

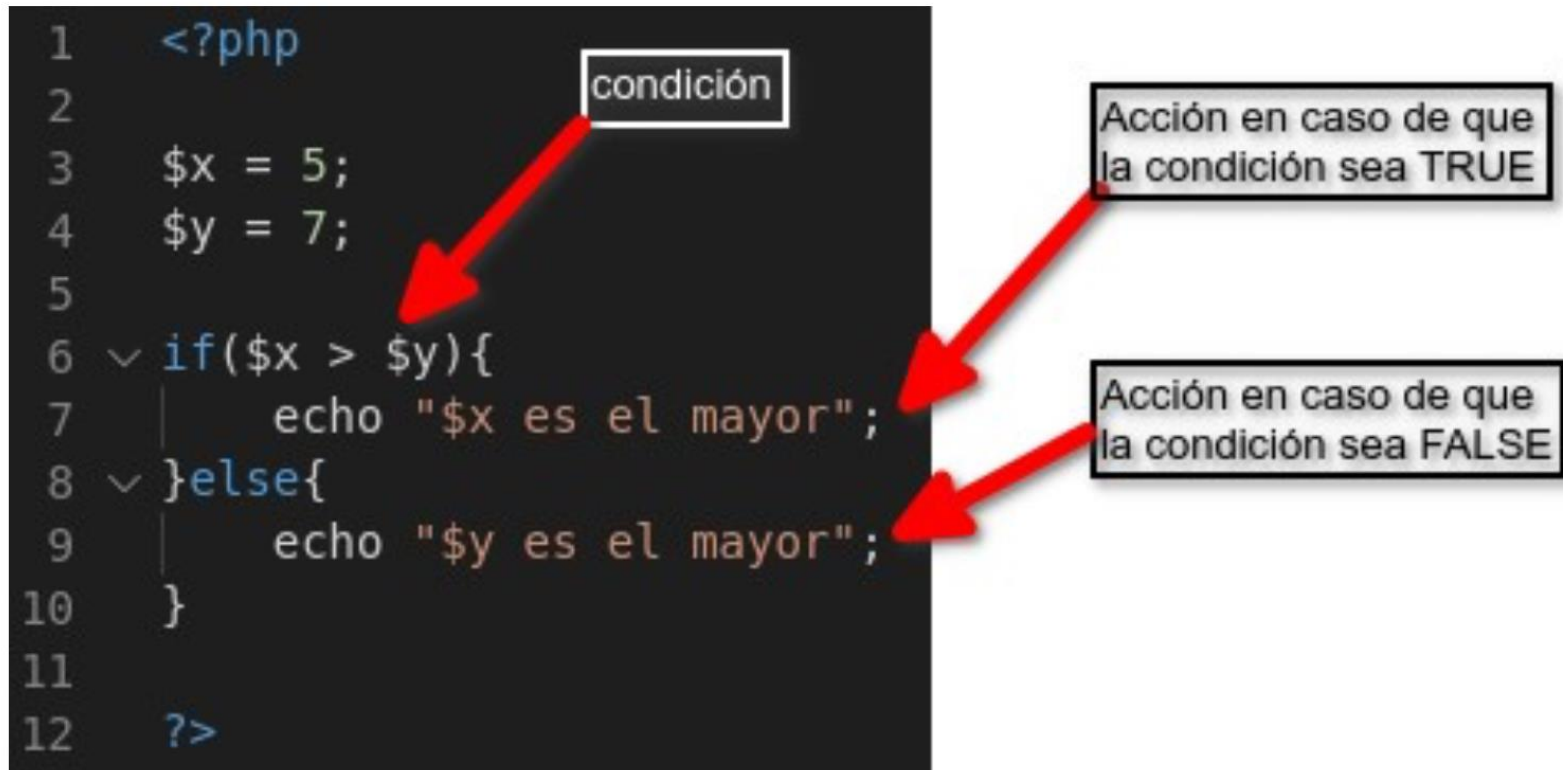


# Programación

---

ESTRUCTURAS DE CONTROL EN PHP

# Decisión: if



- La estructura ***ELSE*** es opcional

# Decisión: switch

```
1  <?php
2
3  $x = 5;
4  $y = 7;
5
6  switch($x){
7      case 5:
8          echo $x * $y;
9          break;
10     case 7:
11         echo $y - $x;
12         break;
13     default:
14         echo $y - $x;;
15         break;
16 }
17
18 ?>
```

Acciones diferentes para cada opción

Valores posibles para \$x

break; delimita las acciones

Acción por defecto

- **Switch** → 'decisión múltiple'
- **Si se omite 'break'**
  - se ejecuta también la siguiente acción
  - Serviría para agrupar valores bajo la misma acción

# Decisión: elseif

```
1  <?php
2
3  $x = 5;
4  $y = 7;
5
6  ✓ if ($x==5) {
7      |     echo $x * $y;
8  ✓ } elseif ($x==7) {
9      |     echo $y - $x;
10 ✓ } else {
11     |     echo $y - $x;
12     |
13
14  ?>
```

- **Se puede considerar el equivalente a *switch*, aunque:**
  - No podemos agrupar valores bajo la misma acción de forma directa
  - Pero si que podemos hacer *comparaciones complejas* (switch solo permite comparar igualando)
- **Podemos encadenar todas las sentencias *elseif* que necesitemos**
- **El último *else* hace la función 'default' de *switch*, pero es opcional**

# Operadores de comparación

---

```
1  <?php
2
3  $x = 5;
4
5  echo $x == 7; // igual a -> muestra FALSE
6  echo $x == "5"; // igual a -> muestra TRUE
7  echo $x === "5"; // estrictamente igual a -> muestra FALSE
8  echo $x != 7; // distinto de -> TRUE
9  echo $x != "5"; // distinto de -> FALSE
10 echo $x !== "5"; // estrictamente distinto a -> muestra TRUE
11
12 ?>
```

**‘Estrictamente’ igual o distinto  
significa ‘*en valor y en tipo*’ de dato**

# Operadores de comparación

```
1  <?php
2
3  $x = 5;
4
5  echo $x > 7; // mayor que -> muestra FALSE
6  echo $x > 5; // mayor que -> muestra FALSE
7  echo $x > 4; // mayor que -> muestra TRUE
8  echo $x < 7; // menor que -> muestra TRUE
9  echo $x < 5; // menor que -> muestra FALSE
10 echo $x < 4; // menor que -> muestra FALSE
11 echo $x >= 7; // mayor o igual que -> muestra FALSE
12 echo $x >= 5; // mayor o igual que -> muestra TRUE
13 echo $x >= 4; // mayor o igual que -> muestra TRUE
14 echo $x <= 7; // menor o igual que -> muestra TRUE
15 echo $x <= 5; // menor o igual que -> muestra TRUE
16 echo $x <= 4; // menor o igual que -> muestra FALSE
17
18  ?>
```

# Operadores lógicos

```
1  <?php
2
3  $x = 5;
4  $y = 7;
5
6  echo (($x > 7) and ($x > 4)); // muestra FALSE
7  echo (($x > 7) && ($x > 4)); // muestra FALSE
8
9  echo (($x > 7) or ($x > 4)); // muestra TRUE
10 echo (($x > 7) || ($x > 4)); // muestra TRUE
11
12 echo !(($x > 7) && ($x > 4)); // muestra TRUE
13 echo !(($x > 7) || ($x > 4)); // muestra FALSE
14
15 ?>
```

FALSE TRUE

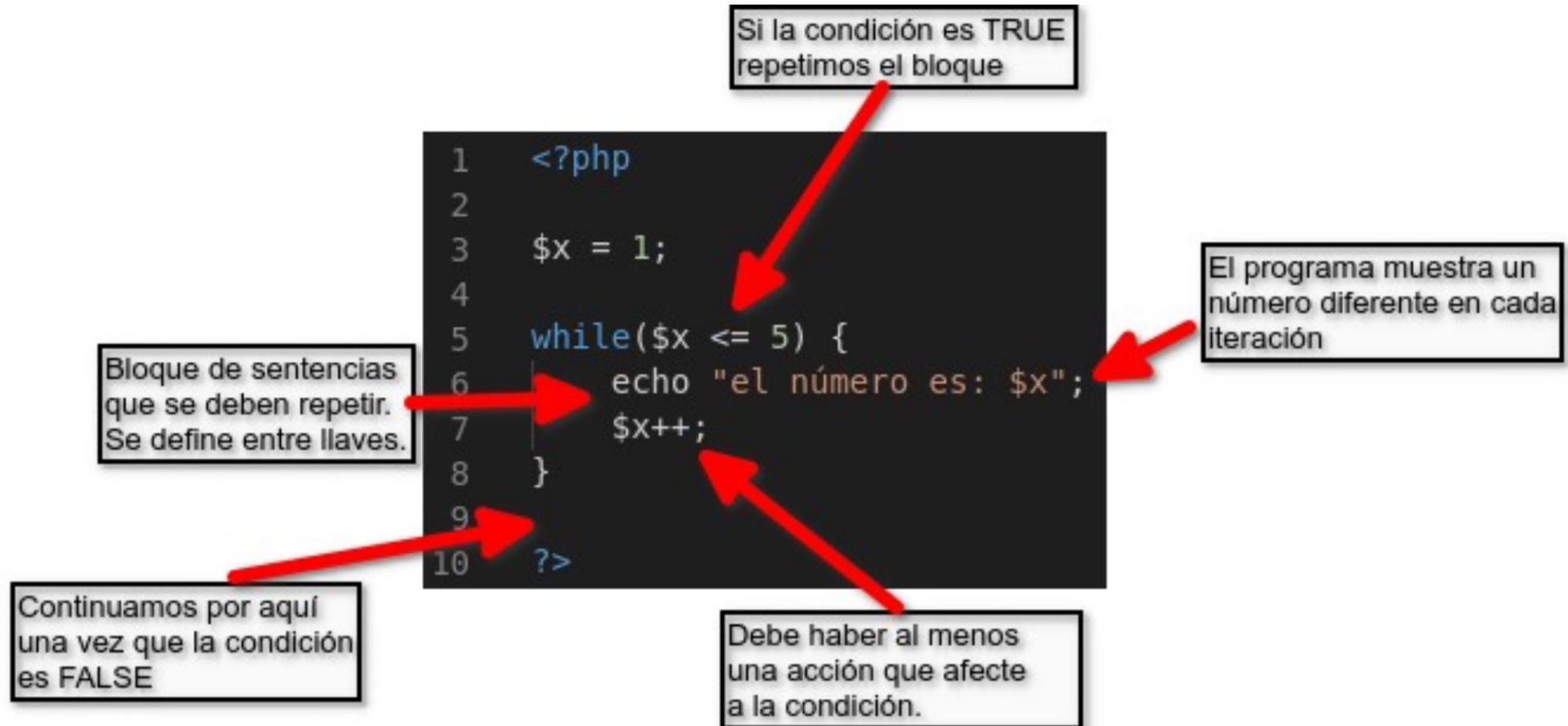
'and' equivale a '&&'

'or' equivale a '||'

el signo de exclamación significa 'negación' y su efecto es invertir el resultado

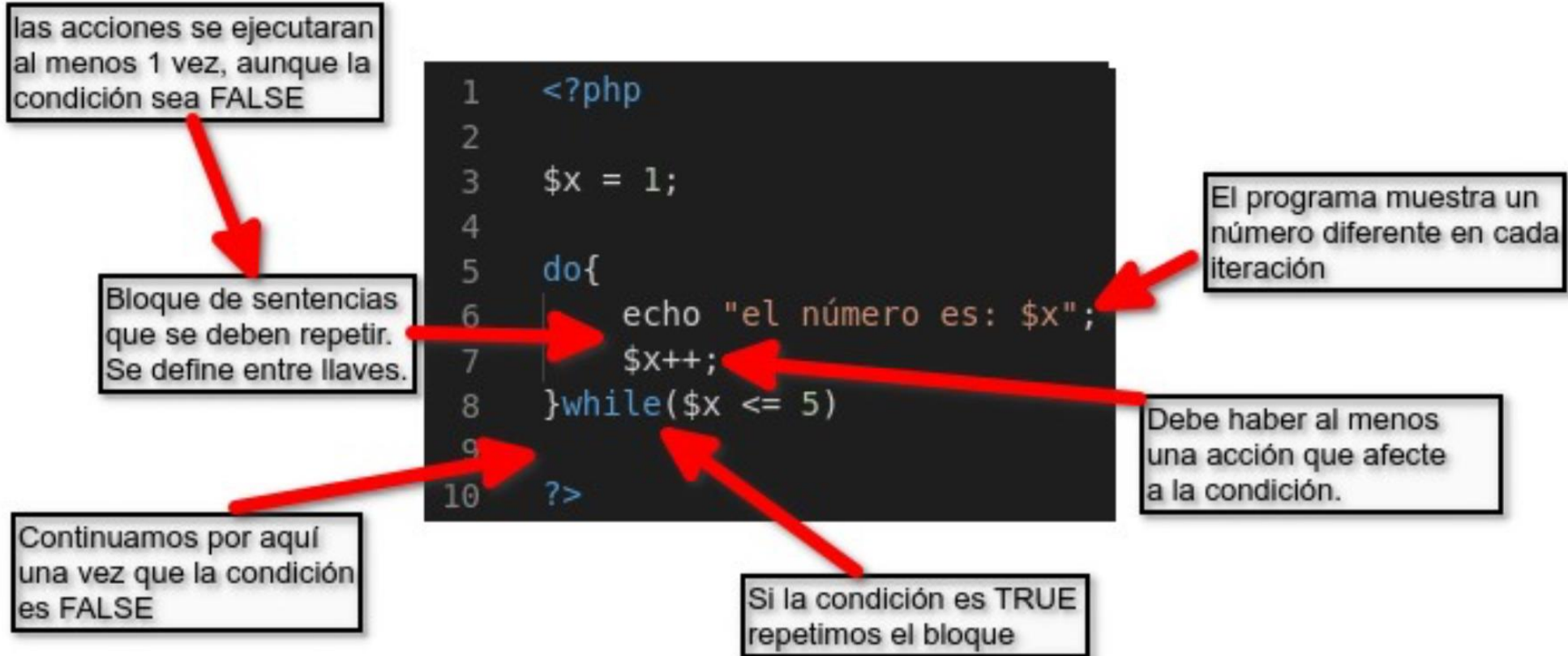


# Bucles: while





# Bucles: do..while



# Bucles: for

The diagram illustrates the components of a PHP `for` loop. A central code block is annotated with five text boxes and red arrows:

- Top-left box:** "Inicializa el contador de bucle (\$x) y pone el valor inicial a 0". An arrow points to the `$x = 0` part of the loop header.
- Top-right box:** "Continúa el bucle siempre y cuando \$x sea menor o igual a 10". An arrow points to the `$x <= 10` part of the loop header.
- Right box:** "Aumentar el valor del contador de bucle en 1 por cada iteración". An arrow points to the `$x++` part of the loop header.
- Bottom-right box:** "El programa mostrará un valor diferente en cada iteración". An arrow points to the `echo` statement inside the loop body.
- Bottom-left box:** "Continuamos por aquí cuando \$x > 10". An arrow points to the closing tag `?>` at the end of the code block.

```
1  <?php
2
3  for ($x = 0; $x <= 10; $x++) {
4      echo "El número es: $x <br>";
5  }
6
7  ?>
```

