



JavaScript Operators

[< Previous](#)[Next >](#)

Example

Assign values to variables and add them together:

```
var x = 5;           // assign the value 5 to x
var y = 2;           // assign the value 2 to y
var z = x + y;       // assign the value 7 to z (x + y)
```

[Try it Yourself »](#)

The **assignment** operator (=) assigns a value to a variable.



HTML

CSS

JAVASCRIPT

MORE ▼



```
var x = 10;
```

[Try it Yourself »](#)

The **addition** operator (**+**) adds numbers:

Adding

```
var x = 5;  
var y = 2;  
var z = x + y;
```

[Try it Yourself »](#)

The **multiplication** operator (*****) multiplies numbers.

Multiplying

```
var x = 5;  
var y = 2;  
var z = x * y;
```



HTML

CSS

JAVASCRIPT

MORE ▼



JavaScript Arithmetic Operators

Arithmetic operators are used to perform arithmetic on numbers:

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
**	Exponentiation (<u>ES2016</u>)
/	Division
%	Modulus (Division Remainder)
++	Increment
--	Decrement



JavaScript Assignment Operators

Assignment operators assign values to JavaScript variables.

Operator	Example	Same As
=	x = y	x = y
+=	x += y	x = x + y
-=	x -= y	x = x - y
*=	x *= y	x = x * y
/=	x /= y	x = x / y
%=	x %= y	x = x % y
**=	x **= y	x = x ** y

The **addition assignment** operator (+=) adds a value to a variable.

Assignment

```
var x = 10;  
x += 5;
```



HTML

CSS

JAVASCRIPT

MORE ▼



Assignment operators are fully described in the **JS Assignment** chapter.

JavaScript String Operators

The **+** operator can also be used to add (concatenate) strings.

Example

```
var txt1 = "John";  
var txt2 = "Doe";  
var txt3 = txt1 + " " + txt2;
```

The result of txt3 will be:

John Doe

[Try it Yourself »](#)

The **+=** assignment operator can also be used to add (concatenate) strings:

```
var txt1 = "What a very ";  
txt1 += "nice day";
```

The result of txt1 will be:

What a very nice day

[Try it Yourself »](#)

When used on strings, the + operator is called the concatenation operator.

Adding Strings and Numbers

Adding two numbers, will return the sum, but adding a number and a string will return a string:

Example

```
var x = 5 + 5;  
var y = "5" + 5;  
var z = "Hello" + 5;
```

The result of x, y, and z will be:



HTML

CSS

JAVASCRIPT

MORE ▼



```
55  
Hello5
```

[Try it Yourself »](#)

If you add a number and a string, the result will be a string!

JavaScript Comparison Operators

Operator	Description
==	equal to
===	equal value and equal type
!=	not equal
!==	not equal value or not equal type
>	greater than
<	less than
>=	greater than or equal to
<=	less than or equal to



Comparison operators are fully described in the [**JS Comparisons**](#) chapter.

JavaScript Logical Operators

Operator	Description
&&	logical and
	logical or
!	logical not

Logical operators are fully described in the [**JS Comparisons**](#) chapter.

JavaScript Type Operators

Operator	Description
typeof	Returns the type of a variable



Type operators are fully described in the **JS Type Conversion** chapter.

JavaScript Bitwise Operators

Bit operators work on 32 bits numbers.

Any numeric operand in the operation is converted into a 32 bit number. The result is converted back to a JavaScript number.

Operator	Description	Example	Same as	Result	Decimal
&	AND	5 & 1	0101 & 0001	0001	1
	OR	5 1	0101 0001	0101	5
~	NOT	~ 5	~0101	1010	10
^	XOR	5 ^ 1	0101 ^ 0001	0100	4
<<	Zero fill left shift	5 << 1	0101 << 1	1010	10
>>	Signed right shift	5 >> 1	0101 >> 1	0010	2
>>>	Zero fill right shift	5 >>> 1	0101 >>> 1	0010	2

Bitwise operators are fully described in the **JS Bitwise** chapter.

Exercise:

Multiply 10 with 5, and alert the result.

```
alert(10  5);
```

Submit Answer »

Start the Exercise

[HTML](#)[CSS](#)[JAVASCRIPT](#)[MORE ▼](#)

HOW TO

Tabs
Dropdowns
Accordions
Side Navigation
Top Navigation
Modal Boxes
Progress Bars
Parallax
Login Form
HTML Includes
Google Maps
Range Sliders
Tooltips
Slideshow
Filter List
Sort List

SHARE



[HTML](#)[CSS](#)[JAVASCRIPT](#)[MORE ▼](#)[HTML](#)[CSS](#)[JavaScript](#)[SQL](#)[Python](#)[PHP](#)[jQuery](#)[Bootstrap](#)[XML](#)[Read More »](#)

[HTML](#)[CSS](#)[JAVASCRIPT](#)[MORE ▼](#)[REPORT ERROR](#)[PRINT PAGE](#)[FORUM](#)[ABOUT](#)[Top Tutorials](#)[Top References](#)

[HTML](#)[CSS](#)[JAVASCRIPT](#)[MORE ▼](#)

JavaScript Tutorial

- [How To Tutorial](#)
- [SQL Tutorial](#)
- [Python Tutorial](#)
- [W3.CSS Tutorial](#)
- [Bootstrap Tutorial](#)
- [PHP Tutorial](#)
- [jQuery Tutorial](#)
- [Java Tutorial](#)
- [C++ Tutorial](#)

Top Examples

- [HTML Examples](#)
- [CSS Examples](#)
- [JavaScript Examples](#)
- [How To Examples](#)
- [SQL Examples](#)
- [Python Examples](#)
- [W3.CSS Examples](#)
- [Bootstrap Examples](#)
- [PHP Examples](#)
- [jQuery Examples](#)
- [Java Examples](#)
- [XML Examples](#)

JavaScript Reference

- [SQL Reference](#)
- [Python Reference](#)
- [W3.CSS Reference](#)
- [Bootstrap Reference](#)
- [PHP Reference](#)
- [HTML Colors](#)
- [jQuery Reference](#)
- [Java Reference](#)
- [Angular Reference](#)

Web Certificates

- [HTML Certificate](#)
- [CSS Certificate](#)
- [JavaScript Certificate](#)
- [SQL Certificate](#)
- [Python Certificate](#)
- [jQuery Certificate](#)
- [PHP Certificate](#)
- [Bootstrap Certificate](#)
- [XML Certificate](#)

[Get Certified »](#)

W3Schools is optimized for learning, testing, and training. Examples might be simplified to improve reading and basic understanding. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using this site, you agree to have read and accepted our terms of use, cookie and privacy policy. Copyright 1999-2020 by Refsnes Data. All Rights Reserved.

Powered by W3.CSS.

