

## TEMA 2

### PHP BUCLES

#### 1. ESTRUCTURAS DE CONTROL ITERATIVAS. BUCLES

##### SENTENCIAS DE ITERACIÓN . BUCLES

Podemos decir, que un bucle es una repetición de una acción o conjunto de acciones hasta que se cumpla una condición o situación final que se ha definido en la cabecera del bucle.

Hay que tener en cuenta que con la utilización de los bucles, tenemos que tener mucha cura en su definición, puesto que PHP es un lenguaje interpretado que se ejecuta en la parte del servidor, un bucle infinito podría llegar a colgar el servidor web.

##### FOR

```
for (expr1; expr2; expr3)
{
    Sentencias;
}
```

Dónde:

**expr1:** inicializa la variable que usaremos en el bucle **for**.

**expr2:** define la condición que tiene que cumplirse para poder entrar en el proceso **for**; en el momento en que no se cumpla se acaba el proceso.

**expr3:** modifica el valor de la variable usada como contador del bucle, puede incrementar, decrementar, etc.

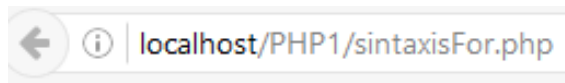
Ejemplo:

*sintaxisFor.php*

```
<?php
print("<ul>\n");

for ($i=1; $i<=5; $i++)
{
    print("<li>Elemento $i</li>");
}
print("</ul>\n");
?>
```

Salida:



- Elemento 1
- Elemento 2
- Elemento 3
- Elemento 4
- Elemento 5

## **FOREACH** Se verá en apartado Arrays

La construcción **foreach** es la más sencilla cuando el bucle esté relacionado con un array, aunque los arrays los veremos en temas posteriores, no resultará complicado entender esta estructura.

Cuando foreach inicia su ejecución, el puntero interno del array se pone automáticamente en el primer elemento del array.

Esto significa que no es necesario decir la función *reset()* antes de un bucle foreach.

- Sintaxis 1:

```
foreach(expresion_array as $value)
{
    Sentencias
}
```

## Ejemplo: Se verá en apartado Arrays

- Sintaxis 2:

```
foreach (expresión_array as $clave => $valor)
{
    Sentencias;
}
```

Le pasamos un array como primer parámetro y una variable como segundo, a cada iteración del bucle la variable irá cogiendo un valor del array, avanzando al siguiente elemento en cada pase, hasta llegar al final del array.

Ejemplo: Se verá en apartado Arrays  
*sintaxisForeachAsKeyValue.php*

```
<?php
$array = array('a'=>1, 'b'=>2, 'c'=>3, 'd'=>4);
foreach ($array as $clave => $valor)
{
    echo "$clave => $valor";
    echo "<br>";
}
?>
```

**Ejemplo: Se verá en apartado Arrays**

KeyValue por Referencia

Ver ejemplo *SintaxisForeachAsValueReferencia.php***Ejemplo: Se verá en apartado Arrays***arrayForeachDiaSemana.php*

```
<?php
// Definimos un array con los dias de la semana
$dia[0] = "Lunes";
$dia[1] = "Martes";
$dia[2] = "Miércoles";
$dia[3] = "Jueves";
$dia[4] = "Viernes";
$dia[5] = "Sabado";
$dia[6] = "Domingo";
// Y ahora se muestran por pantalla
echo "Los días de la semana son:<br><br>\n";
foreach ($dia as $valor)
{
    echo "$valor<br>\n"; // //en cada pasada se lee un nuevo valor del array
}
?>
```

Ejemplo: Se verá en apartado Arrays

*arrayForeachColores.php:*

```
<?php

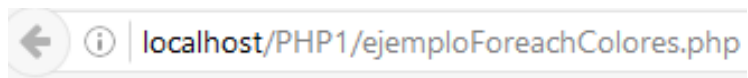
$color = array ("rojo"=>101, "verde"=>51, "azul"=>255);

foreach ($color as $valor)
]   print "Valor: $valor <BR>";

foreach ($color as $clave => $valor)
]   print "Clave: $clave; Valor: $valor<BR>";

?>
```

Salida:



Valor: 101

Valor: 51

Valor: 255

Clave: rojo; Valor: 101

Clave: verde; Valor: 51

Clave: azul; Valor: 255

## WHILE

### Sintaxis:

```
while (condición)
{
    Sentencias;
}
```

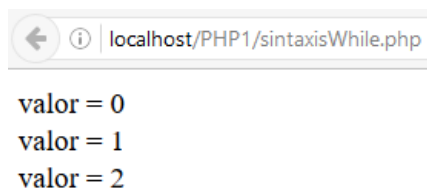
### Ejemplo:

*sintaxisWhile.php:*

```
<?php
$valor = 0;

// mientras el valor sea menor que 3 imprime el valor
while ($valor < 3)
{
    echo 'valor = ' . $valor . '<br>';
    $valor++; // incrementa 1
}
?>
```

### Salida:



valor = 0  
valor = 1  
valor = 2

## DO WHILE

Se trata de un **While** en el cual la primera entrada al bucle no se controla, es decir, siempre se produce. Algunos lenguajes lo implementan como un **Repeat-Until**.

### Sintaxis:

```
do
{
    Sentencias;
} while (condición)
```

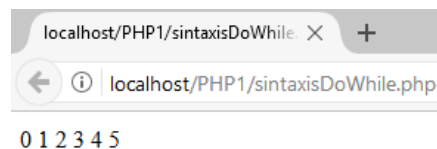
Ejemplo:

*sintaxisDoWhile.php:*

```
<?php
$num = 0;
$fin=5;

do
{
    echo $num;
    echo ' ';
    $num++;
} while ($num <= $fin);
?>
```

salida:



localhost/PHP1/sintaxisDoWhile.php X +

localhost/PHP1/sintaxisDoWhile.php

0 1 2 3 4 5



## REUTILIZAR CÓDIGO. INCLUDE I REQUIRE

Funciones `include()` y `require()` → <http://php.net/manual/es/function.include.php>

- La sentencia *include* sirve para incluir y evaluar el archivo especificado entre comillas.

**Nota:** existe [require](#) es idéntico a `include` excepto que ***require***, en caso de fallo **detiene** el script, mientras que [include](#) solo emitirá una advertencia (**E\_WARNING**) y permite que continúo la carga de la página php.

Ejemplo:

### Ejemplo #1 Ejemplo básico de *include*

```
vars.php
<?php

$color = 'verde';
$fruta = 'manzana';

?>

test.php
<?php

echo "Una $fruta $color"; // Una

include 'vars.php';

echo "Una $fruta $color"; // Una manzana verde

?>
```

- Inicialmente las variables *\$fruta* y *\$color* no tienen contenido, por eso se escribe solo la cadena:  
Una
- Después, al hacer *include 'vars.php'*, estas variables ya toman un valor y por eso, se escribe la cadena  
Una manzana verde

## Uso de include () para generar cabeceras y pies de la página web:

Ver archivo *include\_cabecerapie.php*

```
<html>
<head>
|   <title>Usando Include PHP</title>
</head>
<body>
|   <?php include './include_header.php' ?>
|   <h1>Aquí colocamos el cuerpo del sitio web</p></h1>
|   <?php include './include_footer.php' ?>
</body>
</html>
```

Ver archivo *include\_header.php*

```
<html>
<head>
|   <title>Usando Include PHP</title>
</head>
<body>
|   <body>
|       <?php
|           //echo "Incluimos el archivo header"
|           ?>
|       <h2>Esta es la Header de mi página</h2>
|   </body>
</html>
```

Ver archivo *include\_footer.php*

```
<html>
<head>
  <title>Usando Include PHP</title>
</head>
<body>
  <body>
    <?php
      //echo "Incluimos el archivo Footer"
    ?>
    <h3>Este es el Footer de mi página</h3>
  </body>
</html>
```

## BUCLES ANIDADOS

### Ejemplo de FOR anidado

Archivo *ForAnidado.php*

```
$numFilas=3;  
$numColumnas=4;  
  
for ($i=0 ; $i < $numFilas; $i++)  
{  
    for ($j= 0; $j < $numColumnas; $j++)  
    {  
        echo $i.$j.' ' ;  
    }  
    echo '<br>';  
}
```

#### Nota:

Se verán otras maneras de implementar bucles anidados más adelante.

## 2.- REUTILIZAR CÓDIGO. INCLUDE Y REQUIRE

Este apartado se verá con más detalle en el tema de Funciones

Funciones `include()` y `require()` → <http://php.net/manual/es/function.include.php>

- La sentencia `include` sirve para incluir y evaluar el archivo especificado entre comillas.

**Nota:** existe `require` es idéntico a `include` excepto que **`require`**, en caso de fallo **detiene** lo script, mientras que `include` solo emitirá una advertencia (**`E_WARNING`**) y permite que continúo la carga de la página php.

Ejemplo:

### Ejemplo #1 Ejemplo básico de `include`

```
vars.php
<?php

$color = 'verde';
$fruta = 'manzana';

?>

test.php
<?php

echo "Una $fruta $color"; // Una

include 'vars.php';

echo "Una $fruta $color"; // Una manzana verde

?>
```

- Inicialmente las variables `$fruta` y `$color` no tienen contenido, por eso se escribe solo la cadena:  
Una
- Después, al hacer `include 'vars.php'`, estas variables ya toman un valor y por eso, se escribe la cadena  
Una manzana verde

