JS CheatSheet

Hide comments

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, 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
```

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.length
console.log(dogs[i]);
}
```

Methods

```
dogs.toString();                    // convert to string: results "Bulldog,Beagle,La
dogs.join(" * ");                   // join: "Bulldog * Beagle * Labrador"
dogs.pop();                         // remove last element
dogs.push("Chihuahua");             // add new element to the end
dogs[dogs.length] = "Chihuahua";    // the same as push
dogs.shift();                       // remove first element
dogs.unshift("Chihuahua");          // add new element to the beginning
delete dogs[0];                     // change element to undefined (not recommended)
dogs.splice(2, 0, "Pug", "Boxer");  // add elements (where, how many to remove, elem
var animals = dogs.concat(cats,birds);  // join two arrays (dogs followed by cats and bi
dogs.slice(1,4);                    // elements from [1] to [4-1]
dogs.sort();                        // sort string alphabetically
dogs.reverse();                     // sort string in descending order
x.sort(function(a, b){return a - b});  // numeric sort
x.sort(function(a, b){return b - a});  // numeric descending sort
highest = x[0];                     // first item in sorted array is the lowest (or
x.sort(function(a, b){return 0.5 - Math.random()});     // random order sort
```

concat, copyWithin, every, fill, filter, find, findIndex, forEach, indexOf, isArray, join, lastIndexOf, map, pop, push, reduce, reduceRight, reverse, shift, slice, some, sort, splice, toString, 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" };  // create object
var myJSON = JSON.stringify(myObj);                         // stringify
```

Basics ➤

On page script

```
ript">  ...

></script>

) {

"elementID").innerHTML = "Hello World!"

    // write to the browser console
    // write to the HTML
    // output in an alert box
    // yes/no dialog, returns true/fals
    // input dialog. Second argument is
```

Strings ⊗

```
var abc = "abcdefghijklmnopqrstuvwxyz";
var esc = 'I don\'t \n know';   // \n new line
var len = abc.length;           // string length
abc.indexOf("lmno");            // find substring, -1 if doesn't co
abc.lastIndexOf("lmno");        // last occurance
abc.slice(3, 6);                // cuts out "def", negative val
abc.replace("abc","123");       // find and replace, takes regu
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" doesn't
abc.charCodeAt(2);              // character code at index: "c"
abc.split(",");                 // splitting a string on commas
abc.split("");                  // splitting on characters
128.toString(16);               // number to hex(16), octal (8)
```

Dates 31

Tue Nov 09 2021 16:43:12 GMT-0300 (Horário Padrão de Brasília)

```
var d = new Date();
```

1636486992680 miliseconds passed since 1970

```
                    // date declaration
                    // is set to Jan 01
-09:45");           // date - time YYYY-MM-DDTHH:MM:SSZ
                    // long date format
00 GMT+0100 (Tokyo Time)"); // time zone

etting the weekday

y as a number (1-31)
eekday as a number (0-6)
our digit year (yyyy)
our (0-23)
illiseconds (0-999)
inutes (0-59)
onth (0-11)
econds (0-59)
illiseconds since 1970

7); // adds a week to a date

y as a number (1-31)
ear (optionally month and day)
our (0-23)
illiseconds (0-999)
inutes (0-59)
onth (0-11)
setSeconds();       // seconds (0-59)
setTime();          // milliseconds since 1970)
```

Errors ⚠

```
                    // block of code to try

                    // block to handle errors

;   // throw a text
```

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 custOrder) {
html += "<li>" + i + "</li>";
}
```

While Loop

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

Do While Loop

```
var i = 1;                      // initialize
do {                            // enters cycle at least onc
i *= 2;                     // increment to avoid infinite l
document.write(i + ", ");   // output
} while (i < 100)               // repeats cycle if statemer
```

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 cyc
                                // skips 5
```

Events 🕐

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

Mouse

onclick, oncontextmenu, ondblclick, onmousedown, onmouseenter, onmouseover, onmouseout, onmouseup

Keyboard

onkeydown, onkeypress, onkeyup

Frame

onabort, onbeforeunload, onerror, onhashchange, onload, onpagesh
onscroll, onunload

onfocusin, onfocusout, oninput, oninvalic

enter, ondragleave, ondragover, ondrags

laythrough, ondurationchange, onended,
tart, onpause, onplay, onplaying, onprog
pend, ontimeupdate, onvolumechange,

ation, animationstart

nmousewheel, ononline, onoffline, onpo
cancel, ontouchend, ontouchmove, ontou

Regular Expressions \n

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

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*

atch at beginning or end of a word
ns at least one n
ns zero or more occurrences of n
ns zero or one occurrences of n
f string
f string
e Unicode character
ngle character

section
ge (a, b or c)
the digits between the brackets

Useful Links

```
window.location = "demo.php?x=" + myJSON;                // send to php
```

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 data
obj = JSON.parse(text);
document.write(obj.name);
```

```
entById("mynum").value; // get input va
/";                          // error cases
a number";
nigh";
                                 // if there's an error
document.write("Input is " + err);          // output error
console.error(err);                         // write the error in console
}
finally {
document.write("</br />Done");              // executed regardless of the try / catch re
}
```

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*

JS cleaner Obfuscator Can I use? jQuery RegEx tester

r more, ungreedy
r more of a
r more, ungreedy
y 2 of a
ore of a
5 of a
of a
of a, ungreedy
unctuation symbol
pace character
or tab