**PHP BUỔI 2: BIỂU THỨC ĐIỀU KIỆN, VÒNG LẶP**

I.BIỂU THỨC ĐIỀU KIỆN

1. **Biểu thức điều kiện if trong PHP**

Cú pháp:

|  |
| --- |
| *if(bieu\_thuc){*  *//thực thi hành động*  *}* |

Nếu biểu thức đúng thì chương trình sẽ thực thi toàn bộ câu lệnh trong cặp dấu {}, nếu sai thì sẽ không làm gì

Ví dụ:

|  |
| --- |
| *$soNguyen = 10;*  *if($soNguyen > 0){*  *echo "đây là số nguyên dương";*  *}*  *// kết quả in ra màn hình: đây là số nguyên dương*  *$soNguyen = 10;*  *if($soNguyen < 0){*  *echo "đây là số nguyên âm";*  *}*  *//không in ra gì cả vì 10 > 0* |

1. **Biểu thức điều kiện if else trong PHP**

Cú pháp:

|  |
| --- |
| if(dieu\_kien){  //điều kiện đúng thì thực hiện câu lệnh ở đây  }  else{  //điều kiện sai thì thực hiện câu lệnh ở đây  } |

Ví dụ1:

|  |
| --- |
| $soNguyen = 10;  if($soNguyen >= 0){  echo "đây là số nguyên dương";  }  else{  echo "đây là số nguyên âm";  }  //kết quả in ra: đây là số nguyên dương |

Ví dụ 2:

|  |
| --- |
| $soNguyen = -10;  if($soNguyen >= 0){  echo "đây là số nguyên dương";  }  else{  echo "đây là số nguyên âm";  }  //kết quả in ra: đây là số nguyên âm |

II.VÒNG LẶP

1. **Vòng lặp While.**

Vòng lặp while cũng dùng để lặp dữ liệu như các vòng lặp khác nhưng ưu điểm của nó là có thể giải quyết những bài toán không biết chính xác số vòng lặp (for cũng có thể làm đc) và cách sử dụng nó cũng rất đơn giản.

Cú Pháp:

|  |
| --- |
| while ($dieukien) {  // code...  } |

Trong đó: **$dieukien** là điều kiện của vòng lặp. Nếu điều kiện trả về true thì vòng lặp sẽ chạy và ngược lại false thì sẽ dừng.

VD: in ra các số từ 0 đến 10 bằng vòng lặp while.

|  |
| --- |
| $i = 0; // khởi tạo biến  while ($i <= 10 /\* điều kiện dừng\*/) {  echo $i . "<br>"; // in ra số  $i++; //sau mỗi lần lặp biến $i tăng thêm 1 đơn vị  } |

Khi đó trình duyệt sẽ trả về kết quả:

|  |
| --- |
| 0  1  2  3  4  5  6  7  8  9  10 |

1. **Vòng lặp do-while.**

Đây là một vòng lặp ngược lại với các vòng lặp từ trước đến giờ, các vòng lặp khác kiểm tra điều kiện trước rồi với thực thi câu lệnh, còn vòng lặp do-while thực thi câu lệnh trước rồi mới kiểm tra điều kiện.

Cú Pháp:

|  |
| --- |
| do {  // code....  } while ($dieukien); |

Trong đó: **$dieukien** là điều kiện của vòng lặp. Nếu điều kiện trả về true thì vòng lặp sẽ chạy và ngược lại false thì sẽ dừng.

\*Chú ý:

* Phải có dấu ; ở phía sau while không chương trình sẽ báo lỗi.

VD: in ra các số từ 0 đến 10 bằng vòng lặp do while.

|  |
| --- |
| $i = 0; //khởi tạo biến $i bằng 0  do {  echo "$i <br>"; // thực hiện in ra $i hiện tại  $i++; // cộng thêm 1 đơn vị sau mỗi lần lặp  } while ($i <= 10/\* nếu $i mà lớn hơn 10 dừng vòng lặp\*/); |

1. **While, do-while lồng nhau.**

Cũng giống như vòng lặp for và foreach trong while,do-while cũng có thể lồng nhau được.

VD: vẽ tam giác vuông với while và do-while.

* **while**

|  |
| --- |
| $i = 0;  while ($i <= 10) {  $j = $i;  while ($j <= 10) {  echo "\*";  $j++;  }  echo "<br>";  $i++;  } |

* **do-while**

|  |
| --- |
| $i = 0;  do {  $j = $i;  do {  echo "\*";  $j++;  } while ($j <= 10);  echo "<br>";  $i++;  } while ($i <= 10); |

1. **Vòng lặp for.**

Đây là loại vòn lặp phổ biết nhất trong lập trình:

Cú Pháp:

|  |
| --- |
| for (bienkhoitao; dieukienthucthi; buocnhay) {  // code...  } |

Trong đó:

* **bienkhoitao**: là giá trị khởi tạo ban đầu của vòng lặp.
* **dieukienthucthi**: là điều kiện mà vòng lặp được phép chạy(chú ý: Nếu bạn muốn lặp vô tận thì có thể bỏ trống).
* **buocnhay**: là khoảng đệm nhảy của mỗi vòng lặp.

Để cho dễ hiểu chúng ta cùng xét ví dụ sau nhé:

|  |
| --- |
| for ($i = 0; $i <= 10; $i++) {  echo 'Tôi đang chạy đến số ' . $i . '<br/>';  } |

Ở bài toán trên tôi đã cho khởi tạo vòng lặp chạy từ 1 cho đến 10: và cứ mỗi một lần lặp thì in ra dòng chữ : 'tôi đang chạy đến số' + số mà vòn lặp chạy tới.

Khi chạy đoạn code trên thì kết quả trả về như sau:

|  |
| --- |
| Tôi đang chạy đến số 0  Tôi đang chạy đến số 1  Tôi đang chạy đến số 2  Tôi đang chạy đến số 3  Tôi đang chạy đến số 4  Tôi đang chạy đến số 5  Tôi đang chạy đến số 6  Tôi đang chạy đến số 7  Tôi đang chạy đến số 8  Tôi đang chạy đến số 9  Tôi đang chạy đến số 10 |

1. **Vòng lặp foreach.**

Đây là một loại vòng lặp rất là hay dùng để lặp mảng, object và chính vì điều này nên nó sẽ được sử dụng rất nhiều về sau.

Cú pháp:

|  |
| --- |
| foreach ($variable as $key => $value) {  // code...  }  ?>  hoặc nếu không muốn lặp key của mảng, object.  <?php  foreach ($variable as $value) {  // code...  } |

Trong đó:

* **$variable:** Là biến mà các bạn muốn lặp (có thể là array, object).
* **$key:** Là các key của mảng hoặc properties của object.
* **$value:** Là giá trị của mảng,object.

Để cho dễ hiểu thì mọi người cùng xem qua VD sau rồi tự đưa ra kết luận nhé.

|  |
| --- |
| //Khởi tạo mảng có key tự tăng.  $data = [  'A',  'B',  'C',  'D'  ];  // Lặp cả key lẫn value của mảng  foreach ($data as $key => $value) {  echo "Đây là phần tử có key = $key và có giá trị= $value <br/>";  }  // Tạo khoảng phân cách để dễ phân biệt  echo "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ <br/>";  // Lặp value của mảng  foreach ($data as $value) {  echo "Giá trị phần tử= $value <br/>";  } |

Và Kết quả trình duyệt trả về sẽ có dạng như sau:

|  |
| --- |
| Đây là phần tử có key = 0 và có giá trị= A  Đây là phần tử có key = 1 và có giá trị= B  Đây là phần tử có key = 2 và có giá trị= C  Đây là phần tử có key = 3 và có giá trị= D  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Giá trị phần tử= A  Giá trị phần tử= B  Giá trị phần tử= C  Giá trị phần tử= D |

1. **For và foreach lồng.**

Cũng giống như trong câu lệnh rẽ nhánh, vòng lặp cũng hoàn toàn có thể lồng nhau được.

* **For**

VD: vẽ tam giác vuông bằng vòng lặp for.

|  |
| --- |
| for ($i = 0; $i <= 10; $i++) {  for ($j = $i; $j <= 10; $j++) {  echo "\*";  }  echo "<br>";  } |

* **Foreach**

VD: duyệt mảng 2 chiều với vòng foreach lồng nhau:

|  |
| --- |
| $data = [0 => 1, 1 => [3 => 4, 5 => 6]];  foreach ($data as $val) {  if (is\_array($val)) {  foreach ($val as $v) {  echo $v . "<br/>";  }  } else  echo $val . "<br>";  } |

---/\*Hết\*/---