



JS CheatSheet

If - Else ↓

```
if ((age >= 14) && (age < 19)) { // logical condition
  status = "Eligible."; // executed if condition is true
} else { // else block is optional
  status = "Not eligible."; // executed if condition is false
}
```

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";
}
```

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 eq
a + b // a = a + b (works with - * %...)
a && b // logical and
a || b // logical or
```

Dates [31]

```
Fri Feb 28 2020 09:03:22 GMT+0100 (heure normale d'Europe centrale)
var d = new Date();
1582877002368 milliseconds passed since 1970
Number(d)
Date("2017-06-23"); // date declaration
Date("2017"); // is set to Jan 01
```

Basics ▶

On page script

```
<script type="text/javascript"> ...
</script>
```

include external JS file

```
<script src="filename.js"></script>
```

delay - 1 second timeout

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

functions

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

// ...
addNumbers(1, 2);
```

edit DOM element

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

Output

```
console.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 depending
prompt("Your age?", "0"); // input dialog. Second argument is the initial value
```

comments

```
* Multi line
  comment */
/ One line
```

Data Types ℵ

```
var age = 18; // number
var name = "Jane"; // string
var name = {first:"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:170,
  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(0); // returns 3
pi.toFixed(2); // returns 3.14 - for working with money
pi.toPrecision(2) // returns 3.1
pi.valueOf(); // returns number
Number(true); // converts to number
Number(new Date()) // number of milliseconds since 1970
parseInt("3 months"); // returns the first number: 3
parseFloat("3.5 days"); // returns 3.5
Number.MAX_VALUE // largest possible JS number
Number.MIN_VALUE // smallest possible JS number
Number.NEGATIVE_INFINITY// -Infinity
Number.POSITIVE_INFINITY// Infinity
```

Math.

```
var pi = Math.PI; // 3.141592653589793
Math.round(4.4); // = 4 - rounded
Math.round(4.5); // = 5
Math.pow(2,8); // = 256 - 2 to the power of 8
Math.sqrt(49); // = 7 - square root
Math.abs(-3.14); // = 3.14 - absolute, positive value
Math.ceil(3.14); // = 4 - rounded up
Math.floor(3.99); // = 3 - rounded down
Math.sin(0); // = 0 - sine
Math.cos(Math.PI); // OTHERS: tan, atan, asin, acos,
Math.min(0, 3, -2, 2); // = -2 - the lowest value
Math.max(0, 3, -2, 2); // = 3 - the highest value
Math.log(1); // = 0 natural logarithm
```

Loops ↻

For Loop

```
for (var i = 0; i < document.length; i++) {
  // ...
}
```

While Loop

```
var i = 1;
while (i < 10) {
  // ...
  i *= 2;
}
```

Do While Loop

```
var i = 1;
do {
  // ...
  i *= 2;
} while (i < 10);
```

Break

```
for (var i = 0; i < document.length; i++) {
  // ...
  break;
}
```

Continue

```
for (var i = 0; i < document.length; i++) {
  // ...
  continue;
}
```

String

```
var abc = "a b c";
var esc = 'I am a string';
var len = abc.length;
abc.indexOf("b"); // returns 1
abc.lastIndexOf("b"); // returns 2
abc.slice(3, 5); // returns "c"
abc.replace("a", "x"); // returns "x b c"
abc.toUpperCase(); // returns "A B C"
abc.toLowerCase(); // returns "a b c"
abc.concat("d"); // returns "a b c d"
abc.charAt(2); // returns "c"
abc.charCodeAt(2); // returns 99
abc.split(","); // returns ["a", "b", "c"]
abc.split(" ").length; // returns 3
128.toString(); // returns "128"
```

Events

```
button onclick
Click her
/button>
mouse
onclick, onmouseover, onmouseout
keyboard
keydown, onkeypress
name
onabort, onblur, onfocus, onresize, onscroll
form
onblur, onchange, oninput, oninvalid, onreset, onsubmit
drag
ondrag, ondragend, ondragstart
clipboard
oncopy, oncut, onpaste
media
onabort, oncanplay, onended, onloadeddata, onloadedmetadata, onpause, onplay, onplaying, onprogress, onratechange, onreadystatechange, onseeked, onseeking, onstalled, ontimeupdate, onvolumechange, onwaiting
```

animation
animationend

miscellaneous
transitionend,
onstorage, on
ontouchstart

Global

```
eval();
String(23);
(23).toString()
Number("23")
decodeURI(en
encodeURIComponent
decodeURIComponent
encodeURIComponent
isFinite();
isNaN();
parseFloat()
parseInt();
```

Regula

```
var a = str.
```

Modifiers

Patterns

d
s
b
n+
n*
n?
A

JXXX

Promis

```

Function sum
return Promise.resolve(
  setTime(1000, () => {
    return 1000;
  })
);
}

var myPromise = myPromise.resolve(
  document.write('sum is 1000')
).then(function() {
}).catch(function() {
  console.error('error');
});

```

States
pending, fulfill

Properties

Methods

```
Promise.all(items)
  .then(Promise.resolve)
```