

# JavaScript 1

## What is JavaScript

- Lightweight interpreted (JIT-compiled) language.
  - V8, SpiderMonkey, JavaScriptCore
- Runs on a browser\*
- Used to create dynamic websites (Along with HTML & CSS).
  - DOM
- Object oriented (kinda)!

## How

### External file

```
<script src="location/of/file.js"></script>
```

### <script> tags

```
1 <script>
2 var hello = "Hello";
3 var world = "World";
4
5 console.log(hello + " " + world + "!");
6 </script>
```

### Location

```
1 <html>
2   <head>
3     <script>
4       function newDoc() {
5         window.location.assign("https://www.w3schools.com")
6       }
7     </script>
8     <script src="location/of/file1.js"></script>
9   </head>
10  <body>
11    <p>Other page content</p>
12
13    <!-- Why? -->
```

```
14 <script src="location/of/file2.js"></script>
15 <script src="http://example.com/of/file.js"></script>
16
17 </body>
18 </html>
```

## Debugging

- Browser Tools
  - 'debugger;' keyword.
- Browser Console

## Packaging

<https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js>

vs

<https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.js>

<https://jscompress.com/>

## Basics

```
1 // This is a comment.
2
3 /*
4 Everything in between is a comment.
5 */
6
7 // Variables.
8 var myUndefined;
9 var myNull = null;
10 var myString = "Bob";
11 var myNumber = 10;
12 var myArray = [1, 2, 3, 'Four'];
13 var myBoolean = true;
14 var myObject = {}; // Or anything else.
15 var myDate = new Date();
```

```
1 // Type conversion
2 var myInt = parseInt('1');
3 var myFloat = parseFloat('1.3');
4 var myString = String(1);
5 var myDate = new Date(Date.parse('2017-08-01 00:00:00'));
```

## Arrays

```
1 // Define array.
2 var myArray = [1, 2, 3];
3
4 // String array.
5 var myStringArray = ["one", "two", "three"];
6
7 // Length of an array.
8 var mycount = myArray.length;
9
10 // Sorting an array.
11 myArray.sort();
12 myArray.sort(function(a, b) {return a - b;});
```

## Objects

```
1 // Define an object.
2 var myObj = { "name": "John", "age": 50, "car": { "name": "Toyota", "price": null } };
3
4 // Read or write an object.
5 var name = myObj.name;
6 myObj.name = "Rambo";
7
8 // Read or write an object.
9 var age = myObj['age'];
10 myObj['age'] += 1;
11
12 // Delete an object.
```

```
13 delete myObj.car.price;
14
15 // Check if member exists in an object.
16 if('car' in myObj) {
17     // Code here...
18 }
19
20 // Loop through an object.
21 for(x in myObj) {
22     console.log(x, ":", myObj[x]);
23 }
24
25 // Object methods.
26 var myObject = {
27     firstName:"John",
28     lastName: "Doe",
29     fullName: function () {
30         return this.firstName + " " + this.lastName;
31     }
32 }
33 myObject.fullName();
34
35 // Object constructors.
36 // Constructor.
37 function myFunction(arg1, arg2) {
38     this.firstName = arg1;
39     this.lastName = arg2;
40 }
41
42 // This creates a new object
43 var x = new myFunction("John", "Doe");
44 x.firstName;
```

## Operators

[https://www.w3schools.com/js/js\\_operators.asp](https://www.w3schools.com/js/js_operators.asp)

# Math

```
Math.
```

## Control Structures

### Conditions

```
1  if (condition1) {
2      // block of code to be executed if condition1 is true
3  } else if (condition2) {
4      // block of code to be executed if the condition1 is false and condition
5  2 is true
6  } else {
7      // block of code to be executed if the condition1 is false and condition
8  2 is false
9  }
```

### switch

```
1  switch(expression) {
2      case n:
3          // code block
4          break;
5      case n:
6          // code block
7          break;
8      default:
9          // code block
10 }
```

### Loop for

```
1  for (statement 1; statement 2; statement 3) {
2      // code block to be executed
3  }
```

### While Loop

```
1  for (statement 1; statement 2; statement 3) {
2      // code block to be executed
3  }
```

## Regular Expressions

```
1 var str = "This is JavaScript!";
2
3 // /pattern/modifiers
4 str.search(/java/i);
5
6 // RegExe object
7 var regEx = new RegExp("java", "i");
8 str.search(regEx);
```

## Functions

```
1 function myFunction(x, y) {
2     return x + y;
3 }
4
5 var z = myFunction(3, 2);
6
7 // NOTE: Objects are passed by reference.
```

## Browser Object Model (BOM)

### window

- Represents the browser's window.
- All global objects, functions, variables are members of the window.
- Objects
  - document
  - screen
  - location
  - history
  - navigation
  - alert
  - timing
  - cookies

```
1 window.document.getElementById('id');
2 // same as
3 document.getElementById('id')
```

## Timing

- Set events to run after a certain amount of time.
- Set recurring events.

```
1 // Create a timeout.
2 var timeout = setTimeout(function, milliseconds);
3
4 // Destroy a timeout before it runs.
5 window.clearTimeout(timeout);
6
7 // Create an interval.
8 var interval = setInterval(function, milliseconds);
9
10 // Destroy an interval.
11 clearInterval(interval);
```

## JSON

- JavaScript Object Notation.
- '.json' file extension.
- MIME type "application/json".
- Used for storing and exchanging data.
- Self describing and easy to understand.
- Language independent.\*\*

```
1 var myObj = {"name": "Web Applications", "term": "Fall 2017"};
2 var json = JSON.stringify(myObj);
3
4 var newObj = JSON.parse(json);
```

## Errors

```
1 try {
2     // Block of code to try
3 }
4 catch(err) {
5     // Block of code to handle errors
6 }
```

## References

- <https://www.w3schools.com/js/default.asp>
- <https://developer.mozilla.org/en-US/docs/Web/JavaScript>
- <https://jsfiddle.net/>