**JavaScript**

## References:

## W3 Schools, JavaScript, <https://www.w3schools.com/js/>

## Tutorialspoint, JavaScript, <https://www.tutorialspoint.com/javascript/index.htm>

## Why Study JavaScript?

JavaScript is one of the **3 languages** all web developers **must** learn:

   1. [**HTML**](https://www.w3schools.com/html/default.asp) to define the content of web pages

   2. [**CSS**](https://www.w3schools.com/css/default.asp) to specify the layout of web pages

1. **JavaScript** to program the behavior of web pages

**Versions of JavaScript:**

* The Original JavaScript ES1 ES2 ES3 (1997-1999)
* The First Main Revision ES5 (2009)
* The Second Revision ES6 (2015)
* The Yearly Additions (2016, 2017, 2018)

# **JavaScript Where To**

## inserted between <script> and </script> tags.

## JavaScript in <head> or <body>

Scripts can be placed in the <body>, or in the <head> section of an HTML page, or in both.

**See example: InScript.html**

## External JavaScript

External scripts are practical when the same code is used in many different web pages.

JavaScript files have the file extension **.js**.

To use an external script, put the name of the script file in the src (source) attribute of a <script> tag:

Examples:

<script src="myScript1.js"></script>  
<script src="myScript2.js"></script>

## External JavaScript Advantages

Placing scripts in external files has some advantages:

* It separates HTML and code
* It makes HTML and JavaScript easier to read and maintain
* Cached JavaScript files can speed up page loads

# **JavaScript Output**

## JavaScript Display Possibilities

JavaScript can "display" data in different ways:

* Writing into an HTML element, using innerHTML.
* Writing into the HTML output using document.write().
* Writing into an alert box, using window.alert().
* Writing into the browser console, using console.log().

### **4 Ways to Declare a JavaScript Variable:**

* Using var
* Using let
* Using const
* Using nothing

# **JavaScript Data Types**

JavaScript variables can hold different data types: numbers, strings, objects and more:

let length = 16;                               // Number  
let lastName = "Johnson";                      // String  
let x = {firstName:"John", lastName:"Doe"};    // Object

When adding a number and a string, JavaScript will treat the number as a string.

### **JavaScript:**

let x = 16 + 4 + "Volvo";

Result:

20Volvo

### **JavaScript:**

let x = "Volvo" + 16 + 4;

Result:

Volvo164

## JavaScript Strings

A string (or a text string) is a series of characters like "John Doe".

Strings are written with quotes. You can use single or double quotes:

let carName1 = "Volvo XC60";   // Using double quotes  
let carName2 = 'Volvo XC60';   // Using single quotes

## JavaScript Booleans

Booleans can only have two values: true or false.

## JavaScript Arrays

JavaScript arrays are written with square brackets.

Array items are separated by commas.

The following code declares (creates) an array called cars, containing three items (car names):

### **Example**

const cars = ["Saab", "Volvo", "BMW"];

## JavaScript Objects

JavaScript objects are written with curly braces {}.

Object properties are written as name:value pairs, separated by commas.

### **Example**

const person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};

## Function Invocation

The code inside the function will execute when "something" **invokes** (calls) the function:

* When an event occurs (when a user clicks a button)
* When it is invoked (called) from JavaScript code

**Example: function.html**

# **JavaScript Events**

## HTML Events

An HTML event can be something the browser does, or something a user does.

Here are some examples of HTML events:

* An HTML web page has finished loading
* An HTML input field was changed
* An HTML button was clicked

Often, when events happen, you may want to do something.

JavaScript lets you execute code when events are detected.

HTML allows event handler attributes, **with JavaScript code**, to be added to HTML elements.

Common HTML Events

Here is a list of some common HTML events:

|  |  |
| --- | --- |
| **Event** | **Description** |
| onchange | An HTML element has been changed |
| onclick | The user clicks an HTML element |
| onmouseover | The user moves the mouse over an HTML element |
| onmouseout | The user moves the mouse away from an HTML element |
| onkeydown | The user pushes a keyboard key |
| onload | The browser has finished loading the page |

# **JavaScript Classes**

ECMAScript 2015, also known as ES6, introduced JavaScript Classes.

JavaScript Classes are templates for JavaScript Objects.

* Constructor
* method

Example carClass.html

# **JavaScript Class Inheritance**

## To create a class inheritance, use the extends keyword.

**Example: inheritance.html**

**Run html files with JavaScript on Server**

**Install nodejs**

<https://nodejs.org/en/download/>.

>npm install -g http-server

where ever you need to run a html file run the command http-server

For ex: your html file is in /home/project/index.html you can do /home/project/$ http-server

That will give you a link to access your webpages:

http://localhost:8080/car\_index.html

Example:

* car.js
* car\_index.html