INTRODUCTION TO JAVASCRIPT

JavaScript is a lightweight, cross-platform, single-threaded, and interpreted compiled programming language. It is also known as the scripting language for webpages. It is well-known for the development of web pages, and many non-browser environments also use it.

JavaScript is a weakly typed language (dynamically typed). JavaScript can be used for Client-side developments as well as Server-side developments. JavaScript is both an imperative and declarative type of language. JavaScript contains a standard library of objects, like Array, Date, and Math, and a core set of language elements like operators, control structures, and statements.

What can JavaScript Do?

• JavaScript can dynamically modify an HTML page

• JavaScript can react to user input

• JavaScript can validate user input

• JavaScript can be used to create cookies (yum!)

• JavaScript is a full-featured programming language

• JavaScript user interaction does not require any communication with the server

How to Link JavaScript File in HTML?

JavaScript can be added to HTML file in two ways:

Internal JS: We can add JavaScript directly to our HTML file by writing the code inside the <script> tag. The <script> tag can either be placed inside the <head> or the <body> tag according to the requirement.

External JS: We can write JavaScript code in another files having an extension.js and then link this file inside the <head> tag of the HTML file in which we want to add this code.

Using JavaScript in your HTML

• JavaScript can be inserted into documents by using the SCRIPT tag

```
<html>
    <head>
        <title>Hello World in JavaScript</title>
    </head>
    <body>
        <script type="text/javascript">
        document.write("Hello World!");
        </script>
    </body>
</html>
```

Applications of JavaScript

Web Development: Adding interactivity and behavior to static sites JavaScript was invented to do this in 1995. By using AngularJS that can be achieved so easily.

Web Applications: With technology, browsers have improved to the extent that a language was required to create robust web applications. When we explore a map in Google Maps then we only need to click and drag the mouse. All detailed view is just a click away, and this is possible only because of JavaScript. It uses Application Programming Interfaces(APIs) that provide extra power to the code. The Electron and React are helpful in this department.

Server Applications: With the help of Node.js, JavaScript made its way from client to server and Node.js is the most powerful on the server side.

Games: Not only in websites, but JavaScript also helps in creating games for leisure. The combination of JavaScript and HTML 5 makes JavaScript popular in game development as well. It provides the EaseJS library which provides solutions for working with rich graphics.

Smartwatches: JavaScript is being used in all possible devices and applications. It provides a library PebbleJS which is used in smartwatch applications. This framework works for applications that require the Internet for their functioning.

Art: Artists and designers can create whatever they want using JavaScript to draw on HTML 5 canvas, and make the sound more effective also can be used p5.js library.

Machine Learning: This JavaScript ml5.js library can be used in web development by using machine learning.

Mobile Applications: JavaScript can also be used to build an application for non-web contexts. The features and uses of JavaScript make it a powerful tool for creating mobile applications. This is a Framework for building web and mobile apps using JavaScript. Using React Native, we can build mobile applications for different operating systems. We do not require to write code for different systems. Write once use it anywhere!

Where to Put your Scripts

• You can have any number of scripts

• Scripts can be placed in the HEAD or in the BODY

– In the HEAD, scripts are run before the page is displayed

– In the BODY, scripts are run as the page is displayed

• In the HEAD is the right place to define functions and variables that are used by scripts within the BODY

Example :

```
<html>
    <head>
        <title>Hello World in JavaScript</title>
        <script type="text/javascript">
        function helloWorld()
        {
            document.write("Hello World!");
        }
        </script>
    </head>
```

```
<body>
<script type= "text/javascript">
        helloWorld();
</script>
</body>
</html>
```

External Scripts

• Scripts can also be loaded from an external file

• This is useful if you have a complicated script or set of subroutines that are used in several different documents

```
<script src= "myscript.js"></script>
```

JavaScript Variables

• JavaScript has variables that you can declare with the optional var keyword

• Variables declared within a function are local to that function

• Variables declared outside of any function are global variables

```
var myname = "Pat Morin";
```

JavaScript Operators and Constructs

• JavaScript has most of the operators we're used to from C/Java

– Arithmetic (+, - ,*, /, %)

– Assignment (=, +=, -=,*= /=, %=, ++, --)

– Logical (&&, ||, !)

– Comparison (<, >, <=, >=,==)

• Note: + also does string concatenation

• Constructs:

– if, else, while, for, switch, case

Simple User Interaction

• There are three built-in methods of doing simple user interaction

– alert(msg) alerts the user that something has happened

– confirm(msg) asks the user to confirm (or cancel) something

– prompt(msg, default) asks the user to enter some text

```
alert("There's a monster on the wing!");

confirm("Are you sure you want to do that?");

prompt("Enter you name" , "Adam");
```

JavaScript Functions

• JavaScript lets you define functions using the function keyword

• Functions can return values using the return keyword

```
function showConfirm() {

confirm("Are you sure you want to do that?");

}
```

JavaScript Arrays

• JavaScript has arrays that are indexed starting at 0

• Special version of for works with arrays

```
<script type= "text/javascript">

var colors = new Array();

colors[0] = "red";

colors[1] ="green";

colors[2] ="blue";
```

```
colors[3] ="orange";
colors[4] ="magenta";
colors[5] ="cyan";
for (var i in colors)
{
document.write("<div style=\"background-color:"+ colors[i] + ";\">"+ colors[i] + "</div>\n");
}
</script>
```

JavaScript Events

• JavaScript can be made to respond to user events

• Common Events:

– onload and onunload : when a page is first visited or left

– onfocus, onblur, onchange : events pertaining to form elements

– onsubmit : when a form is submitted

– onmouseover, onmouseout : for "menu effects"

JavaScript Objects

• JavaScript is object-oriented and uses the same method-calling syntax as Java

• We have already seen this with the document object

```
document.write("Hello World!");
```

Built-In JavaScript Objects

• Some basic objects are built-in to JavaScript

– String

– Date

– Array

– Boolean

– Math

JavaScript Strings

• A String object is created every time you use a string literal (just like in Java)

• Have many of the same methods as in Java

– charAt, concat, indexOf, lastIndexOf, match, replace, search, slice, split, substr, substring, toLowerCase, toUpperCase, valueOf

• There are also some HTML specific methods

– big, blink, bold, fixed, fontcolor, fontsize, italics, link, small, strike, sub, sup

• Don't use the HTML methods (use CSS instead)

– This is the worst kind of visual formatting

JavaScript Dates

• The Date class makes working with dates easier

• A new date is initialized with the current date

• Dates can be compared and incremented

```
var myDate = new Date();

myDate.setFullYear(2007,2,14);

var today = new Date();
```

JavaScript Arrays and Booleans

• We have already seen the Array class

• The Boolean class encapsulates a boolean value

```
var nextWeek = today + 7;
if (nextWeek > today) {
alert("You have less than one week left");
}
```

The JavaScript Math Class

• The Math class encapsulates many commonlyused mathematical entities and formulas

• These are all class methods

– abs, acos, asin, atan, atan2, ceil, cos, exp, floor, log, max, min, pow, random, round, sin, sqrt, tan

• These are all class methods

– E, LN2, LN10, LOG2E, LOG10E, PI, SQRT1_2, SQRT2

```
if (Math.cos(Math.PI) != 0) {
alert("Something is wrong with Math.cos");
}
```