

## Switch case

es un control de flujo de lectura muy práctico para cuando hay que evaluar varias opciones hay dos sintaxis

### edad!

```
Switch (condición a evaluar){
    Case valor 1:
        Código;
        Break;
    Case valor 2:
        Código;
        Break;
    Case valor 3:
        Código;
        Break;
    Default:
    }
```

```
Switch (condición a evaluar):
    Case valor 1:
        Código;
        Break;
    Case valor 2:
        Código;
        Break;
    Case valor 3:
        Código;
        Break;
    Default:
endswitch
```

vamos a dar acceso en caso de que se llame Juan / María o Pedro

vamos a evaluar 2 condiciones a la vez

1. hacemos el formulario para que pida Nombre y contraseña

```
<form action="validacion_condicionales.php" method="post" name="datos_usuario" id="datos_usuario">
  <table width="15%" align="center">
    <tr>
      <td>Nombre:</td>
      <td><label for="nombre_usuario"></label>
      <input type="text" name="nombre_usuario" id="nombre_usuario"></td>
    </tr>
    <tr>
      <td>Contraseña:</td>
      <td><label for="edad_usuario"></label>
      <input type="text" name="contra_usuario" id="contra_usuario"></td>
    </tr>
    <tr>
      <td>&nbsp;</td>
      <td>&nbsp;</td>
    </tr>
    <tr>
      <td colspan="2" align="center"><input type="submit" name="enviando" id="enviando">
    </td>
    </tr>
  </table>
</form>
```



2. evaluamos dos condiciones a la vez

```
if(isset($_POST["enviando"])){

    $edad=$_POST["edad_usuario"];

    $nombre=$_POST["nombre_usuario"];

    switch ($nombre){

        case "Juan":

            echo "Usuario autorizado. Hola Juan";

            break;

        case "María":

            echo "usuario autorizado. hola María";

            break;

        case "Pedro":

            echo "usuario autorizado. Hola Pedro";

            break;

        default:

            echo "Usuario no autorizado";

    }

}
```

### 3. otra forma

```
switch (true):  
    case $nombre=="Juan" && $contra=="1234":  
        echo "Usuario autorizado. Hola Juan";  
        break;  
    case $nombre=="María" && $contra=="5555":  
        echo "usuario autorizado. hola María";  
        break;  
    case $nombre=="Pedro" && $contra=="1111":  
        echo "usuario autorizado. Hola Pedro";  
        break;  
    default:  
        echo "Usuario no autorizado";  
  
endswitch
```

### 4. Hacemos el mismo ejemplo aquel en que se pedía la edad, y en funcion de la edad te decia lo que tenías que hacer, pero ahora con switch

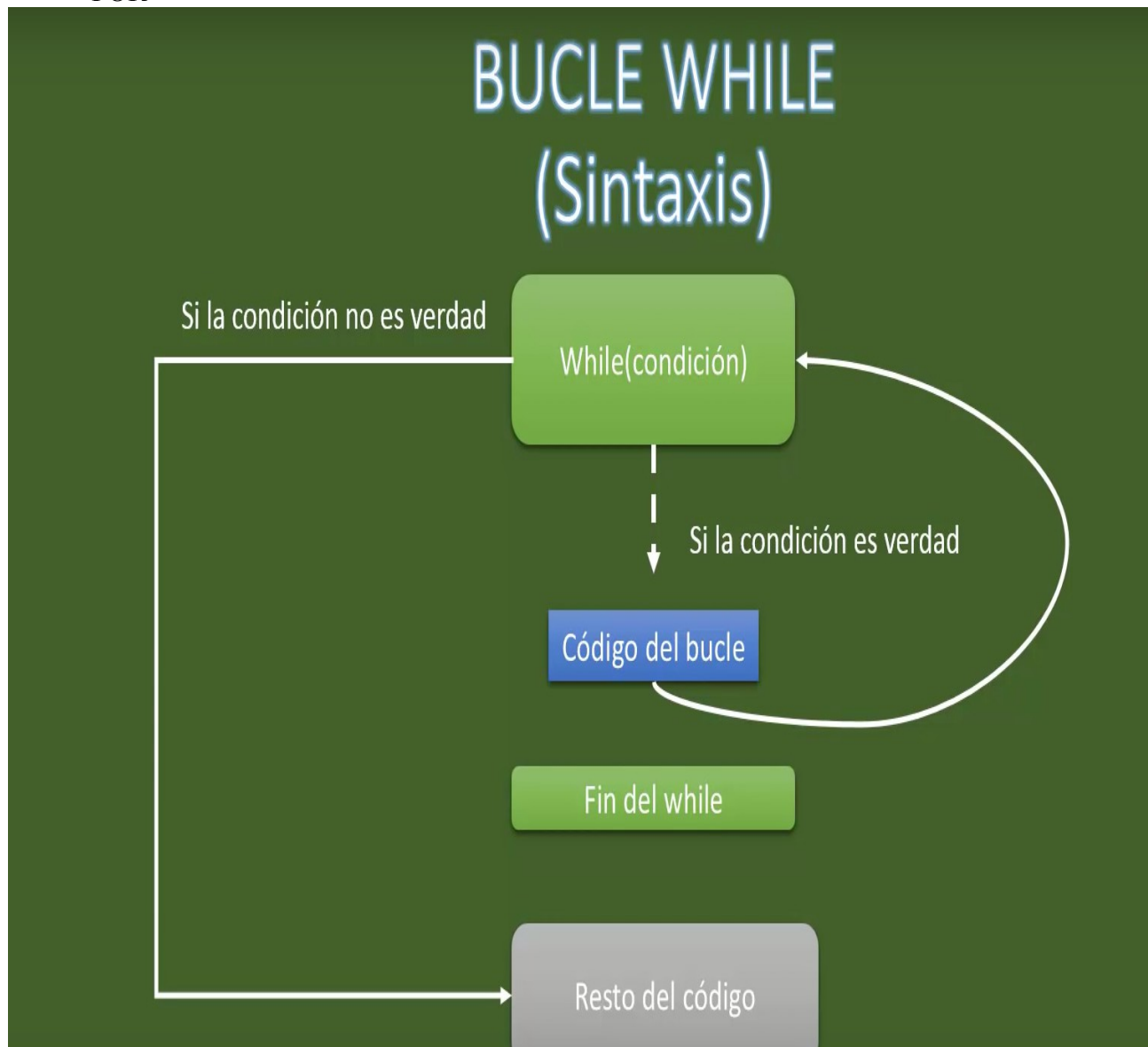
```
<?php  
if(isset($_POST["enviando"])){  
    $edad=$_POST["edad_usuario"];  
    $nombre=$_POST["nombre_usuario"];  
    switch (true):  
        case $edad<=18:  
            echo "Eres menor de edad";  
            break;  
        case $edad<=40:  
            echo "Eres joven";  
            break;  
        case $edad<=65:  
            echo "eres maduro";  
            break;  
        default:
```

## BUCLE WHILE/DOWHILLE (IGUAL A JS)

Estructuras repetitivas que pueden repetir una y otra vez lo que hay en su interior  
los bucles pueden ser :

- indeterminados: Aquellos que no sabemos antes de ejecutar el programa, cuantas veces repetirán el código que hay en su interior  
WHILE (mientras la condición sea verdad, ejecuta la instrucción)  
DO WHILE (ejecuta la instrucción, mientras sea verdad o hasta que sea mentira)
- Determinados, aquellos que sabemos antes incluso de ejecutar el programa cuantas veces repetirán el código que hay en su interior

FOR



Que utilidad tiene el bucle White en programacion:. Muchas  
y en concreto en PHP lo más relevante es acceder a una base de datos  
imaginemos que cada linea es un registro . Con el bucle recorres los registros con While

## EJERCICIO

Partimos de una variable \$var1=1

ejecuta el bloque en caso de que sea verdad. Ejecutará el bloque 5 veces hasta que termine el código. Una vez termine el código, entonces se imprime "hemos salido del bucle"

```
<?php

    $var1=1;

    while($var1<6){
        echo "Estamos ejecutando el código del bucle <br>";

        $var1++;

    }

    echo "Hemos salido del bucle";

?>
```

En caso de que la variable sea igual a 7, este será el flujo de ejecución: Pasa directamente a la línea siguiente sin pasar por el While porque ya es mentira la condición (es mayor a 6)

```
<?php

    $var1=7;

    while($var1<6){

        echo "Estamos ejecutando el código del bucle <br>";

        $var1++;

    }

    echo "Hemos salido del bucle";

?>
```

Bucle infinito: En el caso de estar dentro, y la condicion nunca pasa a ser mentira.

Ejemplo de bucle infinito:

En el siguiente ejemplo no incrementamos, porque la instruccion de incrementar la hemos comentado.

Como no podrá llegar nunca a ser mayor, siempre será menor, entonces SE HACE UN BUCLE INFINITO

```
<?php
    $var1=1;
    while($var1<6){
        echo "Estamos ejecutando el código del bucle <br>";

        // $var1++;

    }

    echo "Hemos sdalido del bucle";
```

## DIFERENCIAS ENTRE WHILE Y DO WHILE

En caso de que queramos que se ejecute la instruccion por lo menos una vez, convertimos el While en Do While.

DE ESTA MANERA PRIMERO EJECUTA Y LUEGO COMPRUEBA SI SE HA CUMPLIDO EL CÓDIGO

```
<?php
    $var1=7;
    //while($var1<6){
    do{
        echo "Estamos ejecutando el código del bucle <br>";

        $var1++;

    }while($var1<6);
    echo "Hemos sdalido del bucle";

?>
```

bucle for  
sintaxis:

- primero: el punto del que partimos para ejecutar el bucle
- condicion que se tiene que cumplir en el bucle dejar de ser verdad para empezar a ser false
- incremento o decremento

```
for (iniciación bucle; condición del bucle; incremento/decremento bucle)
```

ejercicio

abrimos un doc php llamado ejemplo for

¿como funciona?

Partimos de una variable  $i=0$ , luego la condicion para que se ejecute la instruccion

la tercera está diciendo: Después de que hayas ejecutado la instruccion incrementa el valor de la variable en uno.

Conclusion del ejemplo:

cuando  $i=11$  la condicon ya no será verdad; será mentira, por lo que saldrá del bucle for

¿como sería lo mismo pero decrementando?

```
<?php

for ($i=10; $i>=0; $i--) {

    echo "<p> Hemos entrado en el bucle </p>";

}
```

Otro ejemplo: Incrementar en  $i+=2$ . Abrimos un un if dentro de un for .para romper el bucle Breack

```
<?php

for ($i=0; $i<=20; $i+=2) {

    echo "<p> Hemos entrado en el bucle </p>";

    if ($i==6) {

        echo "<p>Bucle interrumpido</p>";

        break;

    }

}
```



## OTRO EJEMPLO

Instruccion que lo que hace no es salir del bucle sino impedir que una vuelta de bucle o una iteracion de bucle no se ejecute

vamos a explicarlo con un ejemplo: imaginemos que queremos imprimir la tabla de multiplicar del 9.

```
<?php
```

```
for($i=0;$i<=10;$i++){  
    echo "9 x $i = " . 9*$i . "<br>";  
}  
  
echo "Hemos salido del bucle";
```

```
?>
```

Si despues dividimos en lugar de multiplicar y pedimos que el bucle haga 20 iteracciones , pero cuando divide entre 0 da un error porque esta division es imposible

```
9 / 10 = 0.9  
9 / 9 = 1  
9 / 8 = 1.125  
9 / 7 = 1.2857142857143  
9 / 6 = 1.5  
9 / 5 = 1.8  
9 / 4 = 2.25  
9 / 3 = 3  
9 / 2 = 4.5  
9 / 1 = 9
```



Warning: Division by zero in C:\wamp\www\CursoPHP\ejemplo for.php on line 15

Entonces ponemos la siguiente instruccion CONTINUE

¿Qué hace continue? Indica que esa instruccion no la haga y pase a la siguiente.

Esta instruccion impide ejecutar una iteracion ycontinuar con la instruccion siguiente

```
for($i=10;$i>=-10;$i--){  
  
    if($i==0){  
  
        echo "División por 0 no es posible <br>";  
  
        continue;  
  
    }  
  
    echo "9 / $i = " . 9/$i . "<br>";  
  
}  
  
echo "Hemos salido del bucle";
```

```
9 / 10 = 0.9  
9 / 9 = 1  
9 / 8 = 1.125  
9 / 7 = 1.2857142857143  
9 / 6 = 1.5  
9 / 5 = 1.8  
9 / 4 = 2.25  
9 / 3 = 3  
9 / 2 = 4.5  
9 / 1 = 9  
División por 0 no es posible  
9 / -1 = -9  
9 / -2 = -4.5  
9 / -3 = -3  
9 / -4 = -2.25  
9 / -5 = -1.8  
9 / -6 = -1.5  
9 / -7 = -1.2857142857143  
9 / -8 = -1.125  
9 / -9 = -1  
9 / -10 = -0.9  
Hemos salido del bucle
```

