Table of contents

Docs

docs MDN

Basic syntax

Code Conventions

```
// variable names
var yearMonthDay = moment().format("YYYY/MM/DD");

// constants
const FIRST_US_PRESIDENT = "George Washington"

// not a magic constant literals
const MINUTES_IN_A_YEAR = 52600;
for(let i = 0; i < MINUTES_IN_A_YEAR; i++){
    runJob();
}

// not use unneeded context
var Car = {
    carMake: "Honda",
    carModel: "Accord",
    carColor: "Blue"
} // car is redundunt in this case
```

Console

to print something for debug

```
console.log('hello world')
```

const let var

Const

For constants. Final variable

```
const Pi = 3.14

Pi = 4  // ERROR
```

let

for block level variables

```
console.log(i)     // ERROR
```

```
for(let i = 0; i< 3, i++){
    console.log(i)     // 1 2
}

console.log(i)     // ERROR
```

var

For global variables

```
console.log(i)     // undefined. Compiler knows that this variable exists.

for(var i = 0; i< 3, i++){
    console.log(i)     // 1 2
}

console.log(i)     // 3
```

null vs undefined

```
var test
console.log(test) // undefined

test = null
console.log(test) // null

console.log(typeof null) // Object
console.log(typeof undefined) // undefined


console.log(null === undefined) // false
console.log(null == undefined) // true
console.log(null === null) // true
console.log(null == null) // true


console.log(!null) // true
console.log(!!null) // false

console.log(1 + null) // 1
console.log(1 + undefined) // NaN
```

Comparison operators

as usual

> < <= >= != ==

if-else

```
var hello = true;

if(hello){
    console.log("Hello World");
} else{
    console.log("Bye");
}
```

```
var age = 18;

if(age >= 18){
    console.log("You are an adult");
} else if (age < 2) {
    console.log("You are a baby");
} else if (age < 18) {
    console.log("You are a child");
}

if(age == 18){
    console.log("You are eighteen");
}


if(age != 18){
    console.log("You are NOT eighteen");
}
```

Strings

Basic string methods

```
var first = "hello"
first.charAt(1) // e
```

```
var second = "world"
// return Unicode code
second.charCodeAt(2) // 114
```

```
String.fromCharCode(114) // r
```

```
var first = "hello"
var second = "world"
first.concat(second) // helloworld
```

```
var first = "hello"
first.endWith("lo") // true
```

```
var first = "hello"
first.startWith("he") // true
```

```
var first = "hello"
first.includes("ell") // true
```

```
var first = "hello"
first.indexOf("l") // 2
```

```
var first = "hello end world end"
first.lastIndexOf("end") // 16
```

```
var first = "hello"
first.slice(2, 4) // ll
```

```
var first = "hello"
first.substr(3, 2) // lo
```

```
var first = "hello"
first.subsstring(2, 4) // ll
```

```
var first = "hello world"
first.split(" ") // ["hello", "world"]
```

```
var first = "HEllo"
first.toLowerCase(first) // hello
```

```
var first = "hello"
first.toUpperCase(first) // HELLO
```

```
var first = "   hello  "
first.trim(first) // "hello"
```

regex with string

```
var first = "hello end world end"
first.match("/end/g") // ["end", "end"]
```

```
var first = "hello end world end"
first.replace("/end/g", "END") // "hello END world END"
```

```
var first = "hello end world end"
first.search("end") // 6
```

repeat

```
var first = "k"
first.repeat(3) // kkk
```

Arrays

Array as a Stack

Mutable operations

```
var arr = ["a", "b", "c"]

arr.push("d")
console.log(arr) // ["a", "b", "c", "d"]

console.log(arr.pop())   // "d"
console.log(arr) // ["a", "b", "c"]
```

Concat arrays

```
var arr2 = ["g", "q"]
console.log(arr.concat(arr2)) // ["a", "b", "c", "g",  "q" ]

console.log(arr2) // ["g",  "q" ]
```

Join array and string

```
var arr = ["a", "b", "c"]
console..log(arr.join("")) // abc
console..log(arr.join("!")) // a!b!c!
```

Reverse array

Mutable operation

```
var arr = ["a", "b", "c"]
console..log(arr.reverse()) // ["c", "b", "a"]

console..log(arr) // ["c", "b", "a"]
```

Shift. Return and delete the first element of array

Mutable operation

```
var arr = ["a", "b", "c"]
console..log(arr.shift()) // a

console..log(arr) // ["b", "c"]
```

Unshift. Add to the beginiing of array

Mutable operation

```
var arr = ["a", "b", "c"]
console..log(arr.unshift("e")) // 4

console..log(arr) // ["e", "a", "b", "c"]
```

Slice. Get sub array

```
var arr = ["a", "b", "c"]
console..log(arr.slice(1, 2)) // ["b"]
```

Sort of array

Mutable operation

```
var arr = ["c", "b", "a"]
console..log(arr.sort()) // ["a", "b", "c"]
```

Splice. Remove items and paste value

```
var arr = ["a", "b", "c", "d", "e"]
console..log(arr.splice(2, 2, "JS")) // ["c", "d"]
console..log(arr) // ["a", "b", "JS", "e"]
```

Modules

```
// After ES6

// libs/module.js

class ShoppingDataType{
    constructor(){
        // private properties
        this.shoppingList = ["coffee", "chicken", "pizza"]
    }

// public methods
    getShoppingList(){
        return this.shoppingList.join(", ")
    }
```

```
    addItem(item){
        this.shoppingList.push(item)
    }
}

export default ShoppingDataType;

// main.js

import ShoppingDataType from 'libs/module';

var shopping = new ShoppingDataType;
console.log(shopping.getShoppingList());
```

Examples

Change internal in html tag

```
<html>
    <body>
        <h1>My First JavaScript</h1>
        <p>JavaScript can change the content of an HTML element:</p>
        <button type="button" onclick="myFunction()">Click Me!</button>
        <p id="demo">This is a demonstration.</p>
        <script>
            function myFunction() {
                document.getElementById("demo").innerHTML = "Hello
JavaScript!";
            }
        </script>
    </body>
</html>
```

Change css

```
<!DOCTYPE html>
<html>
    <body>
        <h1>My First JavaScript</h1>
        <p id="demo">JavaScript can change the style of an HTML element.
</p>
        <script>
            function myFunction() {
                document.getElementById("demo").style.fontSize = "25px";
                document.getElementById("demo").style.color = "red";
```

```
                document.getElementById("demo").style.backgroundColor =
"yellow";
            }
        </script>
        <button type="button" onclick="myFunction()">Click Me!</button>
    </body>
</html>
```

Change tag attribute

```
<!DOCTYPE html>
<html>
    <body>
    <script>
        function light(sw) {
            var pic;
            if (sw == 0) {
                pic = "pic_bulboff.gif"
            } else {
                pic = "pic_bulbon.gif"
            }
            document.getElementById('myImage').src = pic;
        }
    </script>

    <img id="myImage" src="pic_bulboff.gif" width="100" height="180">
    <p>
        <button type="button" onclick="light(1)">Light On</button>
        <button type="button" onclick="light(0)">Light Off</button>
    </p>
    </body>
</html>
```

If JS disabled

```
<script>
document.getElementById("demo").innerHTML = "Hello JavaScript!";
</script>

<noscript>Sorry, your browser does not support JavaScript!</noscript> -->
```