

# Operando Lógicamente

Llegó el momento de poner en práctica toda esta información acerca de los operadores y cómo podemos utilizarlos para crear secuencias lógicas y de comparación, para ello vamos a realizar una serie de ejercicios que nos ayuden a eso (y probablemente nos hagan pensar un poco).

## ¿Qué devuelve cada Expresión? Pt1

Observen estos ejemplos y piensen qué devuelve cada uno:

1. !true // false
2. !false // true
3. !!false // false
4. !!true // true
5. !1 // false
6. !!1 // true
7. !0 // true
8. !!0 // false
9. !!"" // false
10. let x = 5 ;

let y = 9;

* 1. x < 10 && x!==5 // false
  2. x>9 || x===5 // true
  3. !(x===y) // true

**res: cada vez que negamos algún valor, lo convertimos siempre en booleano, es decir, si tenemos el número 1, al negarlo lo convertimos en false, ya que internamente el número 1 era true, porque “tiene contenido”, a diferencia del 0 que está “vacío”, lo mismo sucede con los strings, un string vacío (“”) no es lo mismo que un string que contiene un espacio (“ “).**

## ¿Qué devuelve cada Expresión? Pt2

Sin probar en la consola, piensen que devolverá cada una de estas expresiones, ¿son *true* o *false*?

1. let x=10  
   let y ="a"  
   y==="b" || x >= 10 // true
2. let x=3  
   let y=8  
   !(x == "3" || x === y) && !(y !== 8 && x <= y) // false
3. let str = ""  
   let msj = "jaja!"  
   let esGracioso = "false"  
   !((str || msj) && esGracioso) // false

Code

Crear el código JS que exprese los siguientes enunciados:

1. Para subir a una montaña rusa la edad debe ser mayor a 12 años y la altura debe ser mayor a 1,30 m.

*//creamos nuestras dos variables de edad y de altura, el cual asignamos dos valores que querramos*

let edad = 13;

let altura = 1.45;

*//luego generamos nuestra variable controladora, en donde realizamos la consulta, si altura es mayor a 1.30 y edad mayor a 12*

let puedeSubir = edad > 12 && altura > 1.3;

*// al realizarse la ejecución del código veremos que el valor resultante de esta comparación es true ya que ambas comparaciones son verdaderas*

console.log(puedeSubir);

1. Si no hay suficiente luz o el objeto se mueve rápidamente, la cámara de fotos debe usar el flash.

*//de nuevo creamos nuestras variables*

let luzSuficiente = false;

let velocidadDelObjetoEnKm = 160;

*// en este caso al uno de nuestros valores ya ser un booleano, no necesitamos hacer la comparación entre sí la luz Suficiente es o no igual a verdadero, ya que dentro tiene un valor booleano, lo que en definitiva se termina evaluando*

let debeUsarFlash = luzSuficiente || velocidadDelObjetoEnKm > 150;

*//finalmente imprimimos por consola el resultado para verificar que en efecto debe utilizarse el flash*

console.log(debeUsarFlash);

1. Un estudiante pasa de nivel si su nota es mayor a 7 en sus dos evaluaciones parciales, o si obtiene un 4 en el examen final.

*//nuestras variables de uso*

let parcialUno = 6;

let parcialDos = 8;

let examenFinal = 6;

*// Si bien js separa en terminos, nos aseguramos de que las validaciones de los parciales vayan en conjunto pero separados de la validacion del examen final*

let alumnoAprobado = (parcialUno >= 7 && parcialDos >= 7) || examenFinal >= 4;

1. Dejamos ver la TV a nuestro hijo si realizó la tarea pero además, si tocó sus prácticas de piano y lo hizo de memoria.

*//nuestras variables de uso, en este caso todas booleanas*

let realizoTarea = false;

let practicoPiano = true;

let deMemoria = true;

*//como mencionamos antes, cuando los valores a evaluar son booleanos ya de por si, no necesitamos compararlos con verdadero, simplemente utilizar las variables va a indicarnos si lo que queremos evaluar es true o false*

let verTV = realizoTarea && practicoPiano && deMemoria;

*//en este caso, al ser false realizoTarea, no importa lo que se resuelva en el parentesis, ya que ambas partes deberian ser true, debido al operador &&*

console.log(verTV);