



Open
TO
Inspiration



Estructuras de control

if / else / elseif

if

- La sentencia de control **if**, permite la ejecución condicional de un fragmento de código.

if (expresión) sentencia

Si la expresión es true se ejecuta la sentencia.

Si la sentencia tiene más de una línea deberá estar entre llaves todas las líneas de código que se quieran ejecutar.

if / else / elseif II

if / else

- Se ejecuta junto con la sentencia de control **if**, podemos ejecutar otro fragmento de código en el caso de que la expresión evaluada sea falsa.

if (expresión)

sentencia

else

sentencia2

Si la sentencia/sentencia2 tienen más de una línea deberán estar entre llaves todas las líneas de código que se quieran ejecutar.

if / else / elseif III

if / elseif / else

- Se ejecuta junto con la sentencia de control **if**, podemos volver a realizar una nueva evaluación de la expresión2 con la que se ejecutaría otro fragmento de código en el caso de que la expresión evaluada sea verdadera.

```
if (expresión)  
    sentencia  
elseif (expresión2)  
    sentencia2  
    else  
    sentencia3
```

Si la sentencia/sentencia2/sentencia3 tienen más de una línea deberán estar entre llaves todas las líneas de código que se quieran ejecutar.

if / else / elseif IV

Ejemplos

```
<?php
$a = 1;
$b = 2;
if($a > $b)
    echo "Este código se ejecutará sí a es mayor que b";

if($a < $b)
    echo "Este código se ejecutará sí a es menor que b";
else
    echo "Este código se ejecutara cuando a no sea menor que b";

if($a == $b)
    echo "Este código se ejecutará sí a es igual que b";
elseif($a > $b)
    echo "Este código se ejecutará sí a es mayor que b";
else
    echo "Este código se ejecutara cuando a no sea igual o mayor que b";
?>
```

Switch

Definición

- La sentencia de control **switch**, es muy similar a la sentencia de control if. Pero permite comparar la misma expresión con muchos valores diferentes, ejecutando un fragmento de código para cada uno de esos valores.

```
switch (expresión) {  
    case valor1: sentencia  
        break;  
    ...  
    case valorn: sentenciaN  
        break;  
    case default: sentenciaD  
        break;  
}
```

Switch II

Ejemplo

```
<?php
$a = 35;
switch ($a){
    case 1:
        echo 'Valor 1';
        break;
    case 2:
        echo 'Valor 2';
        break;
    case 3:
        echo 'Valor 3';
        break;
    case 5:
        echo 'Valor 5';
        break;
    case 10:
        echo 'Valor 10';
        break;
    default:
        echo 'No se ha encontrado un valor y se pone el valor por defecto';
}
```

Match

Definición

- La sentencia de control **match**, es muy similar a la sentencia de control switch. Pero en este caso se realiza una comprobación de identidad (===) y no de igualdad (==). **Match está disponible a partir de PHP 8.**

```
$variable = match (expresión) {  
    'expresión' => sentencia  
    'expresion2' => sentencia2  
}
```

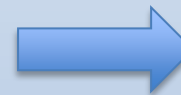

Match II

Ejemplo

```
<?php
    $edad = 25;
    $años = 30;

    echo $resultado = match ($edad){
        1, 2, 3, 4, 5 => 'Eres un bebe',
        10, 12, 15, 17 => 'Eres un niño adolescente',
        18, 19, 20, 25 => 'Eres un adulto joven',
        30, 35, 40 => 'Eres un adulto'
    };

    echo $resultadoEdad = match (true){
        $edad+$años < 10 => 'Eres un bebe',
        $edad+$años < 18 => 'Eres un adolescente',
        $edad+$años < 25 => 'Eres un adulto joven',
        $edad+$años < 50 => 'Eres un adulto',
        $edad+$años >= 50 => 'Ya no eres joven'
    };
?>
```



Eres un adulto joven
Ya no eres joven

While

Definición

- La sentencia de control **while**, es un tipo de bucle muy sencillo.

***while (expresión)
sentencia***

- Mientras la expresión sea verdadera, se ejecutará continuamente la sentencia, una y otra vez.
- Se puede detener la iteración continua del bucle mediante las palabras reservada `break`, `return` o similares que impliquen una ruptura de la sentencia y no vuelva a realizarse el análisis de la expresión.

```
<?php
$a = 10;
$b = 15;

while ($a < $b){
    echo "El valor de A es menor que el de B.";
    echo " A = ".$a."<br>";
    $a++;
}
```



El valor de A es menor que el de B. A = 10
 El valor de A es menor que el de B. A = 11
 El valor de A es menor que el de B. A = 12
 El valor de A es menor que el de B. A = 13
 El valor de A es menor que el de B. A = 14

do-While

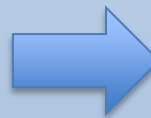
Definición

- La sentencia de control **do-while**, es un bucle muy similar al bucle while.

```
do{  
    sentencia}  
while (expresión)
```

- La diferencia con el bucle while es que en este caso se ejecuta la sentencia al menos una vez. Y se repetirá, tantas iteraciones como la expresión sea true.

```
<?php  
  
$a = 10;  
$b = 15;  
  
do{  
    $a++;  
    echo "El valor de A es menor que el de B.";  
    echo " A = ".$a."<br>";  
}while($a < $b)  
?>
```



```
El valor de A es menor que el de B. A = 11  
El valor de A es menor que el de B. A = 12  
El valor de A es menor que el de B. A = 13  
El valor de A es menor que el de B. A = 14  
El valor de A es menor que el de B. A = 15
```

for

Definición

- La sentencia de control **for**, es un bucle que permite el análisis a raíz de 3 expresiones.

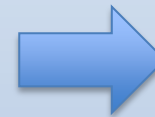
***for (expresión, expresión2, expresión3)
sentencia***

- La primera expresión es evaluada y ejecutada una vez al inicio del bucle.
- Cada iteración se evalúa la expresión2, si es verdadera se ejecuta la sentencia continuamente, hasta que sea falsa.
- Al final de cada iteración se ejecuta la expresión3.
- Indicar que la expresión y expresión3 suelen ser expresiones que ejecutan o condicionan los valores de la expresión2.

for II

Ejemplo

```
<?php
for ($i= 0; $i < 25; $i++ ){
    echo "En este caso el valor de i es igual ".$i."<br>";
}
?>
```



En este caso el valor de i es igual 0
 En este caso el valor de i es igual 1
 En este caso el valor de i es igual 2
 En este caso el valor de i es igual 3
 En este caso el valor de i es igual 4
 En este caso el valor de i es igual 5
 En este caso el valor de i es igual 6
 En este caso el valor de i es igual 7
 En este caso el valor de i es igual 8
 En este caso el valor de i es igual 9
 En este caso el valor de i es igual 10
 En este caso el valor de i es igual 11
 En este caso el valor de i es igual 12
 En este caso el valor de i es igual 13
 En este caso el valor de i es igual 14
 En este caso el valor de i es igual 15
 En este caso el valor de i es igual 16
 En este caso el valor de i es igual 17
 En este caso el valor de i es igual 18
 En este caso el valor de i es igual 19
 En este caso el valor de i es igual 20
 En este caso el valor de i es igual 21
 En este caso el valor de i es igual 22
 En este caso el valor de i es igual 23
 En este caso el valor de i es igual 24



foreach

Definición

- La sentencia de control **foreach**, es un bucle parecido al for pero que permite un modo sencillo de iterar sobre arrays. Podemos usar dos sintaxis muy parecidas que nos permiten trabajar con el contenido de los arrays.

foreach (expresión_array as \$valor)

sentencias

o

foreach(expresión_array as \$clase =>\$valor)

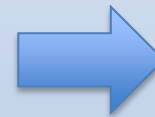
sentencias

foreach II

Ejemplo

```
<?php
$a = array();
$a[] = 1;
$a[25] = 3;
var_dump( $a);
//En este caso no podemos acceder a usar las claves del array
foreach ($a as $value){
    echo "<br>El valor del array para esta iteración es: ".$value;
}
💡 $i = 0;
//En este caso vamos a ir trabajando con los valores de cada una de las posiciones
//del array y con las claves de esas posiciones.
foreach ($a as $key=>$value){

    echo "<br>El valor en la iteración ".$i." es :".$value." para la clave:".$key;
    if($key > 10){
        echo "<br>Hemos pasado los 10 primeros valores";
    }elseif ($key > 20){
        echo "<br>Hemos pasado los 20 primeros valores";
    }elseif ($key > 30){
        echo "<br>Hemos pasado los 30 primeros valores";
    }else{
        echo "<br>El valor está entre las 10 primeros valores";
    }
    $i++;
}
```



```
array(2) { [0]=> int(1) [25]=> int(3) }
El valor del array para esta iteración es: 1
El valor del array para esta iteración es: 3
El valor en la iteración 1 es :1 para la clave:0
El valor está entre las 10 primeros valores
El valor en la iteración 2 es :3 para la clave:25
Hemos pasado los 10 primeros valores
```

Consideraciones

Break

- **break** es finaliza la ejecución de cualquier estructura: for, foreach, while, do-while o switch en curso. Puede ponerse un argumento número, que permitirá indicar de cuantas estructuras anidadas debe salir, por defecto 1.

```
<?php
$a = 35;
switch ($a){
    case 1:
        echo 'Valor 1';
        break;
    case 2:
        echo 'Valor 2';
        break;
    case 3:
        echo 'Valor 3';
        break;
    case 5:
        echo 'Valor 5';
        break;
    case 10:
        echo 'Valor 10';
        break;
    default:
        echo 'No se ha encontrado un valor y se pone el valor por defecto';
}
```


Consideraciones II

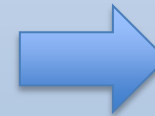
Continue

- **continue** se utiliza para saltar el resto del código de una sentencia de una iteración de un bucle y continuar su ejecución en la evaluación de la condición y poder comenzar en la siguiente iteración si corresponde. En un switch hace lo mismo que un break. Puede ponerse un argumento número, que permitirá indicar de cuantas estructuras anidadas debe saltar, por defecto 1.

```
<?php

for ($i= 0; $i < 25; $i++ ){
    if($i == 20)
        continue;
    echo "En este caso el valor de i es igual ".$i."<br>";
}

?>
```



```
En este caso el valor de i es igual 0
En este caso el valor de i es igual 1
En este caso el valor de i es igual 2
En este caso el valor de i es igual 3
En este caso el valor de i es igual 4
En este caso el valor de i es igual 5
En este caso el valor de i es igual 6
En este caso el valor de i es igual 7
En este caso el valor de i es igual 8
En este caso el valor de i es igual 9
En este caso el valor de i es igual 10
En este caso el valor de i es igual 11
En este caso el valor de i es igual 12
En este caso el valor de i es igual 13
En este caso el valor de i es igual 14
En este caso el valor de i es igual 15
En este caso el valor de i es igual 16
En este caso el valor de i es igual 17
En este caso el valor de i es igual 18
En este caso el valor de i es igual 19
En este caso el valor de i es igual 21
En este caso el valor de i es igual 22
En este caso el valor de i es igual 23
En este caso el valor de i es igual 24
```

Consideraciones III

Return

Return devuelve el control del programa al módulo que lo invoca. Si se llama desde una función, finaliza inmediatamente la función y devuelve el argumento a la llamada de la función.

```
<?php
1 usage
function funcion($a)
{
    return $a+1;
}

echo funcion( a: 3);
?>
```



4

Consideraciones IV

Include, Include_once

Include se encarga de incluir y evaluar el archivo especificado. Es una forma de incluir archivos de código PHP en el mismo archivo. Permitiendo la reutilización de código. En caso de fallo en la inclusión salta un warning y no detendrá el script.

Include_once hace lo mismo que include, pero en este caso si el archivo que se iba a incluir ya se ha incluido no lo volverá a incluir.

```

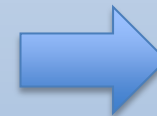
1 <?php
2 1 usage
3 function funcion($a)
4 {
5     return $a+1;
6 }
7 echo funcion( a: 3);
8

```

```

1 <?php
2 include "return.php";
3 ?>

```



4

Consideraciones V

Require, Require_once

Require es parecido a include salvo que en caso de fallo en la inclusión da un error. Y detiene el script.

Require_once hace lo mismo que require, pero en este caso si el archivo que se iba a incluir ya se ha incluido no lo volverá a incluir.

Consideraciones VI

Declare

Declare es un constructor que fija una directiva de ejecución para un determinado código.

***declare (directiva)
estamento.***

Actualmente existen 3 tipos de directivas:

- ticks
- encoding
- strict_types (A partir de PHP 7)

<https://www.php.net/manual/es/control-structures.declare.php>

Consideraciones VII

Goto

Goto es un operador que puede usarse para saltar a otras secciones del programa. Identificado mediante una etiqueta.

- No podrá tampoco saltarse fuera de una función o método o dentro si se está fuera.
- Importante tener en cuenta que no se puede saltar dentro de un bucle o switch. Pero si se puede usar para saltar fuera en lugar de un break multinivel.

<https://www.php.net/manual/es/control-structures.goto.php>

Referencias

Webgrafía

Todos los ejemplos se han sacado de la documentación oficial de PHP

- <https://www.php.net/>

Bibliografía

Existen muchos libros de PHP, pero se recomienda:

Lopez, A. (2023). *Learning PHP 7*. Packt Publishing.