somewhat longer, singly quoted string";  
$string_39 = "This string has thirty-nine characters";  
$string_0 = ""; // a string with zero characters
```

Chuỗi trong PHP

- Các chuỗi được đặt trong cặp dấu nháy đơn " sẽ được coi là hằng chuỗi (literal), trong khi đó những chuỗi được đặt trong cặp dấu nháy kép sẽ thay thế các biến trong chuỗi bằng giá trị của biến.

- Ví dụ:

```
<?php
$variable = "name";
$literally = 'My $variable will not print!\\n';

print($literally);
print "<br />";

$literally = "My $variable will print!\\n";

print($literally);
?>
```

- Kết quả:

```
My $variable will not print!\\n
My name will print!\\n
```

Chuỗi trong PHP

➤ Để nối chuỗi trong PHP, ta sử dụng toán tử dấu chấm (.)

➤ Ví dụ:

```
<?php
$string1="Hello World";
$string2="1234";

echo $string1 . " " . $string2;
?>
```

➤ Kết quả:

Hello World 1234

Chuỗi trong PHP

- Một số hàm thao tác với chuỗi trong PHP
 - Hàm `strlen()`: Trả về độ dài của chuỗi.
 - Hàm `strpos()`
 - Dùng để tìm kiếm một chuỗi con, hoặc một ký tự bên trong chuỗi.
 - Nếu tìm thấy chuỗi con hoặc ký tự, hàm này sẽ trả về vị trí của kết quả đầu tiên.
 - Nếu không tìm thấy, hàm này sẽ trả về `FALSE`.

➤ Ví dụ:

```
<?php  
    echo strpos("Hello world!", "world");  
?>
```

➤ Kết quả:

6

Tổng kết

- Cấu trúc lặp trong PHP
- Các lệnh điều khiển vòng lặp trong PHP
- Giới thiệu mảng trong PHP
- Chuỗi trong PHP