

# Operadores Lógicos

Los operadores lógicos comparan valores booleanos y devuelven respuestas booleanas.

- Operador “Y” (“AND”) && Si ambas expresiones son verdaderas devolverá: verdadero (true)

Ej:

```
If (5 == 5 && 5>2 )
```

en la primera expresión decimos que si el elemento 5 es igual 5 (==), y en la segunda que 5 es mayor que 2.

Como ambas son correctas, devolverá : **true** (ejecuta código)

```
if (5 ==5 || 5>7)
```

en la primera expresión decimos que si el elemento 5 es igual 5 (==), y en la segunda que **5 es mayor que 7**

Solo una expresión es correcta, devolverá : **false** (**NO** ejecuta código).

- Operador “Ó” (“OR”) || Se escribe con dos barras verticales (ASCII Alt+124) a diferencia de AND (&&) este operador, Si determina que **una o las dos**, de las expresiones es verdadera (true), devolverá true y ejecuta el código.

Ej:

```
if (5 ===5 || 5>2)
```

en la primera expresión decimos que si el elemento y el tipo 5 es igual 5 (===), y en la segunda que 5 es mayor que 2.

Como ambas son correctas, devolverá : **true** (ejecuta código)

```
if (5 ===5 || 5>7)
```

en la primera expresión decimos que si el elemento y el tipo 5 es igual 5 (===), y en la segunda que **5 es mayor que 7**.

Solo una expresión es correcta, de todas formas devolverá : **true** (ejecuta código)

```
if (5 ===7 || 5>7)
```

en la primera expresión decimos que si el elemento y el tipo **5 es igual 7** (===), y en la segunda que **5 es mayor que 7**.

Al ser ambas declaraciones incorrectas devolverá : **true** ( **NO** ejecuta código)

- Operador “NOT” (“NO”) ! Se escribe con un signo de cierre de exclamación. Devolverá el valor contrario a la expresión : si es verdadero (true) nos devolverá: falso (true).

Ej:

```
If (!(5 === 5 ) )
```

En la expresión decimos que si el elemento y el tipo 5 es igual 5 (===).

Es totalmente verdadero (**true**), por lo cual devolverá lo contrario: falso (**false**)  
( **NO** ejecuta código)

- Bucles **For** (desde) es una estructura repetitiva ejecutando un código una cierta cantidad de veces. (es aquella en la que el número de iteraciones se conoce por anticipado, y por ello no se precisa poner ninguna condición de salida para detener el bucle. En su lugar un contador cuenta el número de iteraciones fijas y se termina cuando llega al valor final previamente definido). Se emplea en el recorrido de vectores, matrices y estructuras, entre otros y se utiliza cuando :
  - Quiere que su código se ejecute indefinidamente.
  - Quiere ejecutar código hasta que se cumpla una condición.
  - Quiere que el usuario controle cuándo debe detenerse el código.

Ej:

```
for (var i = 0; i <= 100: i++)
```

Tiene 3 partes la primera; es la Variable, la segunda: es la condición de frenado, (Expresión condicional) y la tercera: como quieres que se implemente el valor (Incremento de la variable ).

De este modo la variable i que es 0 aumentara el contador de uno hasta llegar a 100, Nuestro frenado.