

Operadores en JavaScript

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

Tabla 1: Operadores de asignación

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

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.	<code>12 % 5</code> devuelve <code>2</code> .
Incremento (++)	Operador unario. Incrementa en una unidad al operando. Si es usado antes del operando (<code>++x</code>) devuelve el valor del operando después de añadirle 1 y si se usa después del operando (<code>x++</code>) devuelve el valor de este antes de añadirle 1.	Si <code>x</code> es <code>3</code> , entonces <code>++x</code> establece <code>x</code> a <code>4</code> y devuelve <code>4</code> , mientras que <code>x++</code> devuelve <code>3</code> y solo después de devolver el valor, establece <code>x</code> a <code>4</code> .
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 <code>x</code> es <code>3</code> , entonces <code>--x</code> establece <code>x</code> a <code>2</code> y devuelve <code>2</code> , mientras que <code>x--</code> devuelve <code>3</code> y solo después de devolver el valor, establece <code>x</code> a <code>2</code> .
Negación Unaria (-)	Operación unaria. Intenta convertir a número al operando y devuelve su forma negativa.	<code>- "3"</code> devuelve <code>-3</code> . <code>-true</code> devuelve <code>-1</code> .
Unario positivo (+)	Operación unaria. Intenta convertir a número al operando.	<code>+ "3"</code> devuelve <code>3</code> . <code>+true</code> devuelve <code>1</code> .
Exponenciación (**) 	Calcula la potencia de la base al valor del exponente. Es equivalente a <code>base^{exponente}</code>	<code>2 ** 3</code> devuelve <code>8</code> . <code>10 ** -1</code> devuelve <code>0.1</code> .

Tabla 3: Operadores aritméticos

Operador	Uso	Descripción
AND bit a bit	<code>a & b</code>	Devuelve uno por cada posición de bit en la cuales los bits correspondientes de ambos operandos tienen valor uno.
OR bit a bit	<code>a b</code>	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	<code>a ^ b</code>	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	<code>~ a</code>	Invierte los bits del operando.
Desplazamiento a la izquierda	<code>a << b</code>	Desplaza <code>b</code> posiciones a la izquierda la representación binaria de <code>a</code> , 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	<code>a >> b</code>	Desplaza <code>b</code> posiciones a la derecha la representación binaria de <code>a</code> , el exceso de bits desplazados a la derecha se descarta.
Desplazamiento a la derecha con relleno de ceros	<code>a >>> b</code>	Desplaza <code>b</code> posiciones a la derecha la representación binaria de <code>a</code> , 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

