**JavaScript: The Language**

**An Introduction**

1. [***An introduction***](https://javascript.info/getting-started) ***to JavaScript***

*JavaScript* is an implementation of ECMAScript specification. JavaScript programs are called *Scripts* and these are written and executed (interpreted) as plain texts. They don’t require compilation like C# or Java programming.

JavaScript can be executed in the browser, on the server or any device that has a special program called *The JavaScript engine.* The browser has an embedded engine sometimes called a *“JavaScript virtual machine”*.

The different JavaScript engines are,

* V8 – Chrome
* SpiderMonkey – Firefox
* Chakra – IE
* ChakraCore - Edge

*HTML* – consists of static documents.

*CSS* – apply color, alignment and responsiveness to HTML document.

*JavaScript* – make documents or a page interactive (make web pages alive).

1. [***Code editors***](https://javascript.info/code-editors)
2. [***Developer console***](https://javascript.info/devtools)

[**JavaScript Fundamentals**](https://javascript.info/first-steps)

1. [***Hello***](https://javascript.info/first-steps) ***world***

The JavaScript program can be inserted into any HTML documents using ***<script>*** tag. Such as below, ***inline scripts***

<!DOCTYPE *html*>

<html>

<body>

<script>

*alert*("Inline JavaScript!");

</script>

</body>

</html>

JavaScript code inside the *<script>* tags is automatically executed when the browser loads the HTML document & processes the tag.

If we have a lot of JavaScript code, alternately, we can put it into a separate file with an extension of ***.js***. These .js files can be attached to HTML document using ***src*** *attribute of <script> tag*. Like below – ***external scripts***

<!DOCTYPE *html*>

<html>

<body>

<! -- *path can be either relative to root folder or CDN.*

*Inline + external scripts can be combined -->*

<script *src*="app/external.js"></script>

<script *src*="https://unpkg.com/react@16.8.6/umd/react.development.js"></script>

<script *src*="app/tab.js"></script>

<script>

*alert*("Inline JavaScript!");

</script>

<script *src*="file.js">

*alert*(1); // *the content is ignored, because src is set*

</script>

</body>

</html>

*Note: The <script>tag has other attributes, for instance, language, type etc., those are not required anymore for the modern browsers.*

1. ***Code structure***
2. ***The modern mode, "use strict"***
3. ***Variables***

A variable is a *named memory* for data storage. To create a variable in JavaScript, use the ***let****,* ***const***or***var*** keyword.

**let**

letmessage; // *creates/declares/defines a variable with name "message"*

message = "Hey sporty, how are you!!!???"; // *assigning/storing the value*

letmessage="Hey sporty, how are you!!!???"; // *combining the declaration and assignment.*

console*.*log(message); // *Accessing the variable by name*

*alert*(message); // *Accessing the variable by name*

**const** (Once variable is assigned, it can’t be re-assigned)

const *baseDiscount* ="30"; // *assigning/storing the value*

const *PI* =3.14;

const *COLOR\_RED* ="#F00";

baseDiscount = "35"; // *Uncaught TypeError: Assignment to constant variable. error, can't reassign the constant!*

Unlike let, declaration & assignment cannot be separated. It should be always combined.

**TALK ABLOUT RE-ASSIGMENT OF ARRAY & OBJECT**

Limitations on variable names,

1. The name must contain only letters, digits, or the symbols $ and \_.
2. The first character must not be a digit.
3. Variable names are case-sensitive. Ex, message & Message are completely different.
4. Prefer camelCase for declaration, but not mandatory. Ex, customerName, priceTag
5. Don’t use reserved keywords. Ex, *let, class, return, and function* are reserved.
6. ***Data types***

JavaScript is *dynamically typed* language, meaning that there are data types but a variable can at one moment be a string and at another be a number:

Ex,

// *no error*

letmessage="hello";

message = 123456;

There are *seven* basic data types in JavaScript.

1. Number
2. String
3. Boolean
4. null
5. undefined
6. Object
7. Symbol
8. Number

The number type represents both integer & floating point (decimal) numbers, Ex, 7, 100, 3.14 etc. Also few special numeric values **Infinity, -Infinity and NaN**.

JavaScript: DOM