

**MDN Plus** now available in [your](#) country! Support MDN [and](#) make it your own. [Learn more](#) ✨

This page was translated from English by the community. [Learn more](#) and join the MDN Web Docs community.

## Expresiones y operadores

Este capítulo documenta todos los operadores, expresiones y palabras clave del lenguaje JavaScript.

### Expresiones y operadores por categoría

Para obtener una lista alfabética, consulta la barra lateral de la izquierda.

#### Expresiones primarias

Palabras clave básicas y expresiones generales en JavaScript.

##### [this](#)

La palabra clave `this` se refiere a una propiedad especial de un contexto de ejecución.

##### [function](#)

La palabra clave `function` define una expresión de función.

##### [class](#)

La palabra clave `class` define una expresión de clase.

##### [function](#)

La palabra clave `function*` define una expresión de función generadora.

##### [yield](#)

Pausar y reanudar una función generadora.

##### [yield\\*](#)

Delegar a otra función generadora u objeto iterable.

##### [async function](#)

La `async function` define una expresión de función asíncrona.

##### [await](#)

Pausa y reanuda una función asíncrona y espera la resolución/rechazo de la promesa.

##### [\[.\]](#)

Iniciador de arreglo/sintaxis literal.

[{.}\\_\(en-US\)](#)

Iniciador de objeto/sintaxis literal.

[/ab+c/i](#)

Sintaxis de expresión regular literal.

[\( \)](#)

Operador de agrupación.

## Expresiones del lado izquierdo

Los valores de la izquierda son el destino de una asignación.

### [Propiedad accessors](#)

Los operadores miembro proporcionan acceso a una propiedad o método de un objeto ( `object.property` y `object["property"]` ).

[new](#)

El operador `new` crea una instancia auxiliado por un constructor.

[new.target](#)

En los constructores, `new.target` se refiere al constructor que fue invocado por [new](#) .

[super](#)

La palabra clave `super` llama al constructor padre.

[...obj](#)

La sintaxis de extensión permite expandir una expresión en lugares donde se esperan múltiples argumentos (para llamadas a funciones) o múltiples elementos (para arreglos literales).

## Incremento y decremento

Operadores de incremento sufijo/prefijo y decremento sufijo/prefijo.

[A++\\_\(en-US\)](#)

Operador de incremento sufijo.

[A--](#)

Operador de incremento sufijo.

[++A\\_\(en-US\)](#)

Operador de incremento prefijo.

[--A](#)

Operador de decremento prefijo.

## Operadores unarios

Una operación unaria es una operación con un solo operando.

[delete](#)

El operador `delete` elimina una propiedad de un objeto.

### [void](#)

El operador `void` descarta el valor de retorno de una expresión.

### [typeof](#)

El operador `typeof` determina el tipo de un objeto dado.

### [+ \(en-US\)](#)

El operador unario `+` convierte su operando al tipo `Number`.

### [- \(en-US\)](#)

El operador unario de negación convierte su operando al tipo `Number` y luego lo niega.

### [~ \(en-US\)](#)

Operador `NOT` bit a bit.

### [! \(en-US\)](#)

Operador `NOT` lógico.

## Operadores aritméticos

Los operadores aritméticos toman valores numéricos (ya sean literales o variables) como sus operandos y devuelven un solo valor numérico.

### $\pm$

Operador de adición o suma.

### $=$

Operador de sustracción o resta.

### $\angle$

Operador de división.

### [\\* \(en-US\)](#)

Operador de multiplicación.

### $\%$

Operador de residuo.

## Operadores relacionales

Un operador de comparación compara sus operandos y devuelve un valor `Boolean` basado en si la comparación es verdadera o no.

### [in](#)

El operador `in` determina si un objeto tiene una determinada propiedad.

### [instanceof](#)

El operador `instanceof` determina si un objeto es una instancia de otro objeto.

### [<\\_\(en-US\)](#)

Operador menor que.

### [>\\_\(en-US\)](#)

Operador mayor que.

### [<=\\_\(en-US\)](#)

Operador menor o igual a.

### [>=\\_\(en-US\)](#)

Operador mayor o igual a.

**Nota:** `Nota: =>` no es un operador, sino la notación para [Funciones de flecha](#).

## Operadores de igualdad

El resultado de evaluar un operador de igualdad siempre es de tipo `Boolean` basado en si la comparación es verdadera.

### [==](#)

Operador de igualdad.

### [!=\\_\(en-US\)](#)

Operador de desigualdad.

### [===](#)

Operador de igualdad estricta.

### [!==\\_\(en-US\)](#)

Operador de desigualdad estricta.

## Operadores de desplazamiento de bits

Operaciones para cambiar todos los bits del operando.

### [<<\\_\(en-US\)](#)

Operador de desplazamiento bit a bit a la izquierda.

### [>>\\_\(en-US\)](#)

Operador de desplazamiento bit a bit a la derecha.

### [>>>\\_\(en-US\)](#)

Operador de desplazamiento bit a bit a la derecha sin signo.

## Operadores binarios bit a bit

Los operadores bit a bit tratan a sus operandos como un conjunto de 32 bits (ceros y unos) y devuelven valores numéricos estándar de JavaScript.

[& \(en-US\)](#)

AND bit a bit.

[|. \(en-US\)](#)

OR bit a bit.

[^ \(en-US\)](#)

XOR bit a bit.

## Operadores lógicos binarios

Los operadores lógicos se utilizan normalmente con valores booleanos (lógicos); cuando lo son, devuelven un valor booleano.

[&& \(en-US\)](#)

AND lógico.

[||. \(en-US\)](#)

OR lógico.

## Operador condicional (ternario)

[\(condition ? ifTrue : ifFalse\).](#)

El operador condicional devuelve uno de dos valores según el valor lógico de la condición.

## Operadores de asignación

Un operador de asignación asigna un valor a su operando izquierdo basándose en el valor de su operando derecho.

[=](#)

Operador de asignación.

[\\*= \(en-US\)](#)

Asignación de multiplicación.

[/= \(en-US\)](#)

Asignación de división.

[%= \(en-US\)](#)

Asignación de residuo.

[+= \(en-US\)](#)

Asignación de suma.

[-= \(en-US\)](#)

Asignación de sustracción

[<<= \(en-US\)](#)

Asignación de desplazamiento a la izquierda.

[>>= \(en-US\)](#)

Asignación de desplazamiento a la derecha.

```
>>>= \_\(en-US\)
```

Asignación de desplazamiento a la derecha sin signo.

```
&= \_\(en-US\)
```

Asignación de AND bit a bit.

```
^= \_\(en-US\)
```

Asignación de XOR bit a bit.

```
.= \_\(en-US\)
```

Asignación de OR bit a bit.

```
&&= \_\(en-US\)
```

Asignación de AND lógico.

```
||= \_\(en-US\)
```

Asignación de OR lógico.

```
??= \_\(en-US\)
```

Asignación de anulación lógica.

```
[a, b] = [1, 2]. {a, b} = {a:1, b:2}.
```

La desestructuración te permite asignar las propiedades de un arreglo u objeto a variables utilizando una sintaxis que se parece a los arreglos u objetos literales.

Operador coma

```
 ,
```

El operador `coma` permite evaluar múltiples expresiones en una sola declaración y devuelve el resultado de la última expresión.

Especificaciones

Especificación
<a href="#">ECMAScript (ECMA-262)</a> <a href="#">La definición de 'ECMAScript Language: Expressions' en esta especificación.</a>

Compatibilidad del navegador

[Report problems with this compatibility data on GitHub](#)

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	
<a href="#">Addition ( <math>\pm</math> ).</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for Android 4	
<a href="#">Addition assignment ( <math>x \pm= y</math> ).</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for Android 4	
<a href="#">Assignment ( <math>x = y</math> ).</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for Android 4	
<a href="#">async function expression</a>	Chrome 55	Edge 15	Firefox 52	Opera 42	Safari 10.1	Chrome 55 Android	Firefox for Android 52	
<a href="#">async function* expression</a>	Chrome 63	Edge 79	Firefox 55	Opera 50	Safari 12	Chrome 63 Android	Firefox for Android 55	
<a href="#">await</a>	Chrome 55	Edge 14	Firefox 52	Opera 42	Safari 10.1	Chrome 55 Android	Firefox for Android 52	
<a href="#">Use at module top level</a>	Chrome 89	Edge 89	Firefox 89	Opera 75	Safari 15	Chrome 89 Android	Firefox for Android 89	
<a href="#">Bitwise AND ( <math>a \&amp; b</math> ).</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for Android 4	
<a href="#">Bitwise AND assignment ( <math>x \&amp;= y</math> ).</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for Android 4	
<a href="#">Bitwise NOT ( <math>\sim a</math> ).</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for Android 4	
<a href="#">Bitwise OR ( <math>a   b</math> ).</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for Android 4	
<a href="#">Bitwise OR assignment ( <math>x  = y</math> ).</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for Android 4	
<a href="#">Bitwise XOR ( <math>a \wedge b</math> ).</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for Android 4	
<a href="#">Bitwise XOR assignment ( <math>x \wedge= y</math> ).</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for Android 4	
<a href="#">class</a>	Chrome 42	Edge 13	Firefox 45	Opera 29	Safari 7	Chrome 42 Android	Firefox for Android 45	
<a href="#">Comma operator</a>	Chrome 1	Edge 12	Firefox 1	Opera 4	Safari 1	Chrome 18 Android	Firefox for Android 4	
<a href="#">Conditional operator ( <math>c ? t : f</math> ).</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for Android 4	
<a href="#">Decrement ( <math>--</math> ).</a>	Chrome 2	Edge 12	Firefox 1	Opera 3	Safari 4	Chrome 18 Android	Firefox for Android 4	

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	
<a href="#">delete</a>	Chrome 1	Edge 12	Firefox 1	Opera 9	Safari 1	Chrome 18 Android	Firefox for 4 Android	
<a href="#">Destructuring assignment</a>	Chrome 49	Edge 14	Firefox 41	Opera 36	Safari 8	Chrome 49 Android	Firefox 41 for Android	
Computed property names	Chrome 49	Edge 14	Firefox 41	Opera 36	Safari 10	Chrome 49 Android	Firefox for 41 Android	
Rest in arrays	Chrome 49	Edge 16	Firefox 41	Opera 36	Safari 9.1	Chrome 49 Android	Firefox for 41 Android	
Rest in objects	Chrome 60	Edge 79	Firefox 55	Opera 47	Safari 11.1	Chrome 60 Android	Firefox for 55 Android	
<a href="#">Division ( / ).</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for 4 Android	
<a href="#">Division assignment ( x /= y ).</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for 4 Android	
<a href="#">Equality ( a == b ).</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for 4 Android	
<a href="#">Exponentiation ( ** ).</a>	Chrome 52	Edge 14	Firefox 52	Opera 39	Safari 10.1	Chrome 52 Android	Firefox for 52 Android	
<a href="#">Exponentiation assignment ( x **= y ).</a>	Chrome 52	Edge 14	Firefox 52	Opera 39	Safari 10.1	Chrome 52 Android	Firefox for 52 Android	
<a href="#">function expression</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for 4 Android	
Trailing comma in parameters	Chrome 58	Edge 14	Firefox 52	Opera 45	Safari 10	Chrome 58 Android	Firefox for 52 Android	
<a href="#">function* expression</a>	Chrome 49	Edge 12	Firefox 26	Opera 36	Safari 10	Chrome 49 Android	Firefox for 26 Android	
Trailing comma in parameters	Chrome 58	Edge 79	Firefox 52	Opera 45	Safari 10	Chrome 58 Android	Firefox for 52 Android	
<a href="#">Greater than ( a &gt; b ).</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for 4 Android	
<a href="#">Greater than or equal ( a &gt;= b ).</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for 4 Android	
<a href="#">Grouping operator ( . ).</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for 4 Android	



	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android
<a href="#">import</a>	Chrome 63	Edge 79	Firefox 67	Opera 50	Safari 11.1	Chrome 63 Android	Firefox for 67 Android
<a href="#">import.meta</a>	Chrome 64	Edge 79	Firefox 62	Opera 51	Safari 11.1	Chrome 64 Android	Firefox for 62 Android
<a href="#">in</a>	Chrome 1	Edge 12	Firefox 1	Opera 4	Safari 1	Chrome 18 Android	Firefox for 4 Android
<a href="#">Increment</a> ( <code>++</code> )	Chrome 2	Edge 12	Firefox 1	Opera 3	Safari 4	Chrome 18 Android	Firefox for 4 Android
<a href="#">Inequality</a> ( <code>a != b</code> )	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for 4 Android
<a href="#">instanceof</a>	Chrome 1	Edge 12	Firefox 1	Opera 4	Safari 1	Chrome 18 Android	Firefox for 4 Android
<a href="#">Bitwise left shift</a> ( <code>a &lt;&lt; b</code> )	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for 4 Android
<a href="#">Left shift assignment</a> ( <code>x &lt;&lt;= y</code> )	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for 4 Android
<a href="#">Less than</a> ( <code>a &lt; b</code> )	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for 4 Android
<a href="#">Less than or equal</a> ( <code>a &lt;= b</code> )	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for 4 Android
<a href="#">Logical AND</a> ( <code>&amp;&amp;</code> )	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for 4 Android
<a href="#">Logical AND assignment</a> ( <code>x &amp;&amp;= y</code> )	Chrome 85	Edge 85	Firefox 79	Opera 71	Safari 14	Chrome 85 Android	Firefox for 79 Android
<a href="#">Logical NOT</a> ( <code>!</code> )	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for 4 Android
<a href="#">Logical nullish assignment</a> ( <code>x ??= y</code> )	Chrome 85	Edge 85	Firefox 79	Opera 71	Safari 14	Chrome 85 Android	Firefox for 79 Android
<a href="#">Logical OR</a> ( <code>  </code> )	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for 4 Android
<a href="#">Logical OR assignment</a> ( <code>x   = y</code> )	Chrome 85	Edge 85	Firefox 79	Opera 71	Safari 14	Chrome 85 Android	Firefox for 79 Android
<a href="#">Multiplication</a> ( <code>*</code> )	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for 4 Android
<a href="#">Multiplication assignment</a> ( <code>x *= y</code> )	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for 4 Android
<a href="#">new</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for 4 Android

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android
<a href="#">new.target</a>	Chrome 46	Edge 13	Firefox 41	Opera 33	Safari 11	Chrome 46 Android	Firefox for 41 Android
<a href="#">null</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for 4 Android
<a href="#">Nullish coalescing operator ( ?? )</a>	Chrome 80	Edge 80	Firefox 72	Opera 67	Safari 13.1	Chrome 80 Android	Firefox for 79 Android
<a href="#">Object initializer</a>	Chrome 1	Edge 12	Firefox 1	Opera 4	Safari 1	Chrome 18 Android	Firefox for 4 Android
Computed property names	Chrome 47	Edge 12	Firefox 34	Opera 34	Safari 8	Chrome 47 Android	Firefox for 34 Android
Shorthand method names	Chrome 47	Edge 12	Firefox 34	Opera 34	Safari 9	Chrome 47 Android	Firefox for 34 Android
Shorthand property names	Chrome 47	Edge 12	Firefox 33	Opera 34	Safari 9	Chrome 47 Android	Firefox for 33 Android
Spread properties	Chrome 60	Edge 79	Firefox 55	Opera 47	Safari 11.1	Chrome 60 Android	Firefox for 55 Android
<a href="#">Optional chaining operator ( ?. )</a>	Chrome 80	Edge 80	Firefox 74	Opera 67	Safari 13.1	Chrome 80 Android	Firefox for 79 Android
<a href="#">Property accessors</a>	Chrome 1	Edge 12	Firefox 1	Opera 4	Safari 1	Chrome 18 Android	Firefox for 4 Android
<a href="#">Remainder ( % )</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for 4 Android
<a href="#">Remainder assignment ( x %= y )</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for 4 Android
<a href="#">Bitwise right shift ( a &gt;&gt; b )</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for 4 Android
<a href="#">Right shift assignment ( x &gt;&gt;= y )</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome 18 Android	Firefox for 4 Android
<a href="#">Spread syntax (...)</a>	Chrome 46	Edge 12	Firefox 16	Opera 37	Safari 8	Chrome 46 Android	Firefox for 16 Android
<a href="#">Spread in array literals</a>	Chrome 46	Edge 12	Firefox 16	Opera 37	Safari 8	Chrome 46 Android	Firefox for 16 Android
<a href="#">Spread in function calls</a>	Chrome 46	Edge 12	Firefox 27	Opera 37	Safari 8	Chrome 46 Android	Firefox for 27 Android
<a href="#">Spread in object literals</a>	Chrome 60	Edge 79	Firefox 55	Opera 47	Safari 11.1	Chrome 60 Android	Firefox for 55 Android

	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android	
<a href="#">Strict equality ( a === b ).</a>	Chrome 1	Edge 12	Firefox 1	Opera 4	Safari 1	Chrome Android 18	Firefox for Android 4	
<a href="#">Strict inequality ( a !== b ).</a>	Chrome 1	Edge 12	Firefox 1	Opera 4	Safari 1	Chrome Android 18	Firefox for Android 4	
<a href="#">Subtraction ( - ).</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome Android 18	Firefox for Android 4	
<a href="#">Subtraction assignment ( x -= y ).</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome Android 18	Firefox for Android 4	
<a href="#">super</a>	Chrome 42	Edge 13	Firefox 45	Opera 29	Safari 7	Chrome Android 42	Firefox for Android 45	
<a href="#">this</a>	Chrome 1	Edge 12	Firefox 1	Opera 9.5	Safari 1	Chrome Android 18	Firefox for Android 4	
<a href="#">typeof</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome Android 18	Firefox for Android 4	
<a href="#">Unary negation ( - ).</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome Android 18	Firefox for Android 4	
<a href="#">Unary plus ( + ).</a>	Chrome 1	Edge 12	Firefox 1	Opera 4	Safari 1	Chrome Android 18	Firefox for Android 4	
<a href="#">Bitwise unsigned right shift ( a &gt;&gt;&gt; b ).</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome Android 18	Firefox for Android 4	
<a href="#">Unsigned right shift assignment ( x &gt;&gt;&gt;= y ).</a>	Chrome 1	Edge 12	Firefox 1	Opera 3	Safari 1	Chrome Android 18	Firefox for Android 4	
<a href="#">void</a>	Chrome 1	Edge 12	Firefox 1	Opera 4	Safari 3.1	Chrome Android 18	Firefox for Android 4	
<a href="#">yield</a>	Chrome 39	Edge 12	Firefox 26	Opera 26	Safari 10	Chrome Android 39	Firefox for Android 26	
<a href="#">yield*</a>	Chrome 39	Edge 12	Firefox 27	Opera 26	Safari 10	Chrome Android 39	Firefox for Android 27	

Tip: you can click/tap on a cell for more information.

Full support

No support

See implementation notes.

User must explicitly enable this feature.

Has more compatibility info.

Ve también

- [Operator de precedencia](#)

Last modified: 27 sept 2022, [by MDN contributors](#)

