

JS Cheat Sheet

Hide comments

If - Else ↗

```
if ((age >= 10) && (age < 19)) {           // logical condition
  status = "Eligible.";                   // executed if condition is true
} else {
  status = "Not eligible.";              // else block is optional
}

Switch Statement
switch (new Date().getDay()) {           // input is current day
case 6:                                // if (day == 6)
  text = "Saturday";
  break;
case 0:                                // if (day == 0)
  text = "Sunday";
  break;
default:                               // else...
  text = "Whatever";
}
```

Ads ↗

Basics ➤

On page script

Data Types ↗

```
var age = 18;                         // number
var name = "Jane";                    // string
var name = {firstName:"Jane", last:"Doe"}; // object
var truth = false;                     // boolean
var sheets = ["HTML", "CSS", "JS"];    // array
var a; typeof a;                      // undefined
var a = null;                         // value null

Objects
var student = {                       // object name
  firstName: "Jane",                 // list of properties and values
  lastName: "Doe",
  age: 18,
  height: 179,
  fullName: function() {            // object function
    return this.firstName + " " + this.lastName;
  }
};
student.age = 19;                      // setting value
student[age]++;                        // incrementing
name = student.fullName();             // call object function
```

Numbers and Math ↗

```
var pi = 3.141;
pi.toFixed(8);                         // returns 3
```

Dates 📅

```
Mon Dec 08 2025 23:00:06 GMT+0100 (Central European Standard Time)
var d = new Date();
1765231206414 milliseconds passed since 1970

Number(d)

Date("1962-03-23");                  // date object
Date("2017-06-01");                  // date - to Jan 01
Date("2017-06-01T12:00:00-09:45");    // date - time YYYY-MM-DDTHH:MM:SSZ
Date("June 23 2017");                // long date format
Date("Jun 23 2017 07:45:00 GMT+0100 (Tokyo Time)"); // time zone
```

Get Times

```
var d = new Date();
a = d.getDay();                      // getting the weekday

getDay();                            // day as a number (1-31)
getDay();                            // weekday as a number (0-6)
getFullYear();                      // four digit year (yyyy)
getHours();                          // hour (0-23)
getMilliseconds();                  // milliseconds (0-999)
getMinutes();                        // minutes (0-59)
getMonth();                          // month (0-11)
getSeconds();                        // seconds (0-59)
getTime();                           // milliseconds since 1970
```

Setting a date

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

setDate();                           // day as a number (1-31)
setFullYear();                      // year (optionally month and day)
setHours();                          // hour (0-23)
setMilliseconds();                  // milliseconds (0-999)
setMinutes();                        // minutes (0-59)
setMonth();                          // month (0-11)
setSeconds();                        // seconds (0-59)
setTime();                           // milliseconds since 1970
```

Ads ↗

Global Functions ()

```
eval();                                // executes a string as if it was script code
String();                              // return string from number
(23).toString();                      // return string from number
Number("23");                          // return number from string
```

e.asp" e.php"

JSON ↗

```
var str = '{"names":[' +
  '{"first":"Makuna", "last":"Matata"},' +
  '{"first":"Jane", "last":"Doe"},' +
  '{"first":"Bob", "last":"Jordan"}]};'
obj = JSON.parse(str);
document.write(obj.names[1].first);      // access
```

Sent

```
var myObj = { "name": "Jane", "age": 18, "city": "Chicago" }; // create object
var myJSON = JSON.stringify(myObj);
window.location = "demo.php?x=" + myJSON; // send to php
```

Useful Links ↗

Loops ➤

For Loop

```
for (var i = 0; i < 10; i++) {
  document.write(i + ":" + i*3 + "<br />");
}
var sum = 0;
for (var i = 0; i < a.length; i++) {
  sum += a[i];
}
// parsing an array
html = "";
for (var i of customer) {
  html += "<li>" + i + "</li>";
}
```

While Loop

```
var i = 1;                                // initialize
while (i < 100) {                         // enters the cycle if statement is true
  i *= 2;                                  // increment to avoid infinite loop
  document.write(i + ",");                  // output
}
```

Do While Loop

```
var i = 1;                                // initialize
do {                                         // enters cycle at least once
  i *= 2;                                  // increment to avoid infinite loop
  document.write(i + ",");                  // output
} while (i < 100)                          // repeats cycle if statement is true at t
```

Break

for (var i = 0; i < 10; i++) {

if (i == 5) { break; } // stops and exits the cycle
document.write(i + ","); // last output number is 4
}

Continue

for (var i = 0; i < 10; i++) {

if (i == 5) { continue; } // skips the rest of the cycle
document.write(i + ","); // skips 5

Variables x

```
var a;                                     // variable
var b = "init";                            // string
var c = "Hi" + " " + "Joe";                // = "Hi Joe"
var d = [1, 2, 3];                          // array
var f = false;                             // boolean
var g = /(\w)/;                            // RegExp
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 v
```

Strict mode

```
"use strict"; // Use strict mode to write secure cod
x = 1; // Throws an error because variable is
```

Values

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

Strings ☰

```
var abc = "abcdefghijklmnopqrstuvwxyz";
var esc = "I don't \n know";             // \n new line
var len = abc.length;                    // string length
abc.indexOf("no");                      // finds first occurance, -1 if doesn't contain
abc.lastIndexOf("no");                  // last occurrence
abc.slice(3, 6);                        // cuts out "def", negative values count from behind
abc.replace("abc", "123");              // find and replace, takes regular expressions
abc.toUpperCase();                      // convert to upper case
abc.toLowerCase();                      // convert to lower case
abc.concat(*, str2);                   // abc + " " + str2
abc.charAt(2);                         // character at index: " "
abc.charCodeAt();                      // unicode code [1] "C" doesn't work
abc.charCodeAt(2);                      // character code at index: "C" -> 99
abc.split(",");                        // splitting a string on commas gives an array
abc.split("");                          // splitting on characters
128.toString(16);                      // number to hex(16), octal (8) or binary (2)
```

Anonymous

```
a (b c)                         // grouping
person[a]                         // member
person[age]                        // member
```

Events ⓘ

money

70

onmouseleave, onmousemove, onmouseover, onmouseout, onmouseup

ue

show, onpagehide, onresize, onscroll, onunload

lid, onreset, onsearch, onselect, onsubmit

jstart, ondrop

Arrays ≡

```
var dogs = ["Bulldog", "Beagle", "Labrador"];
var dogs = new Array("Bulldog", "Beagle", "Labrador");
d, onerror, onload, dogs[0], // access value at index, first item
illed, onsuspend, d, alert(dogs[1]); // change first item
for (var i = 0; i < dogs.length; i++) { // parsing
  console.log(dogs[i]);
}
```

Methods

```
popstate, onshow, dogs.toString();           // convert to string
dogs.join(" ");                            // join: "Bulldog Beagle Labrador"
dogs.pop();                                // remove last
dogs.push("Chihuahua");                   // add new ele
dogs.length = "Chihuahua";                 // the same as
dogs.shift();                             // remove first
dogs.unshift("Chihuahua");                // add new ele
delete dogs[0];                           // change elem
dogs.splice(2, 0, "Pug", "Boxer");         // add element
var animals = dogs.concat(cats,birds);     // join two arr
dogs.sort();                                // elements fr
dogs.reverse();                            // sort strings
```

Regular Expressions \n

var a = str.search(/CheatSheet/1);

Modifiers

i	perform case-insensitive matching
g	perform a global match
m	perform multiline matching

Patterns

\	Escape character
\d	find a digit
\s	find a whitespace character
\b	find match at beginning or end of a word

Promises p

```
function sum(a, b) {
  return new Promise(function (resolve, reject) {
    setTimeout(function () {
      if (typeof a != "number" || typeof b != "number") {
        return reject(new TypeError("Inputs must be numbers"));
      }
      resolve(a + b);
    }, 1000);
  });
}

var myPromise = sum(10, 5);
myPromise.then(function (result) {
  document.write(" 10 + 5: " + result);
});
```

return another promise

Errors ⚠

```
try {
  undefinedFunction();
} catch(err) {
  console.log(err.message);
}

Throw error
throw "My error message"; // throw a text
```

[JS cleaner](#)[Obfuscator](#)[Can I use?](#)[Node.js](#)[jQuery](#)[RegEx tester](#)

```
    } because of the error  
    // => Please provide two numbers to sum.  
    console.error(err);  
});  
  
States  
pending, fulfilled, rejected  
  
Properties  
Promise.length, Promise.prototype  
  
Methods  
Promise.all(iterable), Promise.race(iterable), Promise.reject(reason), Promise.resolve(value)
```