Bibliografía:

Implantación de Aplicaciones Web, Jorge Sánchez Asenjo

PHP con acceso a bases de datos, Félix Mateos.

TEMA 3. ESTRUCTURAS DE CONTROL

Las instrucciones de un script PHP se ejecutan secuencialmente una detrás de otra, desde la primera hasta la última, a menos que utilicemos algún tipo de estructura de control.

Las estructuras de control nos permiten variar el flujo de ejecución secuencial y pueden ser de los siguientes tipos:

- Bifurcaciones: Ejecutan un bloque de instrucciones concreto entre varios disponibles en función de con qué valor se evalúe una expresión.
- Bucles: Ejecutan un mismo bloque de instrucciones repetidamente hasta que una expresión se evalúe con cierto valor.

Bifurcaciones

En PHP disponemos de tres estructuras de bifurcación: if, switch y operador ?

if

La instrucción if sirve para evaluar una o más expresiones y ejecutar sólo el bloque de instrucciones asociado a la expresión evaluada como TRUE

```
if(condicion_1){
    bloque_de_instrucciones_a_ejecutar_si_condicion_1_es_TRUE
}else if(condicion_2){
    bloque_de_instrucciones_a_ejecutar_si_condicion_2_es_TRUE
}
…
}else{
    bloque_de_instrucciones_a_ejecutar_si_ninguna_condición_anterior_es_TRUE
}
```

Ejemplo:

```
<?php
$curso='primero';
if($curso=='primero')
     echo '1°';
else
     echo '2°';
 ?>
```

En cuanto una de las condiciones se evalúa como TRUE se ejecuta su bloque de instrucciones asociado y ya no se evalúan las demás; en otras palabras: en una instrucción if se ejecuta a lo sumo uno de los bloques de instrucciones (si se omite la sección else puede ser que no se ejecute ninguno si no se evalúa como TRUE ninguna de las condiciones).

switch

En una bifurcación switch se evalúa una sola expresión y se ejecuta el bloque de instrucciones asociado al caso correspondiente a ese valor (o el asignado al caso opcional default si el valor no coincide con ninguno de los casos).

Las principales diferencias respecto a if son:

- Sólo se evalúa una expresión.
- La expresión evaluada no tiene que ser forzosamente de tipo lógico; puede ser numérica o de cadena de caracteres.
- Una vez localizado el caso que coincide con el valor de la expresión, se ejecutan todas las instrucciones de ese caso y los siguientes (incluido el default) hasta que se encuentre un break.

switch (*expresion*){

case *valor_caso_1*:

instrucciones_a_ejecutar_si_expresion_coincide_con_el_valor_de_este_caso

break;

case *valor_caso_2*:

instrucciones_a_ejecutar_si_expresion_coincide_con_el_valor_de_este_caso

break;

…

default:

instrucciones_a_ejecutar_si_expresion_no_coincide_con_el_valor_de_otro_caso

}

Ejemplo:

```
<?php
//Retribuciones en especie para empleados
$puesto='Jefe de obra';
switch($puesto){
  case 'Promotor':
    $regalo='automovil';
    break;
  case 'Arquitecto':
  case 'Aparejador':
    $regalo='Vacaciones en el Caribe';
    break;
  case 'Oficial':
    $regalo='Jamón serrano';
    break;
  default:
    $regalo='Botella de vino';
}
echo($regalo);?>
```

Si no se utiliza el break, la ejecución del programa continúa por el siguiente case.

Operador ternario ?

La sintaxis del operador ternario es:

condicion ? *resultado_si_condicion_TRUE* : *resultado_si_condicion_FALSE*;

Por ejemplo:

```
<?php $edad=17;
echo $edad>=18?'Mayor de edad':'Menor de edad';
?>
```

Bucles

En esta sección vamos a estudiar los bucles while, do…while, for. Todos ellos ejecutan un mismo bloque de instrucciones repetidas veces hasta que la condición se evalúa como FALSE. Cada una de estas ejecuciones se denomina iteración.

Existen dos instrucciones que permiten alterar este funcionamiento predeterminado y que son válidas para todos los tipos de bucles:

- break: Abandona el bucle.
- continue: Abandona la iteración actual y continúa la siguiente.

while

La sintaxis del bucle while es la siguiente:

while (*condicion*){

instrucciones_a_ejecutar_en_cada_iteración

}

Por ejemplo, el siguiente script nos muestra la tabla del 7:

```
<?php
    $i=0;
    while ($i<10){
        echo '7 x '.$i.' = '.($i*7).'<br />';
        $i++;
}
?>
```

do…while

Cuando tenemos un bloque de instrucciones que necesitamos ejecutar al menos una vez independientemente de cómo se evalúe la condición, lo ideal es utilizar un do_while, pues en ellos la condición de continuidad a la siguiente iteración se evalúa al finalizar la iteración, no antes de iniciarla.

La sintaxis general de do…while es:

do{

instrucciones_de_la_iteración

}while (*condición*);

Ejemplo: Mostrar todos los números pares menores de 100:

```
<?php
   $numero=0;

   do{
      echo $numero;
      $numero=$numero+2;
   }while($numero<100);

?>
```

for

La sintaxis general del bucle for es la siguiente:

for(*expresion_inicial*;*expresion_de_continuidad*,*expresión_tras_cada_iteración*){

instrucciones_de_la_iteración

}

Y su ejecución se realiza en los siguientes pasos:

1. Se evalúa la expresión inicial.
2. Si la *expresión_de_continuidad* se evalúa como TRUE se inicia la ejecución de las instrucciones de la iteración, si no se abandona el bucle.

3. Al terminar la iteración se ejecuta la *expresión_tras_cada_iteración* y, a continuación, se vuelve al paso 2.

Por ejemplo, script que escribe la tabla del 7:

```php
<?php
   for( $i=0;$i<=10; $i++)

       echo '7 x '.$i.' = '.($i*7).'<br />';
   }
?>
```