JavaScript Tutorial

**Link JS File** Inside the HTML HEAD/BODY 🡪 <script src= "scripts.js"></script>

**Declare A Variable** (\*Don’t declare types\*) 🡪 var myVariable;

\*STRINGS ARE STORED AS OBJECTS\*:

* Have internal properties: values
* var alpha = "ABCDEFG";
* var length = alpha.length;
* document.write(length); //Displays the length of the string- 7
* Also have methods: take input, do computation, & output answer
* var alpha = "ABCDEFG";
* var result = alpha.substring(3, 5);
* document.write(result); //Outputs DE- the 3rd & 4th letter - (A is in position 0)

**ARRAYS:**

* Holds many values in a single variable
* var a = new Array(7); //Defines size of array
* a[0] = "cat";
* a[1] = "dog;
* document.write(a[1]); //Displays dog

OR

* var a = new Array(cat, dog);

OR

* var a = ["cat", "dog"]; //Shorthand for writing new array

**FUNCTIONS:**

function sayHello() //Keyword function & name

{ document.write("Hello");

}

sayHello(); //Called the function

**FLOW CONTROL STATEMENTS:**

* **IF statements:**
* var a = 7;

if (a>10) {

alert(a)

}

else {

alert("The condition was false");

}

* **LOOPS:** (for loop)
* for (i= 0 ;i < 5 ;i++ ) //initializes & assigns i the value of 0, checks condition, increments by 1

{

document.write("This is iteration " + i + "<br>");

}

------------------------------------------------------------------------------------------------------------------------------------------

\*\*Random Important:

Document:

* The DOM document object is the owner of all other objects in your webpage
* represents your web page
* if you want to access any element in an HTML page, you should always start with accessing the document object
* **getElementById:** (method) most common way to
* **innerHTML:** (property) easiest way to get or change the content of an element

JavaScript W3Schools

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

* **HTML** to define the content of web pages
* **CSS** to specify the layout of web pages
* **JavaScript** to program the behavior of web pages

**\*INTRODUCTION:**

\*JavaScript can change HTML content; JavaScriptaccepts both “double” & ‘single’ quotations

One of many JavaScript HTML methods is **getElementById()** - the method "finds" an HTML element (with id="demo") and changes the element content (**innerHTML**) to "Hello JavaScript"

* document.getElementById(**"**demo**"**).innerHTML = **"**Hello JavaScript**";**
* document.getElementById(**'**demo**'**).innerHTML = **'**Hello JavaScript**'**;

Changing the style of an HTML element, is a variant of changing an HTML attribute:

* document.getElementById("demo").style.fontSize = "35px";

Change The Display Style 🡪

* Hide HTML elements:
* document.getElementById("demo").style.display = "none";
* Show hidden HTML elements:
* document.getElementById("demo").style.display = "block";

**\*WHERE TO:**

Script Tag in HTML 🡪

* <script>  
  document.getElementById("demo").innerHTML = "My First JavaScript";  
  </script>
* In HTML, JavaScript code must be inserted between <script> and </script> tags

Type attribute 🡪

* <script type= “text/javascript”>
* optional – JavaScript is already the default scripting language in HTML

External JavaScript Files 🡪

* <script src="myScript.js"></script>
* Place external script reference in the <head> or <body>
* Must end in ‘.js’

**Function:** a block of JavaScript code, that can be executed when “called for”

* functions can be called for when **events** occur – when the user clicks a button

**\*OUTPUT:**

Display Possibilities:

* **window.alert():**

- can use an alert box to display data

- alert("Hello"); //Outputs in form of dialog box ( of value in ‘( )’ )

* **document.write():**
* using document.write() after an HTML document is fully loaded, will**delete all existing HTML**
* **should only be used for testing purposes**
* document.write("Hello"); //Outputs to browser
* HTML-- <div id="test"></div>

JS-- document.getElementById("test").innerHTML = "Hello";

//Displays the same as above, but uses the getElementById method & innerHTML property

* **Console.log():**
* used for debugging purposes
* console.log("Hi there!"); //Shows up in the console but not on the web page

Using innerHTML:

* To access an HTML element, JavaScript can use the document.getElement.ById(id) method
* id: attribute defines the HTML element
* innerHTML: property defines the HTML content

**\*SYNTAX:**

JavaScript Syntax: the set of rules – how JS programs are constructed

\*A **computer program** is a list of **instructions** to be executed by a computer.

In a programming language these instructions are called **statements**.

JavaScript is a **programming** **language** & its statements are separated by **semicolons**.

In HTML 🡪 JavaScript programs are executed by the web browser

JavaScript Statements 🡪 composed of Values, Operators, Expressions, Keywords, and Comments

JavaScript Values 🡪 2 types:

* **Literals:** fixed values
  + numbers – written w/ or w/out decimals 🡪 100, 100.03
  + strings – text written between double or single quotes 🡪 “Marie”, ‘Marie’
* **Variables:** variable values
* used to store data values;
* JS uses the var keyword to declare & the ‘=’ to assign values

var x; x = 10;

JS Arithmetic Operators & Expressions

JS Keywords, Comments and Identifiers

JS Casing:

\*Character Set 🡪 Unicode

* Camel Casing 🡪 FirstName, firstName
* Underscores 🡪 First\_Name, first\_name
* Hyphens 🡪 not allowed in JavaScript

**\*STATEMENTS:**

\*In HTML, JavaScript statements are "instructions" to be "executed" by the web browser

* JavaScript Programs 🡪 contain many JS statements - executed in the order they were written

**\*COMMENTS:**

* Single line - //
* Comment block - /\* \*/

**\*VARIABLES:**

\*Containers for storing data values

**Identifiers:** JS Variables must be identified w/ these unique names (JS Identifiers are case-sensitive)

Assignment Operator 🡪 = & Equal To Operator 🡪 == ; Data Types 🡪 numbers & strings

Declare/Create:

* var firstVar; 🡪 declaration – variable has no value/ has value of ‘undefined’
* firstVar = “Marie”; 🡪 assigning value to the variable

* var firstVar, secondVar, thirdVar;
* var firstVar = “Marie”, secondVar = “Hello”, thirdVar = 300;

OPERATORS, ARITHMETIC, ASSIGNMENT

**\*DATA TYPES:**

* JavaScript variables can hold many data types
* var length = 16;                               // Number
* var lastName = "Johnson";                      // String
* var x = {firstName:"John", lastName:"Doe"};    // Object
* JS variables are **Dynamic:** the same variable can be used to hold different data types
* var x;               // Now x is undefined
* var x = 5;           // Now x is a Number
* var x = "John";      // Now x is a String
* JavaScript treats a number added w/ a string as a string
* var x = 16 + 4 + “Marie”; 🡪 20Marie
* var x = “Marie” + 16 + 4; 🡪 Marie164

JavaScript Data Types:

* strings
* numbers
* booleans
* Arrays
* Objects

typeof Operator can return one of these **primitive types** 🡪 string, number, boolean, undefined

typeof Operator can return one of these **complex types** 🡪 Function or Object

\*FUNCTIONS:

* A JavaScript function is a block of code designed to perform a particular task
* A JavaScript function is executed when "something" invokes it (calls it)
* function name(parameter1, parameter2, parameter3) {  
      code to be executed  
  }
* Function **parameters** are listed inside the parentheses () in the function definition
* Function **arguments** are the **values** received by the function when it is invoked
* Inside the function 🡪 arguments (parameters) behave as local variables

Function Invocation:

* Code inside the function will execute when something **invokes/calls** it
* When an event occurs
* When it is invoked / called from JavaScript code
* When it is automatically / self invoked

Function Return:

* Return Statement ends the function;
* Return Value: computed & returned back to the calling statement / method

var x = myFunction(4, 3);    // Function is called, return value will end up in x  
  
function myFunction(a, b) {  
    return a \* b;            // Function returns the product of a and b  
}