

PROGRAMA DE ESTUDIOS

DESARROLLO DE SISTEMAS DE INFORMACIÓN

**DESARROLLO DE
BACKEND**

Tema

**MANEJO DE LAS ESTRUCTURAS
REPETITIVAS DO...WHILE Y FOREACH
EN EL LENGUAJE PHP.**

MANEJO DE LAS ESTRUCTURAS REPETITIVAS DO..WHILE Y FOREACH EN EL LENGUAJE PHP.

En PHP, además de los bucles for y while, tienes otras estructuras repetitivas muy útiles: do...while y foreach. Cada una tiene un propósito específico y es crucial saber cuándo usarla.

EL BUCLE DO...WHILE: Repetir al menos una vez

El bucle do...while es muy similar al while, con una diferencia clave: la condición se evalúa al final del bucle. Esto significa que el código dentro del bloque do siempre se ejecutará al menos una vez, incluso si la condición es falsa desde el principio.

Sintaxis

```
do {  
    // Código que se ejecuta en cada iteración  
} while (condicion);
```

Ejemplo:

Imagina que necesitas pedir al usuario que ingrese un número mayor que 10. Con un do...while, puedes asegurarte de que el mensaje se muestre al menos una vez antes de verificar la condición.

```
<?php  
$numero = 5;  
  
do {  
    echo "El número es: " . $numero . "\n";  
    $numero++;  
} while ($numero > 10);  
// Salida: El número es: 5  
?>
```

En este caso, aunque la condición `$numero > 10` es falsa, el bucle ejecuta el bloque do una vez y luego se detiene.

CUANDO USAR DO...WHILE:

Este bucle es ideal para situaciones donde necesitas garantizar que el bloque de código se ejecute al menos una vez, como en menús interactivos, validaciones de entrada de datos o cualquier proceso de inicialización que deba ocurrir antes de la primera verificación de la condición.

EL BUCLE FOREACH: Recorrer arrays y objetos

El bucle foreach es una de las estructuras más potentes y útiles de PHP. Está diseñado específicamente para iterar sobre elementos de arrays y objetos de forma sencilla y legible, sin la necesidad de un contador manual.

Sintaxis

Hay dos formas principales de usar foreach:

1.-Iterar sobre los valores:

```
foreach ($array as $valor) {  
    // Código que se ejecuta para cada valor  
}
```

2.-Iterar sobre claves y valores:

```
foreach ($array as $clave => $valor) {  
    // Código que se ejecuta para cada par clave-valor  
}
```

Ejemplos:**1.-Recorriendo los valores de un array simple:**

```
<?php  
$frutas = ["Manzana", "Banana", "Cereza"];  
  
echo "Mis frutas favoritas son: \n";  
foreach ($frutas as $fruta) {  
    echo "- " . $fruta . "\n";  
}  
  
/* Salida:  
Mis frutas favoritas son:  
- Manzana  
- Banana  
- Cereza  
*/  
?>
```

2.- Recorriendo un array asociativo (clave y valor):

```
<?php
$puntuaciones = [
    "Juan" => 95,
    "Ana"  => 88,
    "Luis" => 76
];

echo "Puntuaciones de estudiantes: \n";
foreach ($puntuaciones as $estudiante => $puntuacion) {
    echo $estudiante . " obtuvo una puntuación de " . $puntuacion . ".\n";
}
/* Salida:
Puntuaciones de estudiantes:
Juan obtuvo una puntuación de 95.
Ana obtuvo una puntuación de 88.
Luis obtuvo una puntuación de 76.
*/
?>
```

CUANDO USAR FOREACH:

Usa foreach siempre que necesites trabajar con todos los elementos de un array. Es más seguro que un bucle for con un contador, ya que evita errores por "salirse de los límites" del array. También es mucho más legible y claro para este propósito.

Resumen de las estructuras repetitivas

Característica	for	while	do...while	foreach
Uso Principal	Repetir un número fijo de veces.	Repetir mientras una condición sea <code>true</code> .	Repetir al menos una vez.	Iterar sobre arrays y objetos.
Condición	Al inicio del bucle.	Al inicio del bucle.	Al final del bucle.	Se encarga automáticamente de la iteración.
Cuándo usarlo	Con contadores, cuando sabes el número de iteraciones.	Cuando la cantidad de repeticiones es desconocida.	Cuando el código debe ejecutarse sí o sí al menos una vez.	Para recorrer todos los elementos de una colección.

