

Estructuras de control

- ♦ Las estructuras de control están formadas por los condicionales y los bucles.
- ♦ Dentro de los condicionales tenemos
 - ♦ if
 - ♦ switch
- ♦ Dentro de los bucles tenemos
 - ♦ for
 - ♦ do...while
 - ♦ while
 - ♦ foreach

Condicionales

La estructura en PHP del IF es la siguiente:

Básica

```
if (condición) {  
    instrucciones;  
} else {  
    instrucciones;  
}
```

Anidada

```
if (cond1) {  
    instrucciones;  
} elseif (cond2) {  
    instrucciones;  
}  
...  
} else {  
    instrucciones;  
}
```


Condicionales, ejemplos

- Todo código:

```
<?php
if($nombre = "Pepe") {
    echo "Tu nombre es Pepe";
}else{
    echo "Tu no eres Pepe";
}
?>
```

- Esquema más eficiente:

```
<?php if($nombre = "Pepe"){ ?>
    <p>Tu nombre es Pepe<p>
<?php }else{ ?>
    <p>Tu no eres Pepe<p>
<?php } ?>
```

- Esquema simplificado de sintaxis:

```
<?php if($nombre = "Pepe"): ?>
    <p>Tu nombre es Pepe<p>
<?php else: ?>
    <p>Tu no eres Pepe<p>
<?php endif; ?>
```


Condicionales

La estructura en PHP del switch es:

```
switch (variable) {  
    case valor1:  
        instrucciones;  
        break;  
    case valor2:  
        instrucciones;  
        break;  
    ...  
    case valor:  
        instrucciones;  
        break;  
    default:  
        instrucciones;  
        break;  
}
```


Condicionales

Aspectos a tener en cuenta:

- En los if anidados podemos separar el **elseif** dejándolo como **else if**
- Dentro de la condición podemos utilizar operadores comparativos y lógicos.

Operadores de comparación

Ejemplo	Nombre	Resultado
\$a == \$b	Igual	TRUE si \$a es igual a \$b después de la manipulación de tipos.
\$a === \$b	Idéntico	TRUE si \$a es igual a \$b, y son del mismo tipo. (a partir de PHP 4)
\$a != \$b	Diferente	TRUE si \$a no es igual a \$b después de la manipulación de tipos.
\$a <> \$b	Diferente	TRUE si \$a no es igual a \$b después de la manipulación de tipos.
\$a !== \$b	No idéntico	TRUE si \$a no es igual a \$b, o si no son del mismo tipo. (a partir de PHP 4)
\$a < \$b	Menor que	TRUE si \$a es estrictamente menor que \$b.
\$a > \$b	Mayor que	TRUE si \$a es estrictamente mayor que \$b.
\$a <= \$b	Menor o igual que	TRUE si \$a es menor o igual que \$b.
\$a >= \$b	Mayor o igual que	TRUE si \$a es mayor o igual que \$b.

Operadores	ejemplo	Uso
&&	(7>2)&&(2<4)	Devuelve verdadero cuando ambas condiciones son verdaderas, "and"
	(7>2) (2<4)	Devuelve verdadero cuando al menos una de las dos es verdadera, "Or"
!	!(7>2)	Niega el valor de la expresion

Estructuras de repetición

Dentro de las estructuras de repetición tenemos:

- while

El bucle while tiene la siguiente estructura en PHP:

```
while (condicion) {  
    instrucciones;  
}
```

- do...while

Y su estructura es:

```
do {  
    instrucciones;  
} while (condicion);
```


Estructuras de repetición

Más estructuras de repetición:

- for

Cuya estructura en PHP es:

```
for ($i=inicio; $i<fin; $i=$i+variacion) {  
    instrucciones;  
}
```

Donde

- \$i es la variable de control del bucle
- inicio es el valor donde comienzan las repeticiones
- fin es el valor que permite terminar las repeticiones
- variación es el incremento o decremento de cada uno de los pasos del bucle, por lo que puede ser tanto positivo como negativo.

Estructuras de repetición

Y la última estructura de repetición:

- foreach

Siendo su estructura en PHP:

```
foreach ($matriz as $clave => $valor) {  
    instrucciones;  
}
```

Donde

- \$matriz es el array que vamos a recorrer
- \$clave es el índice del array
- \$valor es el valor que contiene el array
- Su uso habitual es para mostrar el contenido del array, solo su valor o la pareja clave-valor. Si queremos operar con él mejor se usa un bucle for.

Break y Continue

- **break** permite salir del bucle donde se encuentre
- **break n** sale de tantos bucles como valor tenga n
- **continue** salta una iteración pero no se sale del bucle
- Observacion: **goto** se vuelve a utilizar a partir de PHP5, pero no es nada recomendable

Sintaxis alternativa

Tal y como hemos visto para el condicional, disponemos de una sintaxis alternativa que nos permite separar el código HTML del código de PHP, que aunque ahora no la utilicemos habitualmente, si lo haremos más adelante.

Básicamente consiste en la eliminación de las llaves de apertura y cierre, siendo sustituidas respectivamente por ':' y por 'endif;', 'endwhile;', 'endfor;', 'endforeach;' o 'endswitch;' según corresponda.

Veamos algunos ejemplos a continuación.

Sintaxis alternativa

Sintaxis tradicional:

```
<?php
foreach ($nombres as $nombre) {
    echo "<li>$nombre</li>";
}
?>
```

Sintaxis alternativa (más eficiente)

```
<?php foreach ($nombres as $nombre): ?>
    <li>$nombre</li>
<?php endforeach; ?>
```