UD2.- El Lenguaje PHP (Estructuras de Control)

Estructuras de **Selección** y estructuras de **Repetición**.

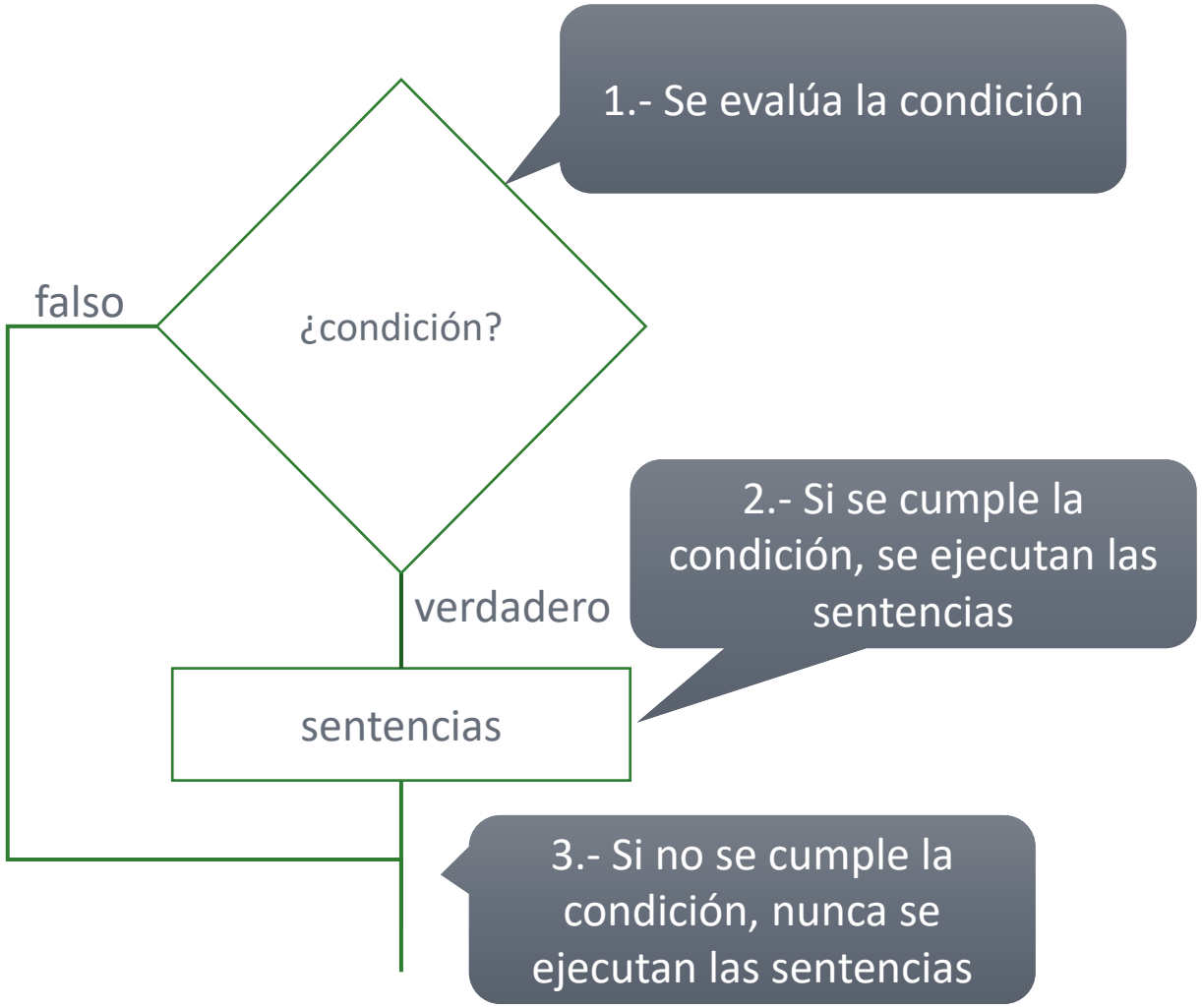
Estructuras de Selección

- if
- if-else
- switch

Estructura

if

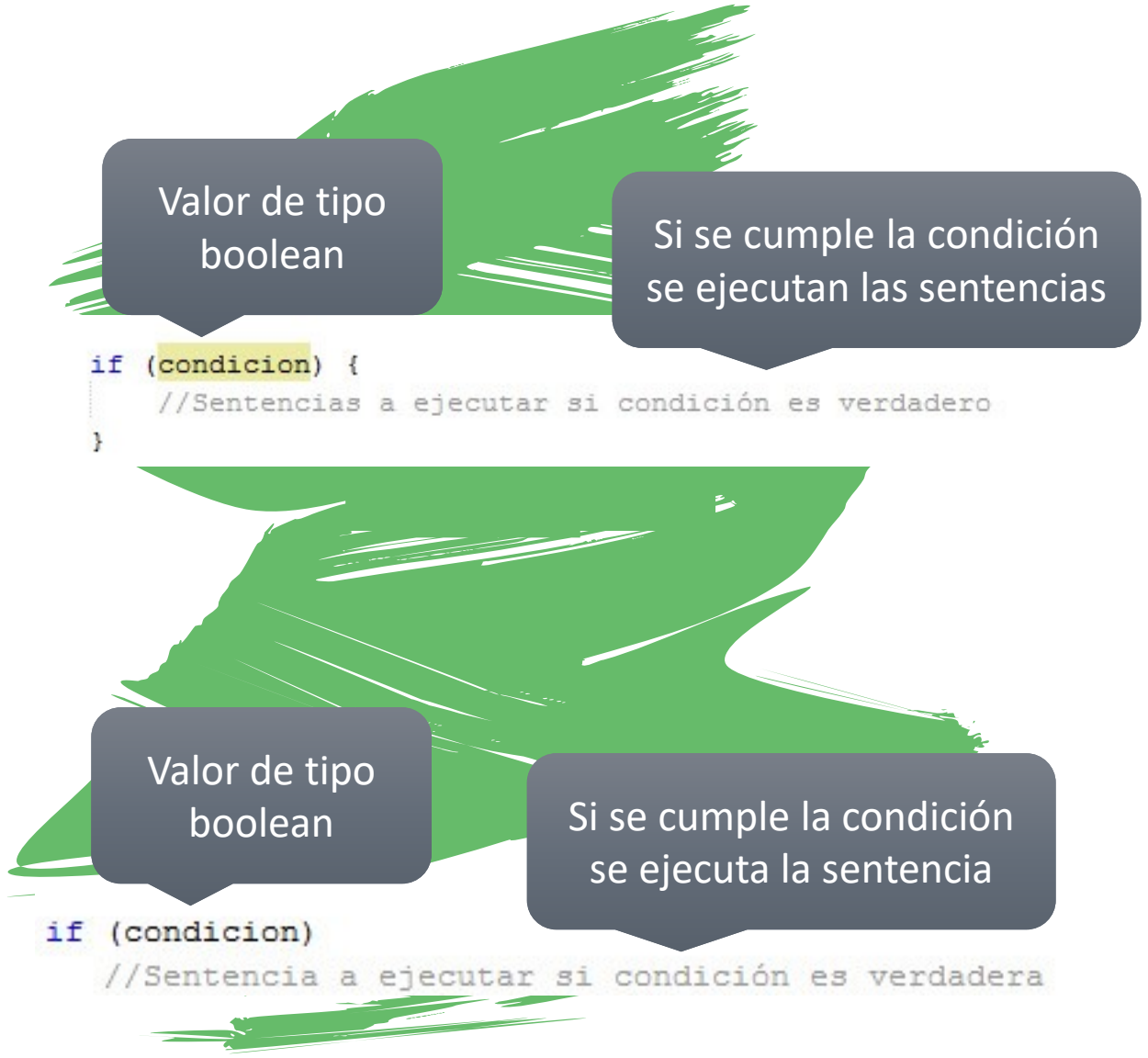
Para iniciar se evalúa una condición. Si esta es verdadera entonces se ejecutan las sentencias contenidas en ese bloque de código. Un bloque de código en la estructura *if* puede llevar corchetes o no ({ … }), esto con el fin de delimitar las sentencias o instrucciones que debe ejecutar el *if.*



Estructura

if

Para iniciar se evalúa una condición. Si esta es verdadera entonces se ejecutan las instrucciones contenidas en ese bloque de código. Un bloque de código en la estructura *if* puede llevar corchetes o no ({ … }), esto con el fin de delimitar las sentencias o instrucciones que debe ejecutar el *if*. En el caso de que no se usen corchetes, entonces sólo se podrá agregar una instrucción después de la palabra *if*, como sigue a continuación:



Ejemplos de uso de sentencia if.

Ejemplo if

Como el valor de la variable edad es mayor que 17, se ejecutarán la/s sentencia/s que hay entre las llaves del *if*, y se mostrará por pantalla el mensaje:
"Eres mayor de edad"
Si **sólo** hay que ejecutar **una** sentencia dentro del *if* podemos poner el código sin los corchetes.
En este caso ambos ejemplos son equivalentes.

```
$edad = 19;

if($edad > 17){
    printf("Eres mayor de edad");
}
```

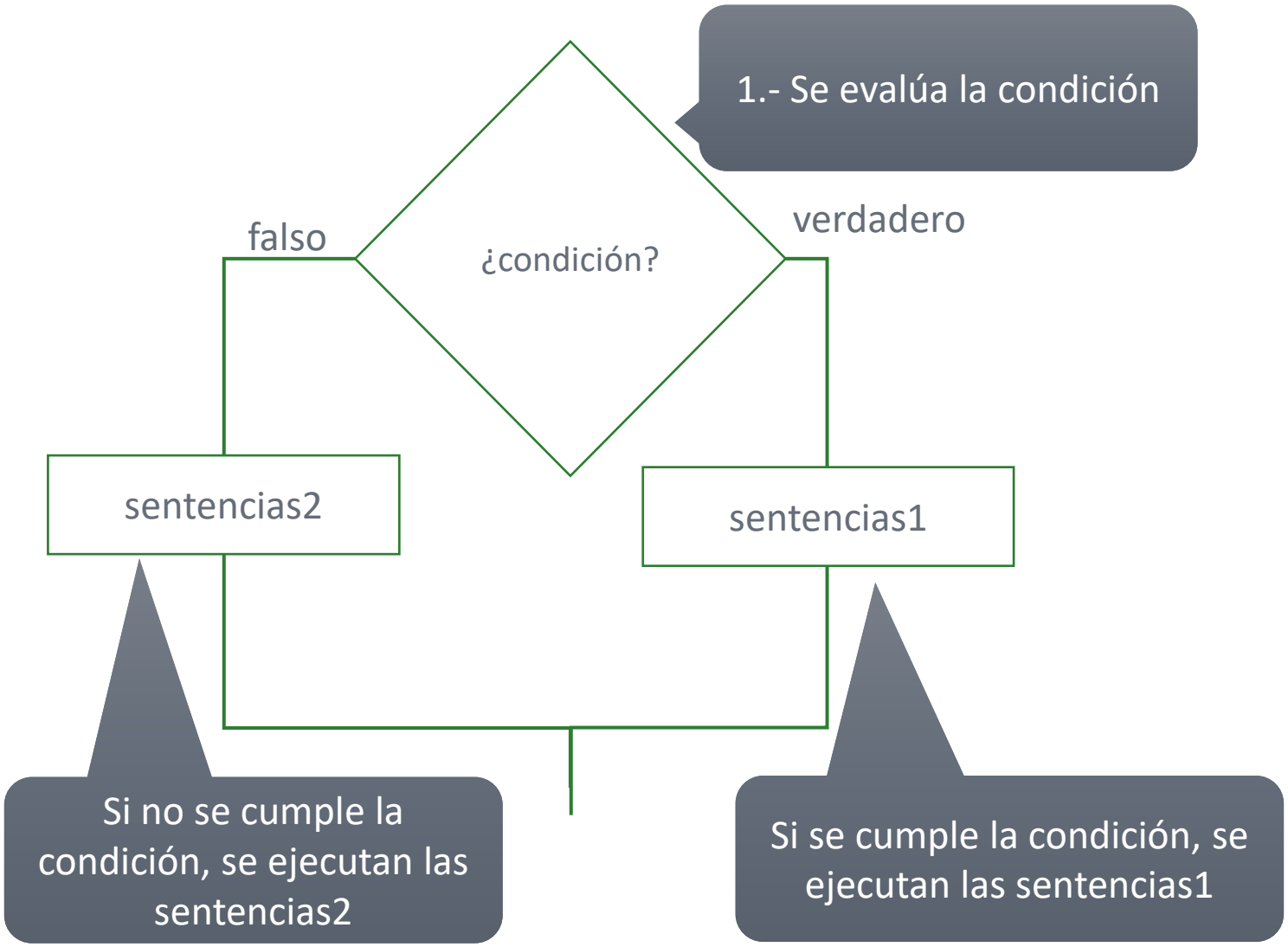
Como se cumple la condición del if (el valor de la variable edad es mayor que 17), el programa mostrará por pantalla el mensaje "Eres mayor de edad"

Si sólo hay que ejecutar una sentencia se puede escribir sin corchetes

```
if($edad > 17)
    printf("Eres mayor de edad");
```

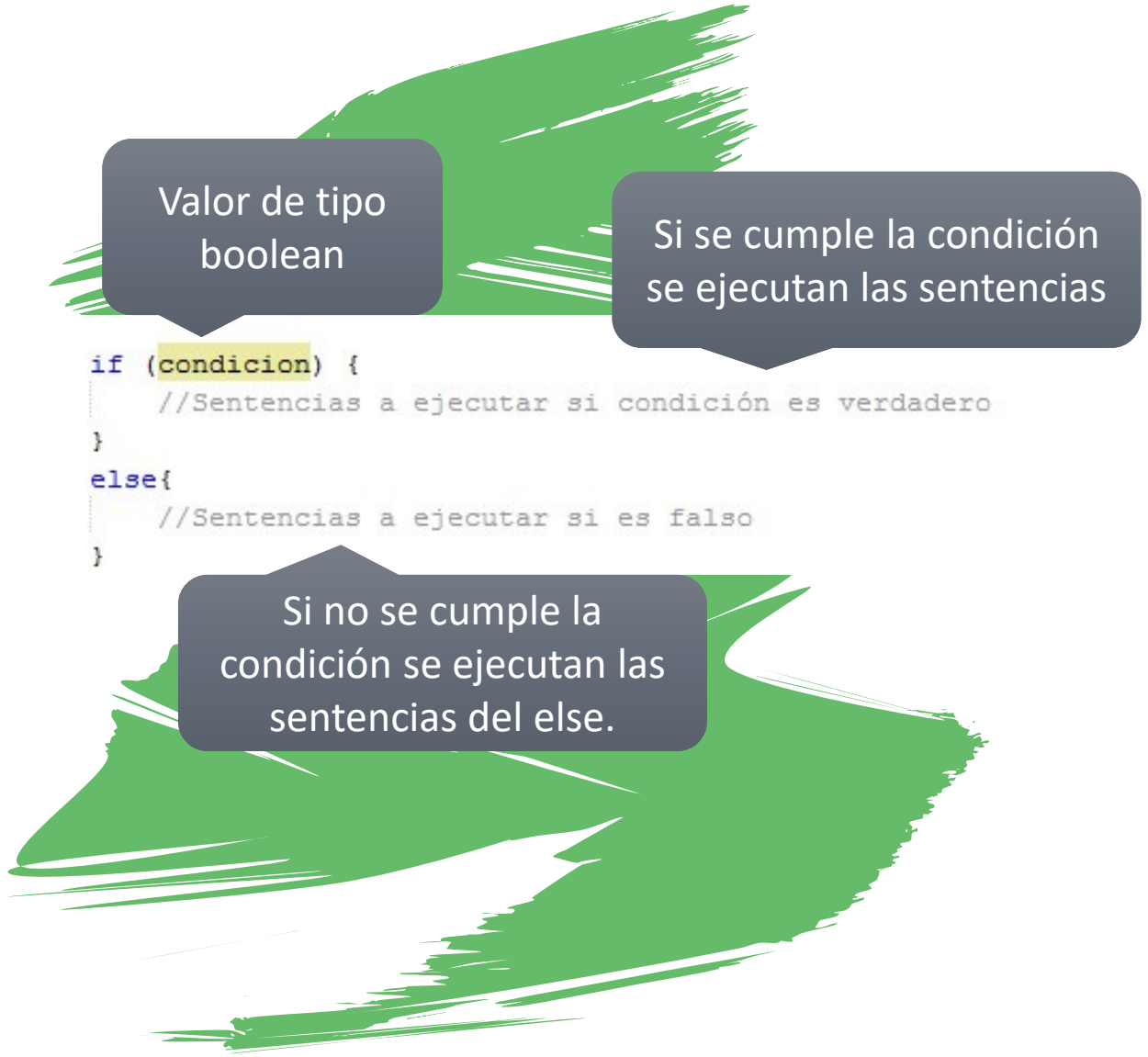
Estructura
If-else

Para iniciar se evalúa una condición. Si esta es verdadera entonces se ejecutan las sentencias1 contenidas en ese bloque de código. Sino se cumple la condición se ejecutan las sentencias2. Un bloque de código en la estructura *if-else* puede llevar corchetes o no ({ … }), esto con el fin de delimitar las sentencias o instrucciones que debe ejecutar el *if-else.*



Estructura if-else

La estructura *else* es opcional. Si no se cumple la condición del *if*, se ejecutan las sentencias del *else*.



Ejemplos de uso de sentencia if-else.

Ejemplo if-else

Como el valor de la variable edad es mayor que 17, se ejecutarán la/s sentencia/s que hay entre las llaves del *if*, y se mostrará por pantalla el mensaje:
“Eres mayor de edad” .
Si el valor de la variable edad fuera menor o igual a 17, se ejecutarían la/s sentencia/s del *else*.

Como se cumple la condición del if (el valor de la variable edad es mayor que 17), el programa mostrará por pantalla el mensaje “Eres mayor de edad”

```
$edad = 19;

if($edad > 17){
    printf("Eres mayor de edad");
}
else{
    printf("Todavía no eres mayor de edad");
}
```

Si la variable edad no fuera mayor de 17, se ejecutaría el else y el programa mostraría por pantalla el mensaje: “Todavía no eres mayor de edad”

Si sólo hay que ejecutar una sentencia (en el if y/o en el else) se puede escribir sin corchetes

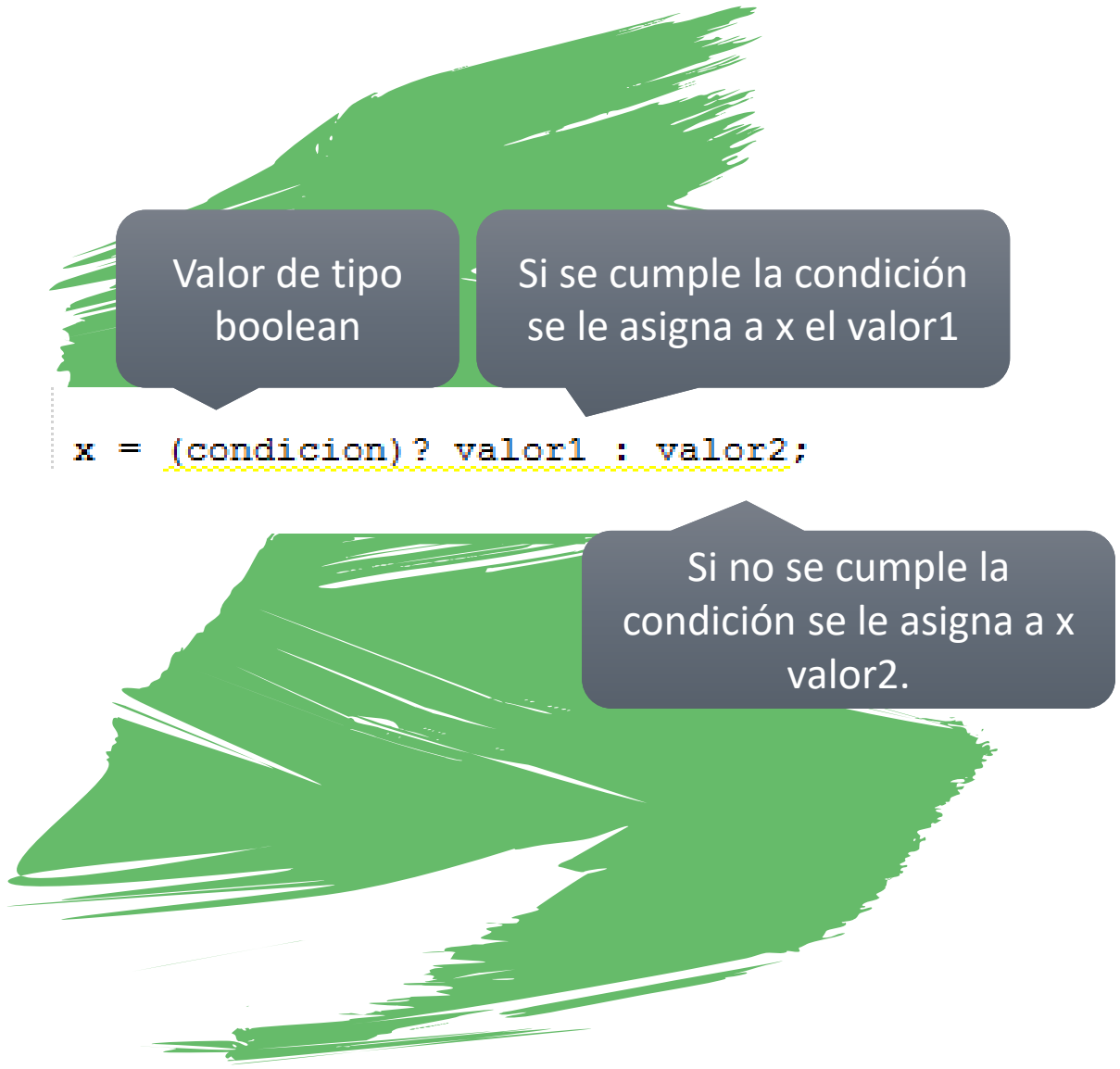
```
$edad = 19;

if($edad > 17)
    printf("Eres mayor de edad");
else
    printf("Todavía no eres mayor de edad");
```

Operador Condicional

?:

El operador condicional (*?:*) puede utilizarse en muchas ocasiones en sustitución de un *if-else*. Su misión es devolver un valor en función de una condición boolena.



Ejemplos de uso de sentencia ?:

Ejemplo ?:

Como el valor de la variable edad es mayor que 17, se ejecutará la sentencia que hay tras el símbolo *?*, y se asignará a la variable mensaje la cadena : “Eres mayor de edad” .
Si el valor de la variable edad fuera menor o igual a 17, se ejecutaría la sentencia situada detrás del símbolo :, y se asignaría a la variable mensaje la cadena: “Todavía no eres mayor de edad”.

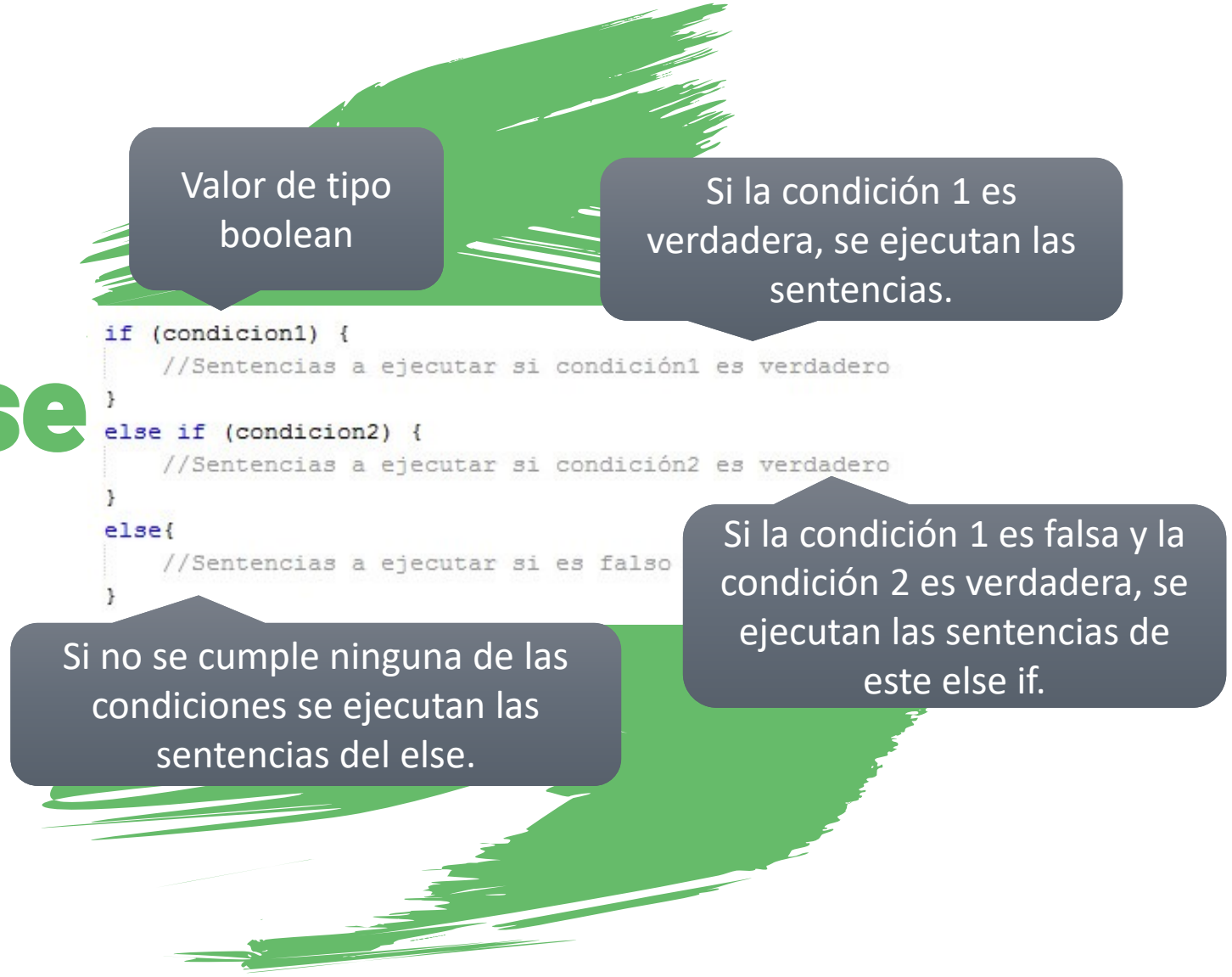
Como se cumple la condición (el valor de la variable edad es mayor que 17), se le asigna a la variable mensaje la cadena “Eres mayor de edad”.

```
$edad = 19;
$mensaje = ($edad > 17)? "Eres mayor de Edad":"Todavía no eres mayor de edad";
echo $mensaje;
```

Si la variable edad no fuera mayor de 17, se le asignaría a la variable mensaje la cadena: “Todavía no eres mayor de edad”

Estructura if-else/if-else

La estructura *else* es opcional. Si no se cumple la condición del *if*, se ejecutan las sentencias del *else*.



Ejemplos de uso de sentencia if-else/if-else.

Ejemplo if-else/if-else

Aquí, se comienza evaluando el primer *if*. Si se cumple su condición se ejecutan sus sentencias y listo. Si no se cumple su condición (x == 10), se procede a evaluar la siguiente condición else if (x == 20). Si se cumple esta condición se ejecutan sus sentencias, en caso contrario se procede a evaluar la siguiente condición y así sucesivamente.

Si no se cumple ninguna condición, se ejecutarían la/s sentencia/s del *else*.

```php
$x = 10;

if($x == 10){
    printf("X es igual a 10");
}
else if($x == 20){
    printf("X es igual a 20");
}
else if($x == 30){
    printf("X es igual a 30");
}
else{
    printf("X no es igual ni a 10, ni a 20, ni a 30.");
}
```

Ejemplo de utilización de las sentencias:
if - else/if - else

Estructura switch

La estructura *switch* se utiliza cuando una expresión puede tener varios valores y dependiendo del valor que tome hay que ejecutar una serie de sentencias.

Variable a evaluar

```
switch (variable){
    case 1:
        //Sentencias1
        break;

    case 2:
        //Sentencias2
        break;

    case 3:
        //Sentencias3
        break;

    default:
        //Sentencias4
```

Si la variable toma el valor 1, se ejecutan las sentencias1.

La sentencia break; evita que se sigan ejecutando las sentencias de los siguientes case.

Si la variable toma el valor 2, se ejecutan las sentencias2.

Si la variable toma el valor 3, se ejecutan las sentencias3.

Si no se cumple ninguna de las condiciones se ejecutan las sentencias4 del default.

Ejemplos de uso de sentencia switch.

Ejemplo switch

En este caso *switch* compara el valor de la variable x. En el caso de que sea 10, se muestra por pantalla el mensaje “X es igual a 10.”. En el caso de que sea 20, se muestra por pantalla el mensaje “X es igual a 20.”. En el caso de que sea 30, se muestra por pantalla el mensaje “X es igual a 30.”. **Si no es igual a ninguno de esos valores, por defecto se muestra:** “X no es igual ni a 10, ni a 20, ni a 30.”.

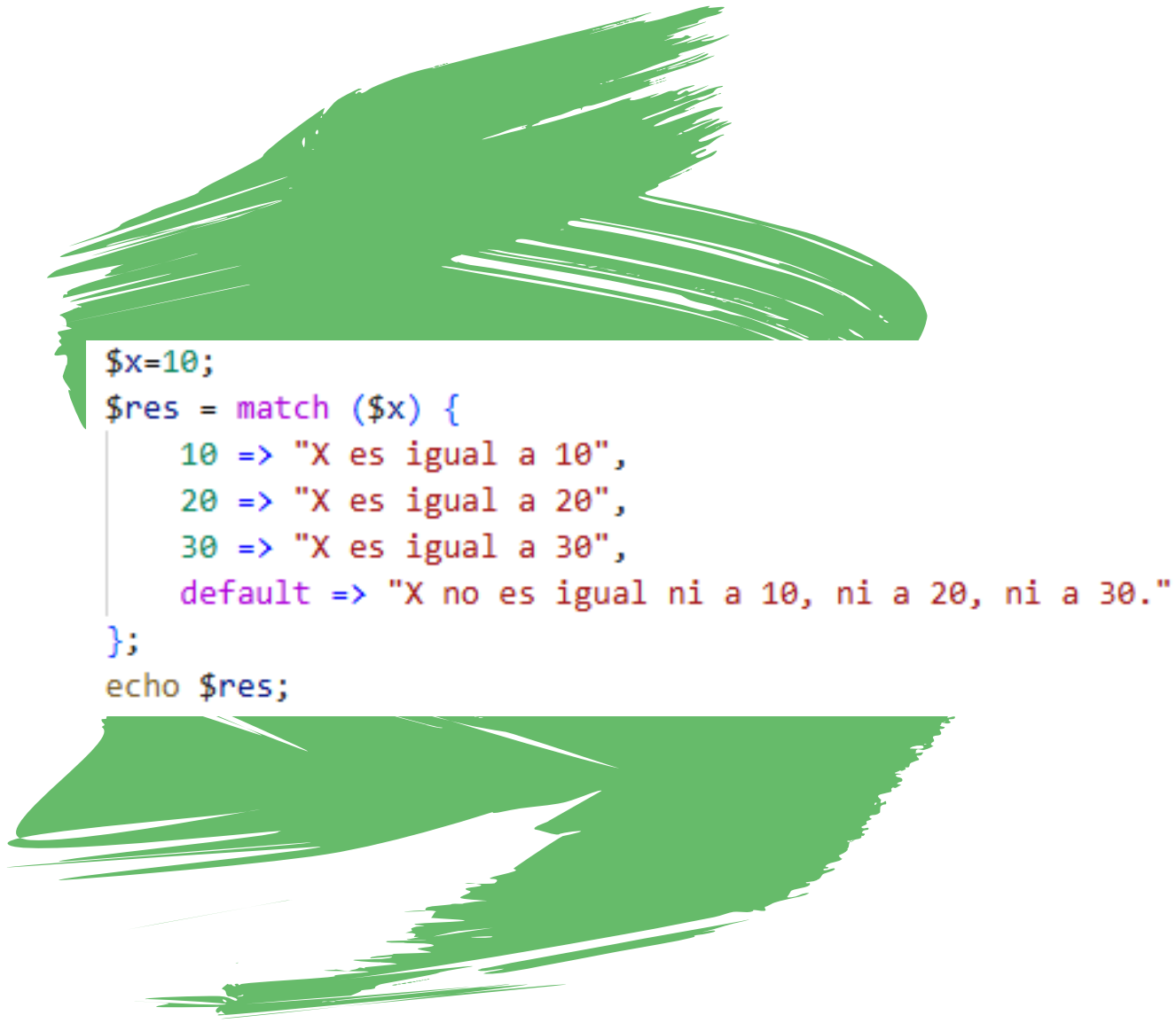
Ejemplo de utilización de la sentencia **switch**

```php
$x=10;
switch($x){
    case 10:
        echo "X es igual a 10.";
        break;
    case 20:
        echo "X es igual a 20.";
        break;
    case 30:
        echo "X es igual a 30.";
        break;
    default:
        echo "X no es igual ni a 10, ni a 20, ni a 30.";
}
```

Este ejemplo es equivalente al anterior programado con sentencias if – else/if - else

Expresión match

La expresión *match* disponible desde la versión 8, es una alternativa **más moderna y segura** a la sentencia ***switch*** para realizar comparaciones estrictas y devolver valores. Su uso básico consiste en evaluar una expresión y comparar su valor con una serie de casos (case), ejecutando el bloque de código correspondiente al primer caso coincidente y retornando su valor. También, permite especificar un caso default para manejar valores que no coinciden con ningún caso anterior.



```php
$x=10;
$res = match ($x) {
    10 => "X es igual a 10",
    20 => "X es igual a 20",
    30 => "X es igual a 30",
    default => "X no es igual ni a 10, ni a 20, ni a 30."
};
echo $res;
```

Ejemplos de uso de sentencia match.

Ejemplo match

En este caso *match* compara el valor de la variable estado. En el caso de que sea ‘pendiente’, se muestra por pantalla el mensaje “La tarea está en progreso”. En el caso de que sea ‘completado, se muestra por pantalla el mensaje “La tarea se completó”. **Si no es igual a ninguno de esos valores, por defecto se muestra:** “Estado desconocido”.

Ejemplo de utilización de la sentencia match

```php
$estado = 'pendiente';

$mensaje = match ($estado) {
    'pendiente' => "La tarea está en progreso.",
    'completado' => "La tarea se completó.",
    default => "Estado desconocido.",
};

echo $mensaje; // Salida: La tarea está en progreso
```

Este ejemplo es equivalente al anterior programado con sentencias if – else/if - else

Estructuras de Repetición

- while
- do-while
- for

Estructura while

El buble *while* en PHP, es el más fundamental para realizar iteraciones. Básicamente como podemos observar en la figura, se ejecuta un bloque de código tantas veces como la condición que se evalúa sea verdadera. Una vez que esta condición es falsa, entonces se termina la iteración.

Debido a que la condición del bucle *while* se evalúa al inicio, el bloque de código a repetir no se ejecutará **ni una sola vez** si esta condición no se cumple. Para ello existe el bucle *do-while* que veremos después.

Ejemplos de uso de bucle while.

Ejemplo while

Aquí se comienza inicializando la variable *num* con el valor 1. El bucle *while* comprueba inicialmente si *num* es menor o igual que 10. Si se cumple esta condición se ejecutan las sentencias que hay dentro del bucle (se muestra por pantalla el valor de *num*, y se incrementa la variable *num* en una unidad). Las sentencias del bucle se volverán a ejecutar mientras la variable *num* sea menor o igual que 10.

Ejemplo de utilización del bucle while

```php
$num = 1;
while($num <=10){
    print $num;
    $num++;
}
```

Este bucle muestra los números del 1 al 10.

Estructura do-while

En algunas ocasiones es deseable que el código a repetir se pueda ejecutar por lo menos una vez antes de terminar el bucle. Para ello se ha creado el bucle *do-while*.
Debido a esto este ciclo evalúa el código a repetir por lo menos una vez. Y en cada iteración, primero se ejecuta el código a repetir, y después se evalúa la condición para saber si el bucle continúa o no.

Ejemplos de uso de bucle do-while.

Ejemplo do-while

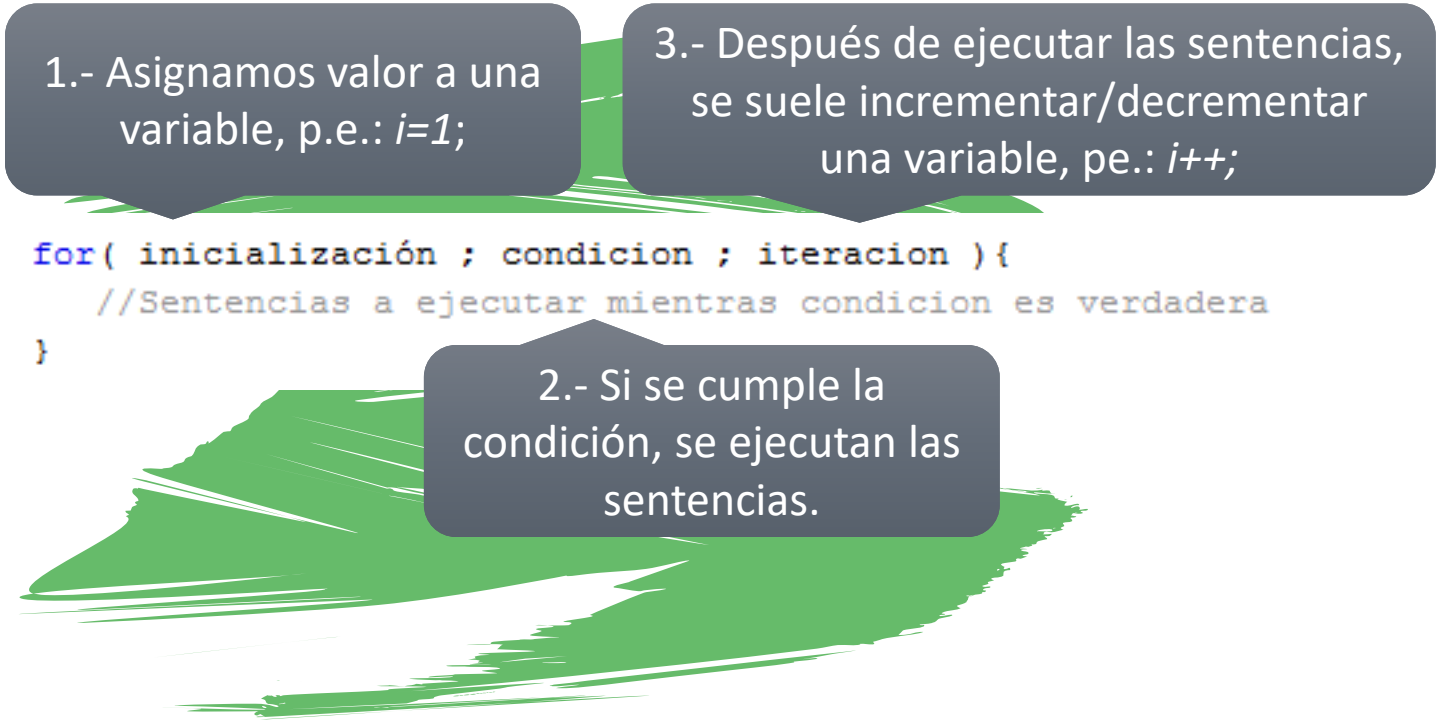
Aquí se comienza inicializando la variable *num* con el valor 1. Las sentencias del bucle se ejecutan por lo menos una vez, aunque no se cumpla la condición. Se ejecutan las sentencias que hay dentro del bucle (se muestra por pantalla el valor de *num*, y se incrementa la variable *num* en una unidad). A continuación, se comprueba si num es menor o igual que 10, en caso afirmativo se vuelven a ejecutar las sentencias del bucle.

Ejemplo de utilización del bucle do-while

```
$num = 1;
do{
    print $num;
    $num++;
}while($num <=10);
```

Este bucle muestra los números del 1 al 10.

Estructura for



El bucle *for* se forma de 3 elementos (inicialización, evaluación de condición e incremento/decremento).
Aquí es importante notar el orden en que se ejecutan los pasos:

1) Se inicializan las variables de control (también conocidas como contadores).
2) Se revisa si la condición es verdadera, si es así se ejecutan las sentencias del bucle. Si es falsa la condición, termina el bucle.
3) Si la condición fue verdadera, se incrementa o decrementa la variable de control. Este elemento, llamado iteración, controla la forma en que el bucle progresa, y normalmente aprovechamos este elemento para incrementar o decrementar nuestro contador.
4) Se vuelve a revisar la condición, si es verdadera se repiten los pasos del 2 al 4. Si la condición es falsa, concluye el bucle.

Ejemplos de uso de bucle for.

Ejemplo for

1.- Se comienza inicializando la variable *num* con el valor 1.
2.- A continuación se comprueba si *num*<=10.
3.- Si se cumple la condición se ejecuta la sentencia del bucle (se muestra por pantalla el valor de *num*).
4.- Se incrementa la variable *num* en una unidad.
5.- Se vuelve a revisar la condición. Si es verdadera se repiten los pasos del 2 al 4, sino concluye el bucle.

Ejemplo de utilización del bucle for

```
for($num=1; $num<=10; $num++){
    print $num;
}
```

Este bucle muestra los números del 1 al 10.

Sentencia break

La sentencia *break* que vimos con la estructura de control *switch*, se puede utilizar también en los bucles *while*, *do-while* y *for*, para forzar la salida inmediata de los mismos.
Su uso más habitual es para escapar anticipadamente de un bucle. Veamos un ejemplo:

Cuando num valga 5, el programa saldrá automáticamente del bucle.

```php
for($num=1; $num<=10; $num++){
    if($num == 5)   #Si num es 5
        break;      #Salimos del bucle automáticamente
    print $num;
}
```

Muestra por pantalla:
1 2 3 4

Sentencia continue

La sentencia *continue* se suele utilizar en las estructuras de repetición *while*, *do-while* y *for*. Se utiliza habitualmente para omitir las instrucciones restantes dentro de estos bucles, y continuar con las iteraciones restantes. Veamos un ejemplo:

Cuando ***num*** valga 5, el programa vuelve automáticamente a la iteración, e incrementa ***num***.

```php
for($num=1; $num<=10; $num++){
    if($num == 5)   #Si num es 5
        continue;   #Omito el código que viene a continuación, y paso a la siguiente iteración
    print $num;
}
```

Muestra por pantalla:
1 2 3 4 6 7 8 9 10