

Variable types

	var	let	const
block scoped*	✗	✓	✓
tdz**	✗	✓	✓
creates global property*	✓	✗	✗
reassignable*	✓	✓	✗
redeclarable*	✓	✗	✗

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Strict_mode
 * <https://medium.com/@allansendagi/block-scope-in-javascript-8fd2f909e848>
 ** <https://discuss.codecademy.com/t/what-is-the-difference-between-let-and-var-when-declaring-and-what-about-reassigning/492581/2>

<https://www.valentinog.com/blog/var/>
<https://www.freecodecamp.org/news/var-let-and-const-whats-the-difference/>
<https://fontawesome.com/v3/icons/>

Data types

var age = 18;	// number
var name = "Jane";	// string
var truth = false;	// boolean
var sheets = ["HTML", "CSS", "JS"];	// array
var a; typeof a;	// undefined
var a = null;	// value null

Objects:

```
var student = {
  firstName: "Jane",
  lastName: "Doe",
  age: 18,
  height: 170,
  fullName: function() {
    return this.firstName + " " + this.lastName;
  }
};
```

Data types (cont)

```
// list of properties and values

// object function

// object end

student.age = 19; // setting value

student[age]++; // incrementing

name = student.fullName(); // call object function
```

Operators

Comparison Operators

Comparison operators **compare** two values and return a boolean value, either **true** or **false**.

==	Equal to	5==5; //true
!=	Not equal to	5!=5; //false
>	Greater than	3>2; //true
>=	Greater than or equal to	3>=3; //true
<	Less than	3<2; //false
<=	Less than or equal to	2<=2; //true
===	Strict equal to	5==='5'; //false
!==	Strict not equal to	5!== '5'; //true

Logical Operators

Logical operators perform logical operations and return a boolean value, either **true** or **false**.

&&	Logical AND	true && false; // false
	Logical OR	true false; // true
!	Logical NOT	!true; // false

Operators (cont)

? : Ternary operator

Returns value based on the condition

```
(5 > 3) ? 'success' : 'error'; //

condition ? expressionIfTrue : expressionIfFalse
```

Assignment operators

Assignment operators are used to assign values to variables.

=	Assignment operator	a = 7;
+=	Addition assignment	a += 5;
*=	Multiplication Assignment	a = 3;
-=	Subtraction Assignment	a -= 2;
/=	Division Assignment	a /= 2;
%=	Remainder Assignment	a %= 2;
**=	Exponentiation Assignment	a = 2;

Arithmetic Operators

Arithmetic operators are used to perform arithmetic operations.

+	Addition	x + y
-	Subtraction	x - y
*	Multiplication	x * y
/	Division	x / y
%	Remainder	x % y



Operators (cont)

++	Increment (increments by 1)	<code>++x</code> or <code>x++</code>
--	Decrement (decrements by 1)	<code>--x</code> or <code>x--</code>
**	Exponentiation (Power)	<code>x ** y</code>

Bitwise Operators

Bitwise operators perform operations on binary representations of numbers.

& Bitwise AND

| Bitwise OR

^ Bitwise XOR

~ Bitwise NOT

<< Left shift

>> Sign-propagating right shift

>>> Zero-fill right shift

String Operators

In JavaScript, you can also use the `+` operator to concatenate (join) two or more strings.

+ Concatenation operator

Other Operators

Operators (cont)

, `let x = 1;`
evaluates `x = (x++, x);`
each of its `console.log(x);` // 2
operands
(from left to right) and
returns the value of the last operand.

delete `delete x`

deletes an object's property, or an element of an array

typeof `typeof 3; // "number"`
returns a string indicating the data type

void `void(x)`
discards the expression's return value

in `prop in object`
returns true if the specified property is in the object

Operators (cont)

instanceof `object instanceof object`
returns `true` if the specified object is of the specified object type

Loopz

https://www.w3schools.com/js/js_loop_for.asp <https://www.programiz.com/javascript/for-loop>

https://www.w3schools.com/js/js_sref_foreach.asp <https://www.programiz.com/javascript/forEach>

https://www.w3schools.com/js/js_loop_forof.asp <https://www.programiz.com/javascript/for-of>

https://www.w3schools.com/js/js_loop_forin.asp <https://www.programiz.com/javascript/for-in>

https://www.w3schools.com/js/js_loop_while.asp <https://www.programiz.com/javascript/while-loop>

https://www.w3schools.com/js/js_break.asp <https://www.programiz.com/javascript/break-statement>

https://www.w3schools.com/js/js_iterables.asp <https://www.programiz.com/javascript/continue-statement>

https://www.w3schools.com/js/js_iterables.asp

https://www.w3schools.com/js/js_switch.asp <https://www.programiz.com/javascript/switch-statement>



By **krabat1 (krabat1)**
cheatography.com/krabat1/

Not published yet.
Last updated 5th January, 2024.
Page 2 of 4.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

Global Methods

Math functions III. - Number Methods

<code>isFinite()</code>	a value is a finite number?
<code>isInteger()</code>	a value is an integer?
<code>isNaN()</code>	a value is Number.NaN?
<code>isSafeInteger()</code>	a value is a safe integer?
<code>toExponential(x)</code>	Converts a num. into an exponential notation
<code>toFixed(x)</code>	Formats a num. with x numbs of digits after the decimal point
<code>toLocaleString()</code>	Converts a num. into a string, based on the locale settings
<code>toPrecision(x)</code>	Formats a num. to x length
<code>toString()</code>	Converts a num. to a string
<code>valueOf()</code>	Returns the primitive value of a num.

Math functions I.

<code>Math.round(x.y);</code> <code>// x.5 ^ x.4</code>	<code>Math.trunc(x.y)</code> <code>// x</code>
<code>Math.ceil(); Math.floor();</code> <code>// ^ v</code>	<code>Math.pow(x,y)</code> <code>// x^y</code>
<code>Math.sign()</code> <code>// -x, 0, x → -1, 0, 1</code>	<code>Math.abs(-x);</code> <code>// x</code>
<code>Math.sqrt(x); Math.cbrt(y);</code> <code>// √x √[3]y</code>	<code>Math.exp(x)</code> <code>// value of Ex</code>
<code>Math.min(a,b,c,d); and Math.max(a,b,c,d);</code>	<code>Math.random();</code>
<code>Math.log(); Math.log2(); Math.log10();</code>	

Math functions II. - Angle functions

<code>sin(x)</code>	// sine of x (x is in rad.)
<code>asin(x)</code>	// arcsine of x, in rad.
<code>asinh(x)</code>	// hyperb. arcsine of x
<code>sinh(x)</code>	// hyperb. sine of x
<code>cos(x)</code>	// cosine of x (x is in rad.)
<code>acos(x)</code>	// arccosine of x, in rad.
<code>acosh(x)</code>	// hyperb. arccosine of x
<code>cosh(x)</code>	// hyperb. cosine of x
<code>tan(x)</code>	// tang. of an angl
<code>atan(x)</code>	// arctang. of x as a numeric value btw. -PI/2 and PI/2 rad.
<code>atan2(y, x)</code>	// arctang. of the quotient of its arguments
<code>atanh(x)</code>	// hyperb. arctang. of x
<code>tanh(x)</code>	// hyperb. tang. of a num.

String functions

length	<code>str.length</code>	Returns the number of characters in a string
substring()	<code>str.substring(indexStart, indexEnd)</code>	Returns a specified part of the string
slice()	<code>str.slice(beginIndex, endIndex)</code>	Extracts and returns a section of the string
substr()	<code>str.substr(beginIndex, length)</code>	
replace()	<code>str.replace(pattern, replacement)</code>	replace a substring/pattern in the string
replaceAll()	<code>str.replaceAll(pattern, replacement)</code>	Returns string by replacing all matching patterns
toUpperCase()	<code>str.toUpperCase()</code>	Returns uppercase representation of string
toLowerCase()	<code>str.toLowerCase()</code>	Returns lowercase representation of string
concat()	<code>str.concat(str1, ...strN)</code>	Concatenates the arguments to the calling string
repeat()	<code>str.repeat(count)</code>	Returns a string by repeating it given times
trim()	<code>str.trim()</code>	Removes whitespace from both ends of a string
padStart()	<code>str.padStart(targetLength, padString)</code>	Pads a string at the start/end to a given length
padEnd()	<code>str.padEnd(targetLength, padString)</code>	
charAt()	<code>str.charAt(index)</code>	Returns character at a specified index in string
charCodeAt()	<code>str.charCodeAt(index)</code>	Returns Unicode of the character at given index
fromCharCode()	<code>String.fromCharCode(...codePoints)</code>	Returns a string from the given UTF-16 code units
codePointAt()	<code>str.codePointAt(index)</code>	Returns the Unicode point value at given index
fromCodePoint()	<code>String.fromCodePoint(...codePoints)</code>	Returns a string using the given code points



String functions (cont)

split() `str.split(separator, limit)` limit: a number (expression)

Returns the string divided into list of substring

-- Search Methods --

indexOf() `str.indexOf(searchValue, fromIndex)`

lastIndexOf() `str.lastIndexOf(searchValue, fromIndex)`

Returns the first index/last index of occurrence of a value

search() `str.search(regex)`

Searches for specified value in the string

match() `str.match(regex)`

Returns result of matching string with a regex

matchAll() `str.matchAll(regex)`

Returns iterator of results matching with a regex

includes() `str.includes(searchString, position)`

Checks if given string is found inside a string

startsWith() `str.startsWith(searchString, position)`

endsWith() `str.endsWith(searchString, position)`

Checks if a string begins/ends with a specified string

localeCompare() `str.localeCompare(compareStr, locales, options)`

Compares two strings in the current locale

-- Template Literals -- \${...}

Template literals provide an easy way to interpolate variables and expressions into strings.

Booleans

The `Boolean()` function is used to convert various data types to boolean values.

You can use the `Boolean()` function to find out if an expression is true.

See [Comparison and Logical Operators](#).

Array Functions

`// const array_name = [item1, item2, ...]; // var firstElement =`

push() and **unshift()**

adds an element at the end/beginning of the array.

pop() and **shift()**

remove the last/first element from an array.

length

returns the number of elements in an array

Arrays are objects...

...but arrays too.

Objects

Events

Event listener

Useful links

<https://www.w3schools.com/jsref/default.asp>

mdn web docs

[educba.com - javascript-tutorial](#)

<https://www.javascripture.com/>

[Programiz - Getting Started](#)

[Codecademy - Cheatsheets / Learn JavaScript](#)

htmlcheatsheet.com/js

[Websitesetup - Javascript-Cheat-Sheet.pdf](#)

javascript.info

Statements Reference

Date Reference

Regular expressions

Class Reference

Error Reference

`// var last = array[ array.length - 1]`

`typeof (array) // object`

`Array.isArray(array)`