w3schools.com

THE WORLD'S LARGEST WEB DEVELOPER SITE

JavaScript Arithmetic

❮ Previous Next ❯

JavaScript Arithmetic Operators

Arithmetic operators perform arithmetic on numbers (literals or variables).

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus (Remainder)
++	Increment
--	Decrement

Arithmetic Operations

A typical arithmetic operation operates on two numbers.

The two numbers can be literals:

Example

```
var x = 100 + 50;
```

Try it Yourself »

or variables:

```
var x = a + b;
```

Try it Yourself »

or expressions:

Example

```
var x = (100 + 50) * a;
```

Try it Yourself »

Operators and Operands

The numbers (in an arithmetic operation) are called **operands**.

The operation (to be performed between the two operands) is defined by an **operator**.

Operand	Operator	Operand
100	+	50

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

Adding

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

Try it Yourself »

The **subtraction** operator (-) subtracts numbers.

Subtracting

```
var x = 5;
```

Try it Yourself »

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

Multiplying

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

Try it Yourself »

The **division** operator (/) divides numbers.

Dividing

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

Try it Yourself »

The **modular** operator (%) returns the division remainder.

Modulus

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

Try it Yourself »

The **increment** operator (++) increments numbers.

Incrementing

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

Try it Yourself »

Decrementing

```
var x = 5;
x--;
var z = x;
```

Try it Yourself »

Operator Precedence

Operator precedence describes the order in which operations are performed in an arithmetic expression.

Example

```
var x = 100 + 50 * 3;
```

Try it Yourself »

Is the result of example above the same as 150 * 3, or is it the same as 100 + 150?

Is the addition or the multiplication done first?

As in traditional school mathematics, the multiplication is done first.

Multiplication (*) and division (/) have higher **precedence** than addition (+) and subtraction (-).

And (as in school mathematics) the precedence can be changed by using parentheses:

Example

```
var x = (100 + 50) * 3;
```

Try it Yourself »

When using parentheses, the operations inside the parentheses are computed first.

When many operations have the same precedence (like addition and subtraction), they are computed from left to right:

Example

```
var x = 100 + 50 - 3;
```

Try it Yourself »

Value	Operator	Description	Example
19	( )	Expression grouping	(3 + 4)
18	.	Member	person.name
18	[]	Member	person["name"]
17	()	Function call	myFunction()
17	new	Create	new Date()
16	++	Postfix Increment	i++
16	--	Postfix Decrement	i--
15	++	Prefix Increment	++i
15	--	Prefix Decrement	--i
15	!	Logical not	!(x==y)
15	typeof	Type	typeof x
14	*	Multiplication	10 * 5
14	/	Division	10 / 5
14	%	Modulo division	10 % 5
14	**	Exponentiation	10 ** 2
13	+	Addition	10 + 5
13	-	Subtraction	10 - 5
12	<<	Shift left	x << 2
12	>>	Shift right	x >> 2
12	>>>	Shift right (unsigned)	x >>> 2
11	<	Less than	x < y

11	>	Greater than	x > y
11	>=	Greater than or equal	x >= y
10	==	Equal	x == y
10	===	Strict equal	x === y
10	!=	Unequal	x != y
10	!==	Strict unequal	x !== y
6	&&	Logical and	x && y
5	||	Logical or	x || y
3	=	Assignment	x = y
3	+=	Assignment	x += y
3	-=	Assignment	x -= y
3	*=	Assignment	x *= y
3	%=	Assignment	x %= y
3	<<=	Assignment	x <<= y
3	>>=	Assignment	x >>= y
3	>>>=	Assignment	x >>>= y
3	&=	Assignment	x &= y
3	^=	Assignment	x ^= y
3	|=	Assignment	x |= y

Pale red entries indicates experimental or proposed technology (ECMAScript 2016 or ES7)

Expressions in parentheses are fully computed before the value is used in the rest of the expression.

Test Yourself with Exercises!

Exercise 1 » Exercise 2 » Exercise 3 » Exercise 4 » Exercise 5 »