## PROTECO 38 Edivaldo Gómez Hernández Prebecario No. 6



## Operadores en JavaScript

Nombre	Operador abreviado	Significado
Operadores de asignación	x = y	x = y
Asignación de adición	x += y	x = x + y
Asignación de sustracción	x -= y	x = x - y
Asignación de multiplicación	x *= y	x = x * y
Asignación de división	x /= y	x = x / y
Asignación de resto	x %= y	x = x % y
Asignación de exponenciación	x **= y	x = x ** y
Asignación de desplazamiento a la izquierda	x <<= y	$x = x \ll y$
Asignación de desplazamiento a la derecha	x >>= y	$x = x \gg y$
Asignación de desplazamiento a la derecha sin signo	x >>>= y	$x = x \gg y$
Asignación AND binaria	x &= y	x = x & y
Asignación XOR binaria	x ^= y	$x = x \wedge y$
Asignación OR binaria	x  = y	x = x   y

Tabla 1: Operadores de asignación

Operador	Descripción	Ejemplos devolviendo true
Igualdad (==)	Devuelve true si ambos operandos son iguales.	3 == var1 "3" == var1 3 == "3"
Desigualdad (!=)	Devuelve true si ambos operandos no son iguales.	var1 != 4 var2 != "3"
Estrictamente iguales	Devuelve true si los operandos son igual y tienen el mismo tipo. Mira también Object.is y sameness in JS.	3 === var1
Estrictamente desiguales (!==)	Devuelve true si los operandos no son iguales y/o no son del mismo tipo.	var1 !== "3" 3 !== "3"
Mayor que (>)	Devuelve true si el operando de la izquierda es mayor que el operando de la derecha.	var2 > var1 "12" > 2
Mayor o igual que (>=)	Devuelve true si el operando de la izquierda es mayor o igual que el operando de la derecha.	var2 >= var1 var1 >= 3
Menor que (<)	Devuelve true si el operando de la izquierda es menor que el operando de la derecha.	var1 < var2 "2" < 12
Menor o igual que (<=)	Devuelve true si el operando de la izquierda es menor o igual que el operando de la derecha.	var1 <= var2 var2 <= 5

Tabla 2: Operadores de comparación

Operador	Descripción	Ejemplo
Resto (%)	Operador binario correspondiente al módulo de una operación. Devuelve el resto de la división de dos operandos.	12 % 5 devuelve 2.
Incremento (++)	Operador unario. Incrementa en una unidad al operando. Si es usado antes del operando $(++x)$ devuelve el valor del operando después de añadirle 1 y si se usa después del operando $(x++)$ devuelve el valor de este antes de añadirle 1.	Si x es 3, entonces ++x establece x a 4 y devuelve 4, mientras que x++ devuelve 3 y, solo después de devolver el valor, establece x a 4.
Decremento ()	Operador unario. Resta una unidad al operando. Dependiendo de la posición con respecto al operando tiene el mismo comportamiento que el operador de incremento.	Si x es 3, entoncesx establece x a 2 y devuelve 2, mientras que x devuelve 3 y, solo después de devolver el valor, establece x a 2.
Negación Unaria	Operación unaria. Intenta convertir a número al operando y devuelve su forma negativa.	-"3" devuelve -3. -true devuelve -1.
Unario positivo (+)	Operación unaria. Intenta convertir a número al operando.	+"3" devuelve 3. +true devuelve 1.
Exponenciación (**) 🕹	Calcula la potencia de la base al valor del exponente. Es equivalente a base <sup>exponente</sup>	2 ** 3 devuelve 8. 10 ** -1 devuelve 0.1.

Tabla 3: Operadores aritméticos

Operador	Uso	Descripción
AND bit a bit	a &	Devuelve uno por cada posición de bit en la cuales los bits correspondientes de ambos operandos tienen valor uno.
OR bit a bit	a   b	Devuelve uno por cada posición de bit en la cual al menos uno de los bits correspondientes de ambos operandos tiene valor uno.
XOR bit a bit	a ^	Por cada posición de bit en la cual los bits correspondientes de ambos operandos son iguales devuelve cero y si son diferentes devuelve uno.
NOT bit a bit	~ a	Invierte los bits del operando.
Desplazamiento a la izquierda	a <<	Desplaza <b>b</b> posiciones a la izquierda la representación binaria de <b>a</b> , el exceso de bits desplazados a la izquierda se descarta, dejando ceros a la derecha de los bits desplazados.
Desplazamiento a la derecha con propagación de signo	a >>	Desplaza ${\bf b}$ posiciones a la derecha la representación binaria de ${\bf a}$ , el exceso de bits desplazados a la derecha se descarta.
Desplazamiento a la derecha con relleno de ceros	a >>> b	Desplaza <b>b</b> posiciones a la derecha la representación binaria de <b>a</b> , el exceso de bits desplazados a la derecha se descarta, dejando ceros a la izquierda de los bits desplazados.

Tabla 4: Operadores bit a bit

Operador	Uso	Descripción
AND Lógico (&&)	expr1 && expr2	Devuelve expr1 si puede ser convertido a false de lo contrario devuelve expr2. Por lo tanto, cuando se usa con valores booleanos, && devuelve true si ambos operandos son true, en caso contrario devuelve false.
OR Lógico	expr1    expr2	Devuelve expr1 si puede ser convertido a true de lo contrario devuelve expr2. Por lo tanto, cuando se usa con valores booleanos,    devuelve true si alguno de los operandos es true, o false si ambos son false.
NOT Lógico	!expr	Devuelve false si su operando puede ser convertido a true, en caso contrario, devuelve true.

Tabla 5: Operadores lógicos

Ejemplo 1: Uso de AND

Ejemplo 2: Uso de corrimiento izquierda

## Referencias

[1] Operadores y Expresiones. MDN Web Doc. Consultado el 08 de abril de 2019. Disponible en línea en: https://developer.mozilla.org/es/docs/Web/JavaScript/Guide/Expressions\_and\_Operators#Opera dores\_l%C3%B3gicos