JavaScript

JavaScript is the language used for HTML and CSS. This helps HTML elements to perform actions i.e behavior of web pages. The JavaScript function is executed when an event is occurred i.e when a button is clicked.

getElementById() is the most common method used in JavaScript. It is used in accessing the elements of HTML by using its ID.

Ex: document.getElementById(“demo”).innerHTML = “Welcome to the JavaScript coding”;

Document.getElementById(“demo”).style.display = ‘none’;

document.getElementById("demo").style.display="block";

In a HTML page the javascript code is written inside a script tag and the script tag can be placed in the head section and body section of the webpage.

<script>

Function myFunction(){

Document.getElementById(“demo”).innerHTML = “Content Changed”;

}

</script>

It is always preferred to separate the HTML code and javascript. It will be easy to read and maintain. The cached js files can speed up the page loads.The javascript file must be saved as .js file. The reference of the file can be inserted in the HTML page.

<!Doctype HTML>

<head>

<title>Javascript</title>

</head>

<body>

<script src=”script.js”>

</script>

</body>

To print out the result there are no built in functions in JavaScript. It is displayed in different ways.

Document.write() – It writes the result

Ex: <script>

Document.write(“This is one of the function used to write the result.”);

</script>

Document.alert(“This page may contain spam content”);

Document.getElementById(“id”).innerHTML = “Content of a HTML element can be changed using this way”;

Console.log(8+1); - This result is displayed in the console.

JavaScript Syntax: The programming language is a set of instructions i.e giving statements. JavaScript programs are executed by the browser.

Var x =10;

Var y=20;

JavaScript statements are composed of values,operators,expressions,keywords and comments.

JavaScript values are of two types. Fixed values are literals and variable values are called varaiables.

Literals ex: document.getElementById(“id”).innerHTML = “This is the ex of literals”;

Variables ex: Var x;

x=9;

JavaScript is case sensitive and in order to join two words we cannot use hyphens in JavaScript and can use camel case and underscore.

Single line comments are given using // and multiline comments are done by placing the code in between /\* \*\.

JavaScript Variables:

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Variables</h1>

<p>In this example, x, y, and z are variables</p>

<p id="demo"></p>

<script>

var x = 5;

var y = 6;

var z = x + y;

document.getElementById("demo").innerHTML = z;

</script>

</body>

</html>

Multiple values can be given using one variable var.

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Variables</h1>

<p>Create a variable, assign a value to it, and display it:</p>

<p id="demo"></p>

<script>

var carName = "Volvo";

document.getElementById("demo").innerHTML = carName;

</script>

</body>

</html>

If a variable is re-declared then it doesn’t lose the previous value.

Ex: Var companyName = “IBM”;

Var companyName;

Here the value of the companyName will be still IBM. It doesn’t lose the value.

JS Arithmetic: Operands are numbers in arithmetic operations. Operation performed between two operands is called operator.

<!DOCTYPE html>

<html>

<body>

<p>When adding a number and a string, JavaScript will treat the number as a string.</p>

<p id="demo"></p>

<script>

var x = "Volvo" + 16;

document.getElementById("demo").innerHTML = x;

</script>

</body>

</html>

JavaScript Operators:

The difference between == and === is when we use == data type doesn’t matter i.e when we compare the values it doesn’t bother about the datatype. When we use === operator the data type must be same on both sides. This is called type coercion operator.

JavaScript functions:

When a function is invoked without () operator, function definition will be returned.

Ex:

Function myCelsius(f){

Return (5/9)\*(f-32);

}

Solution: Function myCelsius(f){Return (5/9)\*(f-32);}

JavaScript objects are containers of name value pairs.

Object Definition:

Var person = {firstName:”Anusha”,lastName:”Thedla”,Age:24};

Properties of a object can be accessed in two way.

Ex:1: person.lastName;

Ex:2: person[“lastName”];

<script>

var person = {

firstName: "John",

lastName : "Doe",

id : 5566,

fullName : function() {

return this.firstName + " " + this.lastName;

}

};

document.getElementById("demo").innerHTML = person.fullName;

If you assign a value to a variable without declaring then the variable becomes global automatically.

myFunction();  
  
// code here can use carName   
  
function myFunction() {  
    carName = "Volvo";  
}

JavaScript Events:

When we click on a button or when content of a HTML page is changed then those are called events. JS lets to execute code when an event is occurred.

Sample syntax: <some-HTML-element some-event=**"*some JavaScript*"**>

Ex: <button onclick=”document.getElementById(“demo”).innerHTML = Date()”>The time is</button>

the code changes the content of its own element (using **this**.innerHTML):

<button onclick="this.innerHTML=Date()">The time is?</button>

|  |  |
| --- | --- |
| **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 |