

# Expressions and operators

This chapter documents all the JavaScript language operators, expressions and keywords.

## Expressions and operators by category

For an alphabetical listing see the sidebar on the left.

### Primary expressions

Basic keywords and general expressions in JavaScript. These expressions have the highest precedence (higher than [operators](#)).

#### [this](#)

The `this` keyword refers to a special property of an execution context.

#### [Literals](#)

Basic `null`, boolean, number, and string literals.

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

Array initializer/literal syntax.

#### [{}](#)

Object initializer/literal syntax.

#### [function](#)

The `function` keyword defines a function expression.

#### [class](#)

The `class` keyword defines a class expression.

#### [function\\*](#)

The `function*` keyword defines a generator function expression.

#### [async function](#)

The `async function` defines an async function expression.

#### [async function\\*](#)

The `async function*` keywords define an async generator function expression.

#### [/ab+c/i](#)

Regular expression literal syntax.

#### [`string`](#)

Template literal syntax.

[\( \)](#)

Grouping operator.

## Left-hand-side expressions

Left values are the destination of an assignment.

### [Property accessors](#)

Member operators provide access to a property or method of an object ( `object.property` and `object["property"]` ).

[?.](#)

The optional chaining operator returns `undefined` instead of causing an error if a reference is [nullish](#) ( `null` or `undefined` ).

[new](#)

The `new` operator creates an instance of a constructor.

[new.target](#)

In constructors, `new.target` refers to the constructor that was invoked by `new`.

[import.meta](#)

An object exposing context-specific metadata to a JavaScript module.

[super](#)

The `super` keyword calls the parent constructor or allows accessing properties of the parent object.

[import\(\)](#)

The `import()` syntax allows loading a module asynchronously and dynamically into a potentially non-module environment.

## Increment and decrement

Postfix/prefix increment and postfix/prefix decrement operators.

[A++](#)

Postfix increment operator.

[A--](#)

Postfix decrement operator.

[++A](#)

Prefix increment operator.

[--A](#)

Prefix decrement operator.

## Unary operators

A unary operation is an operation with only one operand.

#### `delete`

The `delete` operator deletes a property from an object.

#### `void`

The `void` operator evaluates an expression and discards its return value.

#### `typeof`

The `typeof` operator determines the type of a given object.

#### `+`

The unary plus operator converts its operand to Number type.

#### `-`

The unary negation operator converts its operand to Number type and then negates it.

#### `~`

Bitwise NOT operator.

#### `!`

Logical NOT operator.

#### `await`

Pause and resume an async function and wait for the promise's fulfillment/rejection.

## Arithmetic operators

Arithmetic operators take numerical values (either literals or variables) as their operands and return a single numerical value.

#### `**`

Exponentiation operator.

#### `*`

Multiplication operator.

#### `/`

Division operator.

#### `%`

Remainder operator.

#### `+` (Plus)

Addition operator.

#### `-`

Subtraction operator.

## Relational operators

A comparison operator compares its operands and returns a boolean value based on whether the comparison is true.

`<` (Less than)

Less than operator.

`>` (Greater than)

Greater than operator.

`<=`

Less than or equal operator.

`>=`

Greater than or equal operator.

`instanceof`

The `instanceof` operator determines whether an object is an instance of another object.

`in`

The `in` operator determines whether an object has a given property.

**Note:** `=>` is not an operator, but the notation for [Arrow functions](#).

## Equality operators

The result of evaluating an equality operator is always of type boolean based on whether the comparison is true.

`==`

Equality operator.

`!=`

Inequality operator.

`===`

Strict equality operator.

`!==`

Strict inequality operator.

## Bitwise shift operators

Operations to shift all bits of the operand.

`<<`

Bitwise left shift operator.

>>

Bitwise right shift operator.

>>>

Bitwise unsigned right shift operator.

## Binary bitwise operators

Bitwise operators treat their operands as a set of 32 bits (zeros and ones) and return standard JavaScript numerical values.

&

Bitwise AND.

|

Bitwise OR.

^

Bitwise XOR.

## Binary logical operators

Logical operators implement boolean (logical) values and have [short-circuiting](#) behavior.

&&

Logical AND.

||

Logical OR.

??

Nullish Coalescing Operator.

## Conditional (ternary) operator

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

The conditional operator returns one of two values based on the logical value of the condition.

## Assignment operators

An assignment operator assigns a value to its left operand based on the value of its right operand.

=

Assignment operator.

\*=

Multiplication assignment.

`/=`

Division assignment.

`%=`

Remainder assignment.

`+=`

Addition assignment.

`-=`

Subtraction assignment

`<<=`

Left shift assignment.

`>>=`

Right shift assignment.

`>>>=`

Unsigned right shift assignment.

`&=`

Bitwise AND assignment.

`^=`

Bitwise XOR assignment.

`|=`

Bitwise OR assignment.

`**=`

Exponentiation assignment.

`&&=`

Logical AND assignment.

`||=`

Logical OR assignment.

`??=`

Nullish coalescing assignment.

`[a, b] = arr, {a, b} = obj`

Destructuring assignment allows you to assign the properties of an array or object to variables using syntax that looks similar to array or object literals.

## Yield operators

`yield`

Pause and resume a generator function.

`yield*`

Delegate to another generator function or iterable object.

## Spread syntax

`...obj`

Spread syntax allows an iterable, such as an array or string, to be expanded in places where zero or more arguments (for function calls) or elements (for array literals) are expected. In an object literal, the spread syntax enumerates the properties of an object and adds the key-value pairs to the object being created.

## Comma operator

`,`

The comma operator allows multiple expressions to be evaluated in a single statement and returns the result of the last expression.

## Specifications

Specification
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-addition-operator-plus</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-assignment-operators</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-async-function-definitions</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-async-generator-function-definitions</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># prod-BitwiseANDExpression</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-bitwise-not-operator</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># prod-BitwiseORExpression</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># prod-BitwiseXORExpression</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-class-definitions</a>

Specification
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-comma-operator</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-conditional-operator</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-postfix-decrement-operator</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-delete-operator</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-destructuring-assignment</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-destructuring-binding-patterns</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-multiplicative-operators</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-equality-operators</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-exp-operator</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-function-definitions</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-generator-function-definitions</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-relational-operators</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-grouping-operator</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-import-calls</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># prod-ImportMeta</a>
<a href="#">HTML Standard</a> <a href="#"># hostgetimportmetaproperties</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-postfix-increment-operator</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-left-shift-operator</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># prod-LogicalANDExpression</a>



Specification
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-logical-not-operator</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># prod-LogicalORExpression</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-new-operator</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-built-in-function-objects</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-null-value</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># prod-CoalesceExpression</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-object-initializer</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># prod-OptionalExpression</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-property-accessors</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-signed-right-shift-operator</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># prod-SpreadElement</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># prod-ArgumentList</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># prod-PropertyDefinition</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-subtraction-operator-minus</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-super-keyword</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-this-keyword</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-typeof-operator</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-unary-minus-operator</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-unary-plus-operator</a>

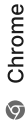
Specification
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-unsigned-right-shift-operator</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-void-operator</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># prod-YieldExpression</a>
<a href="#">ECMAScript Language Specification</a> <a href="#"># sec-generator-function-definitions-runtime-semantics-evaluation</a>


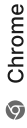
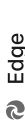
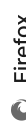
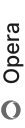
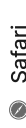
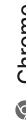
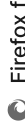
## Browser compatibility


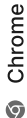

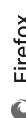
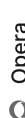
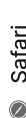
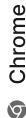

[Report problems with this compatibility data on GitHub](#)

	<div></div>						
	<div>Chrome</div>	<div>Edge</div>	<div>Firefox</div>	<div>Opera</div>	<div>Safari</div>	<div>Chrome Android</div>	<div>Firefox for Android</div>
<a href="#">Addition ( <code>±</code> ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for Android 4
<a href="#">Addition assignment ( <code>x += y</code> ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for Android 4
<a href="#">Assignment ( <code>x = y</code> ).</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 ( <code>a &amp; b</code> ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for Android 4
<a href="#">Bitwise AND assignment ( <code>x &amp;= y</code> ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for Android 4
<a href="#">Bitwise NOT ( <code>~a</code> ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for Android 4
<a href="#">Bitwise OR ( <code>a   b</code> ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for Android 4

<a href="#">Bitwise OR assignment ( <code>x  = y</code> )</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">Bitwise XOR ( <code>a ^ b</code> )</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">Bitwise XOR assignment ( <code>x ^= y</code> )</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<code>class</code>	✓ Chrome 42	✓ Edge 13	✓ Firefox 45	✓ Opera 29	✓ Safari 7	✓ Chrome 42 Android	✓ Firefox for 45 Android
<a href="#">Comma operator</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 4	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">Conditional operator ( <code>c ? t : f</code> )</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">Decrement ( <code>--</code> )</a>	✓ Chrome 2	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 4	✓ Chrome 18 Android	✓ Firefox for 4 Android
<code>delete</code>	✓ 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 ( <code>/</code> )</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">Division assignment ( <code>x /= y</code> )</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">Equality ( <code>a == b</code> )</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">Exponentiation ( <code>**</code> )</a>	✓ Chrome 52	✓ Edge 14	✓ Firefox 52	✓ Opera 39	✓ Safari 10.1	✓ Chrome 52 Android	✓ Firefox for 52 Android
<a href="#">Exponentiation assignment ( <code>x **= y</code> )</a>	✓ Chrome 52	✓ Edge 14	✓ Firefox 52	✓ Opera 39	✓ Safari 10.1	✓ Chrome 52 Android	✓ Firefox for 52 Android
<code>function</code> <a href="#">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*</a> <a href="#">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

							
	 Chrome	 Edge	 Firefox	 Opera	 Safari	 Chrome Android	 Firefox for Android
<a href="#">Greater than ( <code>a &gt; b</code> ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">Greater than or equal ( <code>a &gt;= b</code> ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">Grouping operator ( <code>()</code> ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">import</a>	✓ Chrome 63	✓ Edge 79	✓ Firefox 67	✓ Opera 50	✓ Safari 11.1	✓ Chrome 63 Android	✓ Firefox for 67 Android
The <code>options</code> parameter 	✓ Chrome 91	✓ Edge 91	✗ Firefox No *	✗ Opera No	✓ Safari 15	✓ Chrome 91 Android	✗ Firefox No * for Android
Available in workers	✓ Chrome 80	✓ Edge 80	✓ Firefox 114	✓ Opera 67	✓ Safari 15	✓ Chrome 80 Android	✓ Firefox for 114 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="#">import.meta.resolve</a>	✓ Chrome 105	✓ Edge 105	✓ Firefox 106	✓ Opera 91	✓ Safari 16.4	✓ Chrome 105 Android	✓ Firefox 106 for 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 ( <code>++</code> ).</a>	✓ Chrome 2	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 4	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">Inequality ( <code>a != b</code> ).</a>	✓ 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 ( <code>a &lt;&lt; b</code> ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">Left shift assignment ( <code>x &lt;&lt;= y</code> ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">Less than ( <code>a &lt; b</code> ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">Less than or equal ( <code>a &lt;= b</code> ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">Logical AND ( <code>&amp;&amp;</code> ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">Logical AND assignment ( <code>x &amp;&amp;= y</code> ).</a>	✓ Chrome 85	✓ Edge 85	✓ Firefox 79	✓ Opera 71	✓ Safari 14	✓ Chrome 85 Android	✓ Firefox for 79 Android

							
	 Chrome	 Edge	 Firefox	 Opera	 Safari	 Chrome Android	 Firefox for Android
<a href="#">Logical NOT ( ! ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for Android 4
<a href="#">Logical OR (    ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for Android 4
<a href="#">Logical OR assignment ( x    = y ).</a>	✓ Chrome 85	✓ Edge 85	✓ Firefox 79	✓ Opera 71	✓ Safari 14	✓ Chrome 85 Android	✓ Firefox for Android 79
<a href="#">Multiplication ( * ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for Android 4
<a href="#">Multiplication assignment ( x *= y ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for Android 4
<a href="#">new</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for Android 4
<a href="#">new.target</a>	✓ Chrome 46	✓ Edge 13	✓ Firefox 41	✓ Opera 33	✓ Safari 11	✓ Chrome 46 Android	✓ Firefox for Android 41
<a href="#">null</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for Android 4
<a href="#">Nullish coalescing operator ( ?? ).</a>	✓ Chrome 80	✓ Edge 80	✓ Firefox 72	✓ Opera 67	✓ Safari 13.1	✓ Chrome 80 Android	✓ Firefox for Android 79
<a href="#">Nullish coalescing assignment ( x ?? = y ).</a>	✓ Chrome 85	✓ Edge 85	✓ Firefox 79	✓ Opera 71	✓ Safari 14	✓ Chrome 85 Android	✓ Firefox for Android 79
<a href="#">Object initializer</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 4	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for Android 4
Computed property names	✓ Chrome 47	✓ Edge 12	✓ Firefox 34	✓ Opera 34	✓ Safari 8	✓ Chrome 47 Android	✓ Firefox for Android 34
Shorthand method names	✓ Chrome 47	✓ Edge 12	✓ Firefox 34	✓ Opera 34	✓ Safari 9	✓ Chrome 47 Android	✓ Firefox for Android 34
Shorthand property names	✓ Chrome 47	✓ Edge 12	✓ Firefox 33	✓ Opera 34	✓ Safari 9	✓ Chrome 47 Android	✓ Firefox for Android 33
Spread properties	✓ Chrome 60	✓ Edge 79	✓ Firefox 55	✓ Opera 47	✓ Safari 11.1	✓ Chrome 60 Android	✓ Firefox for Android 55
<a href="#">Optional chaining operator ( ?. ).</a>	✓ Chrome 80	✓ Edge 80	✓ Firefox 74	✓ Opera 67	✓ Safari 13.1	✓ Chrome 80 Android	✓ Firefox for Android 79
<a href="#">Property accessors</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 4	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for Android 4
<a href="#">Remainder ( % ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for Android 4

							
	 Chrome	 Edge	 Firefox	 Opera	 Safari	 Chrome Android	 Firefox for Android
<a href="#">Remainder assignment ( <code>x %= y</code> ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">Bitwise right shift ( <code>a &gt;&gt; b</code> ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">Right shift assignment ( <code>x &gt;&gt;= y</code> ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">Spread syntax ( <code>...</code> ).</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
<a href="#">Strict equality ( <code>a === b</code> ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 4	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">Strict inequality ( <code>a !== b</code> ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 4	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">Subtraction ( <code>-</code> ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">Subtraction assignment ( <code>x -= y</code> ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">super</a>	✓ Chrome 42	✓ Edge 13	✓ Firefox 45	✓ Opera 29	✓ Safari 7	✓ Chrome 42 Android	✓ Firefox for 45 Android
<a href="#">this</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 9.5	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">typeof</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">Unary negation ( <code>-</code> ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">Unary plus ( <code>+</code> ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 4	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">Bitwise unsigned right shift ( <code>a &gt;&gt;&gt; b</code> ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android
<a href="#">Unsigned right shift assignment ( <code>x &gt;&gt;&gt;= y</code> ).</a>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 3	✓ Safari 1	✓ Chrome 18 Android	✓ Firefox for 4 Android

	<div><div></div></div>						
	Chrome	Edge	Firefox	Opera	Safari	Chrome Android	Firefox for Android
<code>void</code>	✓ Chrome 1	✓ Edge 12	✓ Firefox 1	✓ Opera 4	✓ Safari 3.1	✓ Chrome 18 Android	✓ Firefox for Android 4
<code>yield</code>	✓ Chrome 39	✓ Edge 12	✓ Firefox 26 *	✓ Opera 26	✓ Safari 10	✓ Chrome 39 Android	✓ Firefox 26 * for Android
<code>yield*</code>	✓ Chrome 39	✓ Edge 12	✓ Firefox 27 *	✓ Opera 26	✓ Safari 10	✓ Chrome 39 Android	✓ Firefox 27 * for Android

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

✓ Full support   ♦ Partial support   ✕ No support   ⚠ Non-standard. Check cross-browser support before using.   \* See implementation notes.

🔧 User must explicitly enable this feature.   ... Has more compatibility info.

See also

- [Operator precedence](#)

This page was last modified on Apr 5, 2023 by [MDN contributors](#).