JavaScript Cheat Sheet

Zhihao Li

November 20, 2025

1 Basics

On page script

Embeding the `JavaScript` code in the `html` file just as follows. That ensures the browser can load the program script and run it.

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

Include external JS file

If more codes cann't be directly placed in the `<script></script>`, we can import the external JS file.

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

Delay - 1 second timeout

This is a delayed function. When the time ends (1000 ms), it will execute the function which is empty in the example.

```
setTimeout(function () {
  // something to do
}, 1000);
```

Functions

```
function addNumbers(a, b) {
  return a + b; ;
}
x = addNumbers(1, 2);
```

Edit DOM element

Code for modifying the DOM (Document Object Model). `JavaScript` code will be execute to dynamicly change the HTML elements.

```
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 on
```

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

Comments

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

2 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 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 the end
```

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
}
```

3 Branch

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";
}
```

4 Variables

Defination

1. `var` defines the variable in the function scope and become global variable if it's defined in the outside of function. It can be used with the value of `undefined` before defination and be alse defined repeatly.
2. `let` defines the variable in the block scope, such as `for`, `if while` or `{}`. It cann't be used before defination and not be defined repreatly.
3. `var g = /()/;` defines a regular expression using the pair symbols of `/ /` and `()` means a capturing group.

```
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

Directly writing the code of `"use strict";` in the first line of `JavaScript`.

```
"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
```

5 Data Types

Basics

```
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
```

6 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 contain
abc.lastIndexOf("lmno");        // last occurance
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: "c"
abc[2];                         // unsafe, abc[2] = "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)
```

7 Dates

Objects

```
Wed Jun 11 2025 18:31:19 GMT+0800 (中国标准时间)
var d = new Date();
1749637879070 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)
```

8 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,Labrador"
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, element list)
var animals = dogs.concat(cats,birds);  // join two arrays (dogs followed by
cats and birds)
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 highest)
value
x.sort(function(a, b){return 0.5 - Math.random()}); // random order sort
```

9 References

1. JS Cheat Sheet: https://htmlboxedsheet.com/js/
2. HTML Cheat Sheet: https://htmlboxedsheet.com/