JS CheatSheet

Hide comments

Ads

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";
}
```

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
```

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 conta
abc.lastIndexOf("lmno");        // last occurance
abc.slice(3, 6);                // cuts out "def", negative values cou
abc.replace("abc","123");       // find and replace, takes regular exp
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 work
```

Basics

On page script

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

nal JS file

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

ond timeout

```
(function () {
```

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

bers(1, 2);
```

ment

```
document.getElementById("elementID").innerHTML = "Hello Wor
```

Output

```
console.log(a);             // write to the browser console
rite(a);                    // write to the HTML
                            // output in an alert box
eally?");                   // yes/no dialog, returns true/
ur age?","0");              // input dialog. Second argumen
```

```
ine

e
```

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
 3.14;                      // constant
 b = 2, c = a + b;          // one line
zz';                        // block scope local variab
```

```
t";   // Use strict mode to write secure code
      // Throws an error because variable is not
```

```
e                           // boolean
0b10011, 0xF6, NaN          // number
'John'                      // string
 null , Infinity            // special
```

```
- d;      // addition, substraction
 / d);    // multiplication, division
48;       // modulo. 100 / 48 remainder = 4
          // postfix increment and decrement
```

ators

&	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)
	hift	5 >> 1 (0101 >> 1)	2 (10)
	l right shift	5 >>> 1 (0101 >>> 1)	2 (10)

```
)          // grouping
           // member
]          // member
           // logical not
           // not equal
           // type (number, object, function...)
>> 3       // minary shifting
           // assignment
```

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;
sum + = a[i];
}                   // parsing an
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 + ", ");
```

Ads

Events

```
abc.charCodeAt(2);              // character code at index: "c" -> 99
abc.split(",");                 // splitting a string on commas gives
abc.split("");                  // splitting on characters
128.toString(16);               // number to hex(16), octal (8) or bir
```

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 197(
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
Math.exp(1);            // = 2.7182pow(E,x)
Math.random();          // random number between 0 and 1
Math.floor(Math.random() * 5) + 1;  // random integer, from
```

Constants like Math.PI:

E, PI, SQRT2, SQRT1_2, LN2, LN10, LOG2E, Log10E

Global Functions ()

```
eval();                     // executes a string as if it was script c
String(23);                 // return string from number
(23).toString();            // return string from number
Number("23");               // return number from string
decodeURI(enc);             // decode URI. Result: "my page.asp"
encodeURI(uri);             // encode URI. Result: "my%page.asp"
decodeURIComponent(enc);    // decode a URI component
encodeURIComponent(uri);    // encode a URI component
isFinite();                 // is variable a finite, legal r
isNaN();                    // is variable an illegal numbe
parseFloat();               // returns floating point numbe
parseInt();                 // parses a string and returns a
```

Ads 📣

```
        // equals
        // unequal
        // strict equal
        // 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 - * %...)
        // logical and
        // logical or
```

Dates 📅

Thu Feb 03 2022 21:06:24 GMT-0500 (Eastern Standard Time)

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

1643940384565 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:SS
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)
();                 // minutes (0-59)
;                   // month (0-11)
();                 // seconds (0-59)
                    // milliseconds since 1970)
```

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
                    find a digit
                    find a whitespace character
                    find match at beginning or end of a word
                    contains at least one n
                    contains zero or more occurrences of n
                    contains zero or one occurrences of n
                    Start of string
                    End of string
                    find the Unicode character
                    Any single character
                    a or b
                    Group section
                    In range (a, b or c)
                    any of the digits between the brackets
```

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
try {
if(x == "")  throw "empty";                 // error cases
```

Arrays ≡

```
var dogs = ["Bulldog", "Beagle
var dogs = new Array("Bulldog"

alert(dogs[1]);             //
dogs[0] = "Bull Terier";    //

for (var i = 0; i < dogs.lengt
console.log(dogs[i]);
}
```

```
ing();
" * ");
;
"Chihuahua");
length] = "Chihuahua
();
ft("Chihuahua");
s[0];
e(2, 0, "Pug", "Boxe
s = dogs.concat(cats
(1,4);
);
se();
ction(a, b){return a
ction(a, b){return b
x[0];
ction(a, b){return
```

Within, every, fill, filter, fi
map, pop, push, reduce,
g, unshift, valueOf

JSON j

```
var str = '{"names":[' +
'{"first":"Hakuna","lastN":"Ma
'{"first":"Jane","lastN":"Doe"
'{"first":"Air","last":"Jordan
obj = JSON.parse(str);
document.write(obj.names[1].fi
```

Send

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

Mouse

onclick, oncontextmenu, ondblclick, o
onmousemove, onmouseover, onmou

Keyboard

onkeydown, onkeypress, onkeyup

Frame

eforeunload, onerror, on
scroll, onunload

nge, onfocus, onfocusir
select, onsubmit

agend, ondragenter, onc

t, onpaste

anplay, oncanplaythroug
a, onloadedmetadata, or
nratechange, onseeked
nge, onwaiting

d, animationiteration, ani

s

, onmessage, onmousev
ntoggle, onwheel, ontou

```
if(isNaN(x)) throw "not a number";
x = Number(x);
if(x > 10)   throw "too high";
}
catch(err) {                                    // if there
document.write("Input is " + err);              // output error
console.error(err);                             // write the er
}
finally {
document.write("</br />Done");                  // executed reg
}
```

Error name values

rror	*A number is "out of range"*
	An illegal reference has occurred
	A syntax error has occurred
	A type error has occurred
	An encodeURI() error has occurred

```
var myObj = { "name":"Jane", "
var myJSON = JSON.stringify(my
window.location = "demo.php?x=
```

Storing and retrieving

```
myObj = { "name":"Jane", "age"
myJSON = JSON.stringify(myObj)
localStorage.setItem("testJSON
text = localStorage.getItem("t
obj = JSON.parse(text);
document.write(obj.name);
```

Promises Þ

```
function sum (a, b) {
return Promise(function (resolve, reject) {
 setTimeout(function () {                                  // ser
   if (typeof a !== "number" || typeof b !== "number") {   // tes
         return reject(new TypeError("Inputs must be numbers"));
   }
   resolve(a + b);
 }, 1000);
});
}
var myPromise = sum(10, 5);
myPromsise.then(function (result) {
document.write(" 10 + 5: ", result);
return sum(null, "foo");                 // Invalid data and return anoth
}).then(function () {                      // Won't be called because of
}).catch(function (err) {                  // The catch handler is callec
console.error(err);                      // => Please provide two numbers
});
```

States

pending, fulfilled, rejected

Properties

Promise.length, Promise.prototype

Methods

Promise.all(iterable), Promise.race(iterable), Promise.reject(reason), Promise.resolve(value)

Useful Links ↵

JS cleaner Obfuscat

jQuery