JS CheatSheet

Hide comments

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

```
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 declar
```

Values

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

Operators

```
a = b + c - d;      // addition, substraction
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      // minary 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

Basics

On page script

```
e="text/javascript">  ...
```

nal JS file

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

ond timeout

```
function () {
```

```
dNumbers(a, b) {
; ;

ers(1, 2);
```

ment

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

Output

```
(a);            // write to the browser console
ite(a);         // write to the HTML
                // output in an alert box
ally?");        // yes/no dialog, returns true/false
age?","0");     // input dialog. Second argument is t
```

```
ne
```

Data Types

```
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
```

Loops

For Loop

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

While Loop

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

Do While Loop

```
var i = 1;
do {
i *= 2;                 // i
document.write(i + ", ");   // o
} 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 + ", ");
```

Strings

```
var abc = "abcdefghijklmnopqrstu
var esc = 'I don\'t \n know';
var len = abc.length;
abc.indexOf("lmno");
abc.lastIndexOf("lmno");
abc.slice(3, 6);
abc.replace("abc","123");
abc.toUpperCase();
abc.toLowerCase();
abc.concat(" ", str2);
abc.charAt(2);
abc[2];
abc.charCodeAt(2);
abc.split(",");
abc.split("");
128.toString(16);
```

Events

```
lick="myFunction();">
```

ntextmenu, ondblclick, o
ve, onmouseover, onmou

onkeypress, onkeyup

eforeunload, onerror, onl
scroll, onunload

ange, onfocus, onfocusin
select, onsubmit

agend, ondragenter, ond

Fri Jul 16 2021 11:36:18 GMT+0530 (India Standard Time)
```
var d = new Date();
```
1626415578963 miliseconds passed since 1970
```
Number(d)
```
```
Date("2017-06-23");                 // date declaration
Date("2017");                       // is set to Jan 01
Date("2017-06-23T12:00:00-09:45");  // 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

getDate();          // 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 part of 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)
```

Errors ⚠

```
try {                           // block of code to try
undefinedFunction();
}
catch(err) {                    // block to handle errors
console.log(err.message);
}
```

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

Input validation
```
var x = document.getElementById("mynum").value; // get input value
try {
if(x == "")  throw "empty";                 // error cases
if(isNaN(x)) throw "not a number";
x = Number(x);
if(x > 10)   throw "too high";
}
catch(err) {                                    // if there's an
document.write("Input is " + err);          // output error
console.error(err);                         // write the error in
}
finally {
document.write("</br />Done");              // executed regardles
}
```

Error name values

RangeError	*A number is "out of range"*
ReferenceError	*An illegal reference has occurred*
SyntaxError	*A syntax error has occurred*
TypeError	*A type error has occurred*
URIError	*An encodeURI() error has occurred*

Useful Links ↲

JS cleaner **Obfuscator** **Can I use?** **Node.js** **jQuery** **RegEx tester**

```
.14);        // = 3.14 - absolute, positive value
.14);        // = 4 - rounded up
3.99);       // = 3 - rounded down
;            // = 0 - sine
th.PI);      // OTHERS: tan,atan,asin,acos,
 3, -2, 2);  // = -2 - the lowest value
 3, -2, 2);  // = 3 - the highest value
;            // = 0 natural logarithm
;            // = 2.7182pow(E,x)
();          // random number between 0 and 1
Math.random() * 5) + 1;  // random integer, from 1 to 5
```

e Math.PI:

2, SQRT1_2, LN2, LN10, LOG2E, Log10E

Arrays ≡

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

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

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

Methods
```
dogs.toString();                    // convert to string: res
dogs.join(" * ");                   // join: "Bulldog * Beagl
dogs.pop();                         // remove last element
dogs.push("Chihuahua");             // add new element to the
dogs[dogs.length] = "Chihuahua";    // the same as push
dogs.shift();                       // remove first element
dogs.unshift("Chihuahua");          // add new element to the
delete dogs[0];                     // change element to undefined (not
dogs.splice(2, 0, "Pug", "Boxer");  // add elements (where, how many to
var animals = dogs.concat(cats,birds);  // join two arrays (dogs followed by
dogs.slice(1,4);                    // elements from [1] to [4-1]
dogs.sort();                        // sort string alphabetic
e();                                // sort string in descend
tion(a, b){return a - b});          // numeric sort
tion(a, b){return b - a});          // numeric descending sor
[0];                                // first item in sorted a
tion(a, b){return 0.5 - Math.random()});     // random
```

Within, every, fill, filter, find, findIndex, forEach, indexOf, isArra
map, pop, push, reduce, reduceRight, reverse, shift, slice, son
g, unshift, valueOf

JSON j

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

Send
```
var myObj = { "name":"Jane", "age":18, "city":"Chicago" };  // cr
var myJSON = JSON.stringify(myObj);                         // st
window.location = "demo.php?x=" + myJSON;                   // se
```

Storing and retrieving
```
myObj = { "name":"Jane", "age":18, "city":"Chicago" };
myJSON = JSON.stringify(myObj);                 // storing data
localStorage.setItem("testJSON", myJSON);
text = localStorage.getItem("testJSON");        // retrieving dat
obj = JSON.parse(text);
document.write(obj.name);
```

Global Functions ()

```
eval();                     // ex
String(23);                 // re
(23).toString();            // re
Number("23");               // re
decodeURI(enc);             // de
encodeURI(uri);             // en
decodeURIComponent(enc);    // de
encodeURIComponent(uri);    // en
isFinite();                 // is
isNaN();                    // is
parseFloat();               // re
parseInt();                 // pa
```

Regular Expression

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

Modifiers

i	*perform case-in*
g	*perform a globa*
m	*perform multilin*

Patterns

****	*Escape charact*
\d	*find a digit*
\s	*find a whitespa*
\b	*find match at be*
n+	*contains at leas*
n*	*contains zero o*
n?	*contains zero o*
^	*Start of string*
$	*End of string*
[]	*find the Unicod*
.	*Any single char*
a|b	*a or b*
()	*Group section*
[0-3]	*In range (a, b o*
[^0-3]	*any of the digits*

Promises Þ

```
function sum (a, b) {
return Promise(function (resolve,
 setTimeout(function () {
   if (typeof a !== "number" || t
        return reject(new TypeE
   }
   resolve(a + b);
 }, 1000);
});
}
var myPromise = sum(10, 5);
myPromise.then(function (result)
document.write(" 10 + 5: ", resu
return sum(null, "foo");
}).then(function () {
}).catch(function (err) {
console.error(err);
});
```

States

pending, fulfilled, rejected

Properties

Promise.length, Promise.prototype

Methods

Promise.all(iterable), Promise.race(ite
Promise.resolve(value)