



# JS CheatSheet

## Basics ▶

## Loops ↕

### Variables x

```
var a; // variable
var b = "init"; // string
var c = "Hi" + " " + "Joe"; // = "Hi Joe"
var d = 1 + 2 + "3"; // = "33"
var e = [2,3,5,8]; // array
var f = false; // boolean
var g = /()/; // RegEx
var h = function(){}; // function object
const PI = 3.14; // constant
var a = 1, b = 2, c = a + b; // one line
let z = 'zzz'; // block scope local variable
```

#### Strict mode

```
"use strict"; // Use strict mode to write secure code
x = 1; // Throws an error because variable is not declared
```

#### Values

```
false, true // boolean
18, 3.14, 0b10011, 0xF6, NaN // number
"flower", 'John' // string
undefined, null, Infinity // special
```

#### Operators

```
a = b + c - d; // addition, subtraction
a = b * (c / d); // multiplication, division
x = 100 % 48; // modulo. 100 / 48 remainder = 4
a++; b--; // postfix increment and decrement
```

#### Bitwise operators

```
& AND 5 & 1 (0101 & 0001) 1 (1)
| OR 5 | 1 (0101 | 0001) 5 (101)
~ NOT ~5 (~0101) 10 (1010)
^ XOR 5 ^ 1 (0101 ^ 0001) 4 (100)
<< left shift 5 << 1 (0101 << 1) 10 (1010)
>> right shift 5 >> 1 (0101 >> 1) 2 (10)
>>> zero fill right shift 5 >>> 1 (0101 >>> 1) 2 (10)
```

#### Arithmetic

```
a * (b + c) // grouping
person.age // member
person[age] // member
!(a == b) // logical not
a != b // not equal
typeof a // type (number, object, function...)
x << 2 x >> 3 // binary shifting
a = b // assignment
a == b // equals
a != b // unequal
a === b // strict equal
a !== b // strict unequal
a < b a > b // less and greater than
a <= b a >= b // less or equal, greater or equal
a += b // a = a + b (works with - * %...)
a && b // logical and
a || b // logical or
```

### Arrays ≡

```
var dogs = ["Bulldog", "Beagle", "Labrador"];
var dogs = new Array("Bulldog", "Beagle", "Labrador"); // declarati

alert(dogs[1]); // access value at index, first item bei
dogs[0] = "Bull Terrier"; // change the first item
```

```
for (var i = 0; i < dogs.length; i++) { // parsing with array.le
  console.log(dogs[i]);
}
```

#### Methods

```
dogs.toString(); // convert to string: result
dogs.join(" * "); // join: "Bulldog * Beagle *
dogs.pop(); // remove last element
dogs.push("Chihuahua"); // add new element to the en
dogs[dogs.length] = "Chihuahua"; // the same as push
dogs.shift(); // remove first element
dogs.unshift("Chihuahua"); // add new element to the be
delete dogs[0]; // change element to undefin
dogs.splice(2, 0, "Pug", "Boxer"); // add elements (where, how
var animals = dogs.concat(cats, birds); // join two arrays (dogs fol
dogs.slice(1,4); // elements from [1] to [4-1]
dogs.sort(); // sort string alphabetically
dogs.reverse(); // sort string in descending
x.sort(function(a, b){return a - b}); // numeric sort
x.sort(function(a, b){return b - a}); // numeric descending sort
```

#### On page script

```
<script type="text/javascript"> ...
</script>
<script src="filename.js"></script>
```

#### Second timeout

```
setInterval(function () {
  // ...
}, 1000);
```

```
function addNumbers(a, b) {
  return a + b;
}
addNumbers(1, 2);
```

#### element

```
document.getElementById("elementID").innerHTML = "Hello World";
```

```
log(a); // write to the browser console
document.write(a); // write to the HTML
alert(a); // output in an alert box
confirm("Really?"); // yes/no dialog, returns true/false
prompt("Your age?", "0"); // input dialog. Second argument
```

#### Strings

```
var line1 = "line 1";
var line2 = "line 2";
var line3 = "line 3";
```

### Strings

```
var abc = "abcdefghijklmnopqrstuvwxyz";
var esc = 'I don\'t know'; // \n new line
var len = abc.length; // string length
abc.indexOf("lmno"); // find substring, -1 if does not
abc.lastIndexOf("lmno"); // last occurrence
abc.slice(3, 6); // cuts out "def", negative values work too
abc.replace("abc", "123"); // find and replace, takes a function
abc.toUpperCase(); // convert to upper case
abc.toLowerCase(); // convert to lower case
abc.concat(" ", str2); // abc + " " + str2
abc.charAt(2); // character at index: "c"
abc[2]; // unsafe, abc[2] = "C" does work
abc.charCodeAt(2); // character code at index: 99
abc.split(","); // splitting a string on commas
abc.split(""); // splitting on characters
128.toString(16); // number to hex(16), octal(8), binary(2)
```

### Dates

```
Wed May 06 2020 04:03:03 GMT+0530 (India Standard Time)
var d = new Date();
```

```
1588717983194 milliseconds passed since 1970
```

```
Number(d)
```

```
17-06-23"; // date declaration
17"; // is set to Jan 01
17-06-23T12:00:00-09:45"; // date - time YYYY-MM-DDTHH:MM:SS
ne 23 2017"; // long date format
n 23 2017 07:45:00 GMT+0100 (Tokyo Time)"; // time zone
```

```
new Date();
tDay(); // getting the weekday
```

```
); // day as a number (1-31)
; // weekday as a number (0-6)
ear(); // four digit year (yyyy)
(); // hour (0-23)
seconds(); // milliseconds (0-999)
es(); // minutes (0-59)
(); // month (0-11)
ds(); // seconds (0-59)
); // milliseconds since 1970
```

#### rt of a date

```
new Date();
e(d.getDate() + 7); // adds a week to a date
```

```
); // day as a number (1-31)
ear(); // year (optionally month and day)
n 23; // hour (0-23)
```

#### For Loop

```
for (var i = 0; i < 10; i++) {
  document.write(i + " ");
}
var sum = 0;
for (var i = 0; i < a.length; i++) {
  sum += a[i];
} // parsing
html = "";
for (var i of custOrder) {
  html += "<li>" + i + "</li>";
}
```

#### While Loop

```
var i = 1;
while (i < 100) {
  i *= 2;
  document.write(i + " ");
}
```

#### Do While Loop

```
var i = 1;
do {
  i *= 2;
  document.write(i + " ");
} while (i < 100)
```

#### Break

```
for (var i = 0; i < 10; i++) {
  if (i == 5) { break; }
  document.write(i + " ");
}
```

#### Continue

```
for (var i = 0; i < 10; i++) {
  if (i == 5) { continue; }
  document.write(i + " ");
}
```

### Events

```
<button onclick="myFunction()">
  Click here
</button>
```

#### Mouse

```
onclick, oncontextmenu, ondblclick, onmousedown, onmousemove, onmouseover, onmouseout, onmouseup
```

#### Keyboard

```
onkeydown, onkeypress, onkeyup
```

#### Frame

```
onabort, onbeforeunload, onerror, onpagehide, onresize, onscroll, onload
```

```
change, onfocus, onblur, onselect, onsubmit
```

```
ondragend, ondragstart, ondragover, ondragleave
```

```
oncut, onpaste, oncopy, ondrop
```

```
oncanplay, oncanplaythrough, ondataavailable, onloadedmetadata, onloadeddata, onprogress, onratechange, onseeked, onstalled, ontimeupdate, onvolumechange
```

```
end, animationend, animationiteration, animationstart
```

```
eous, onabort, onblur, oncancel, onchange, onclick, oncontextmenu, ondblclick, ondrag, ondragend, ondragstart, ondragover, ondragleave, ondrop, onerror, onfocus, oninput, oninvalid, onkeydown, onkeypress, onkeyup, onload, onloadeddata, onloadedmetadata, onmouseenter, onmouseleave, onmousemove, onmouseout, onmouseover, onmouseup, onpaste, onreadystatechange, onscroll, onselect, onsubmit, onstorage, onunload, onwheel
```

### Regular Express

```
var a = str.search(/CheatSheet/);
```

#### Modifiers

