

Tema 1 : Estructuras de control



Ciclo Superior DAW

Asignatura: Desarrollo web en entorno servidor

Curso 20/21



Introducción

- En este capítulo veremos aspectos como:
 - Utilizar diversos tipos de variables y operadores disponibles en el lenguaje.
 - Utilizar mecanismos de decisión en la creación de bloques de sentencias.
 - Utilizar bucles y verificar su funcionamiento.
 - Añadir comentarios al código.



Estructuras. Variables

- Las variables en PHP son bastante diferentes de otras lenguas compiladas como C y Java.
- Se usan las variables sin declararlas ni especificar su tipo, la variable se adapta al tipo del dato que se le guarda.
- Ejemplos:
 - `$PI = 3.14;`
 - `$radio = 5;`



Estructuras. Variables

- Vienen precedidas con un \$
- Pueden empezar con una letra (A- Z a- z) o _ (guión bajo)
- A continuación, pueden contener cuantos caracteres alfanuméricos y guiones bajos queramos.



Actividad 1

*Declararemos 3 variables diferentes, asegurándonos que su nombre es correcto,
siguiendo las normas que comentamos anteriormente*



Estructuras. Operadores aritméticos

Operador	Función
■ +	Suma dos valores numéricos
■ −	Resta dos valores numéricos
■ *	Multiplica dos valores numéricos
■ /	Divide dos valores numéricos
■ ♂	Resto de dividir dos valores
■ ++	Aumenta un valor numérico en una unidad
■ --	Decrece un valor numérico en una unidad



Estructuras. Operadores de asignación

Operador	Función
■ =	Asigna el valor de la derecha a la variable de la izquierda
■ +=	Suma o valor de la derecha con el de la izquierda, y asigna el total al de la izquierda
■ -=	Resta el valor de la derecha al de la izquierda, y asigna el resultado al de la izquierda
■ *=	Multiplica el valor de la derecha con el de la izquierda, y asigna el resultado al de la izquierda
■ /=	Divide el valor de la izquierda entre el de la derecha, y asigna el resultado al de la izquierda
■ %=	Divide el valor de la izquierda entre el de la derecha, y asigna el resultado del resto al de la izquierda
■ . =	Concatena la cadena de la derecha con la izquierda, y asigna el resultado a la de la izquierda



Estructuras. Operadores condicionales

Operador	Función
■ ==	Comprueba si dos cadenas son idénticas
■ !=	Comprueba si dos cadenas son diferentes
■ <	Comprueba si el valor de la izquierda es menor que el de la derecha
■ >	Comprueba si el valor de la izquierda es mayor que el de la derecha
■ <=	Comprueba si el valor de la izquierda es menor o igual que el de la derecha
■ >=	Comprueba si el valor de la izquierda es mayor o igual que el de la derecha



Estructuras. Operadores lógicos

Operador	Función
▪ and	Si ambos valores se cumplen es verdadero
▪ or	Si uno o todos los valores se cumplen es verdadero
▪ xor	Si solo uno de los valores se cumple es verdadero
▪ &&	Idéntico a and
▪	Idéntico a or



Actividad 2

Define una operación para cada tipo de operadores utilizando dos o más variables



Estructuras. If

- Permite la ejecución condicional de fragmentos de código.
- Si la expresión se evalúa como TRUE, PHP ejecutará la sentencia, y si se evalúa como FALSE, la ignorará.
- Sintaxis:

```
<?php  
if (expr)  
    sentencia  
?>
```



Estructuras. If

- En este ejemplo, veremos cómo comparar dos variables.

```
<?php
if ($a > $b)
{
    print "a es mayor que b";
    $b = $a;
}
?>
```

- Las sentencias if se pueden anidar indefinidamente dentro de otras sentencias if



Estructuras. Else

- Extiende una sentencia If para ejecutar una sentencia en caso de que la expresión de If sea FALSE.
- Ejemplo:

```
<?php
if ($a > $b)
{
    print "a es mayor que b";
} else
{
    print "a NO es mayor que b";
}
?>
```



Actividad 3

Tenemos dos hermanos.

Carlos ha nacido en 1990, por lo que tiene 30 años, mientras que Marcos ha nacido en el 2000.

Define dos variables y compara sus edades para ver qué hermano es mayor.



Estructuras. Elseif

- Es una combinación de if y else.
- A diferencia de Else, se ejecutará esa expresión alternativa solamente si la expresión condicional elseif se evalúa como TRUE.
- Ejemplo:

```
<? php
    if ($a > $b) {
        print "a es mayor que b";
    } elseif ($a == $b) {
        print "a es igual que b";
    } else {
        print "a es mayor que b";
    }
?>
```



Actividad 4

Además, también tienen una hermana Sandra, que es melliza de Carlos, por lo que comparten edad.

Define una variable nueva y muestra qué hermano es el mayor.



Estructuras. While

- Esta estructura le dice a PHP que ejecute la(s) sentencia(s) anidada(s) repetidamente, mientras la expresión while se evalúa como TRUE.
- Sintaxis:

```
while (expr) sentencia
```



Estructuras. While

- En el siguiente ejemplo, mostraremos por pantalla la secuencia numérica del 1 al 10 de dos maneras diferentes.
- Ejemplo:

```
<?php
/* ejemplo 1 */
$i = 1;
while ($i <= 10) {
    print $i++;
}
/* ejemplo 2 */
$i = 1;
while ($i <= 10):
    print $i;
    $i++;
endwhile;
?>
```



Actividad 5

Utilizando las variables de edad de los hermanos, define un bucle para simular el paso del tiempo, haciendo que se muestre en cada iteración la edad de cada hermano.

Finaliza el bucle cuando el hermano mayor se jubile (65 años)



Estructuras. Do...while

- Son muy similares a los bucles while, excepto que las condiciones se comprueban al final de cada iteración en vez de al principio.
- Ejemplo:

```
<?php
    $i = 0;
    do {
        print $i;
    } while ($i>0);
?>
```



Actividad 6

Repita la actividad 5 utilizando la estructura Do While



Estructuras. For

- Son de los bucles más comunes de programación
- Al principio del bucle evalúa la primera expresión, al inicio de cada iteración se evalúa la segunda expresión y al final de cada iteración se evalúa la tercera.
- Sintaxis:

```
for (expr1; expr2; expr3) sentencia
```



Estructuras. For

- La primera expresión (expr1) se evalúa incondicionalmente una vez al principio del bucle.
- Al inicio de cada iteración, se evalúa expr2.
 - Si se evalúa como TRUE, el bucle continúa y las sentencias anidadas se ejecutan.
 - Si se evalúa como FALSE, la ejecución del bucle finaliza.
- Al final de cada iteración, se evalúa expr3.



Estructuras. For

- Ejemplos para mostrar la secuencia de 1 a 10

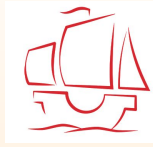
```
<?php

/* ejemplo 1 */
for ($i = 1; $i <= 10; $i++) {
    print $i;
}

/* ejemplo 2 */
for ($i = 1; ;$i++) {
    if ($i > 10) {
        break;
    }
    print $i;
}

/* ejemplo 3 */
$i = 1;
for (;;) {
    if ($i > 10) {
        break;
    }
    print $i;
    $i++;
}

/* ejemplo 4 */
for ($i = 1; $i <= 10; print $i, $i++);
?>
```

Actividad 7

Repite la actividad 5 utilizando la estructura For



Estructuras. Switch

- Se quiere comparar la misma variable con muchos valores para ejecutar una parte de código distinta dependiendo del resultado.
- Parecido al Elseif
- Ejemplo:

```
<?php
if ($i == 0) {
    print "i igual a 0";
} elseif ($i == 1) {
    print "i igual a 1";
} elseif ($i == 2) {
    print "i igual a 2";
}

switch ($i) {
    case 0:
        print "i igual a 0";
        break;
    case 1:
        print "i igual a 1";
        break;
    case 2:
        print "i igual a 2";
        break;
}
?>
```



Estructuras. Switch

- La sentencia switch ejecuta línea por línea.
- Solo cuando se encuentra una sentencia **case** con un valor que coincide con el valor de la expresión switch se comienza a ejecutar las sentencias.
- PHP continúa ejecutando las sentencias hasta el final del bloque switch, o la primera vez que vea una sentencia **break**.
- **Si no se escribe una sentencia break** al final de una lista de sentencias

case, PHP seguirá ejecutando las sentencias del siguiente case



Estructuras. Switch

- En este ejemplo, ¿qué sentencias se ejecutarán? ¿Es correcto?
- Ejemplo:

```
<?php
switch ($i) {
    case 0:
        print "i es igual a 0";
    case 1:
        print "i es igual a 1";
    case 2:
        print "i es igual a 2";
}
?>
```



Actividad 8

*Reescribe el problema de las edades de los hermanos utilizando la cláusula
Switch*



Estructuras. Break

- Permite salir de los bucles for, while, o switch.
- Acepta un parámetro opcional, lo cual determina cuántas estructuras de control hay que escapar.



Estructuras. Break

```
<?php
$arr = array ('one', 'two', 'three', 'four', 'stop', 'five');
while (list (, $val) = each ($arr)) {
    if ($val == 'stop') {
        break; /* Se podría escribir 'break 1;' */
    }
    echo "$val<br>\n";
}
/* Usando el argumento opcional. */
$i = 0;
while (++$i) {
    switch ($i) {
        case 5:
            echo "Al 5<br>\n";
            break 1; /* Sale sólo del switch. */
        case 10:
            echo "Al 10; saliendo<br>\n";
            break 2; /* Sale del switch y del bucle. */
        default:
            break;
    }
}
?>
```



Actividad 9

Utiliza una de las estructuras de bucle que hemos visto para mostrar la secuencia del 1 al 10, pero cuando llegue al 8, finaliza el bucle.



Estructuras. Continue

- Se usa dentro del bucle para saltar el resto de la iteración actual del bucle y continuar la ejecución al inicio de la siguiente iteración.
- Acepta un parámetro opcional, lo cual determina cuántos niveles (bucles) hay que saltar



Estructuras. Continue

```
<?php

while (list ($key, $value) = each ($arr)) {
    if (!($key % 2)) { // saltar pares
        continue;
    }
    muestra_numeros_impares($value);
}
?>
```



Actividad 10

Utiliza una de las estructuras de bucle que hemos visto para mostrar la secuencia del 1 al 10, pero solamente muestra los números pares.



Estructuras. Return

- Sirve para terminar la ejecución de una función y retornar su argumento como valor de la función.
- También terminará la ejecución de una sentencia `eval()` o un script PHP.
- Los veremos más adelante cuando veamos las funciones



Estructuras. Foreach

- Es una instrucción que nos permite realizar bucles For anidados de una manera más sencilla
- También es un modo fácil de iterar sobre matrices
- Recorre la matriz seleccionando cada elemento
- La veremos más a fondo en el siguiente tema



Comentarios

- Se pueden escribir comentarios de tres maneras diferentes:

- Lenguaje C

```
/* Este es un comentario como en C  
* que puede tener líneas  
* múltiples hasta el cierre  
*/
```

- Lenguaje C++

```
// Este es un comentario de una sola línea
```

- Shell

```
# Este es un comentario que acaba al final de la línea
```

Tema 1 : Estructuras de control



Ciclo Superior DAW

Asignatura: Desarrollo web en entorno servidor

Curso 20/21