JavaScript

JavaScript (JS) is a client side scripting language. JS is case sensitive. JS can be used inside any page or it can be referenced in a page from an external .js file.

JS code is either placed inside the <script> tag (in head of body) or referenced from an external .js file like **<script src="../xyz.js" type="text/javascript" />**

JavaScript code inside <script> tags should ideally be placed at the end of body element, so that the HTML page loads first and then JavaScript loads.

Comments inside JS should be used either as **//** for single line comment or **/* */** for multi line comment.

document.write() is used to write any text on the page.

To display quotes, escape it by using back slash.

<script> Document.write(“My name is \”Apurva\”””); </script>

Both ' and " can be used for enclosing text. HTML tags can also be used inside quotations.

For showing pop-up in a web page, use **alert**.

<script> alert(“”Hello!”); </script>

To get confirmation from the user, **confirm** can be used.

<script> confirm(“Do you want to leave?”); </script>

For getting input from the user, **prompt** is used. Prompt has 2 parameters. Can take value of any data type.

Syntax: **prompt("text", "default value");**

<script> prompt(“How old are you?”,”Doesn’t matter”); </script>

Variables are case sensitive.

var varName = “VALUE”;

var num = 23.5;

Variable names must start with letter, $, _

With the **window.location** variable, we can change the location of the current page.

<script> window.location=http://google.com/ </script>

Window.open has 4 optional parameters: url, name, options, replace (accepted in this order). Replace=true, will replace in the browser’s history with the url specified.

<script>

window.open(http://pinterest.com, “_blank”, “width=200,height=400,scrollable=no”, true);
</script>

Using # in href brings the user to the top of the page. **<a href=”#”>Link</a>**

If we want the anchor tag to not bring us to the top of the page (do nothing), we use:

<a href=”javascript:void(0);”>Empty Link</a>

Some important methods for String variables: length, toUpperCase(), toLowerCase(), replace(), match(), split(), substr(num1,num2)

//num1 is the starting index and num2 is the length to extract (optional)

Eg.: myStr.substr(2,5) or myStr.substr(2);

Boolean variables accepts true/false or 1/0.

Functions in JavaScript:

```
function funcName(param1,param2){ alert(param1+ +param2); }
funcName(Hello,World);

function addThis(num1,num2){ var sum = num1+num2; return sum; }
alert(addThis(1,2));
document.write(addThis(1,2));
var result = addThis(1,2);
```

Events:

```
<script>
    function Name(){
        alert("Hello");
    }
</script>
<span onClick='Name()'>Click Me!</span>
```

Other important events are:

onDblClick, onFocus, onBlur, onMouseDown, onMouseUp, onMouseMove, onMouseOut, onMouseOver, onKeyDown, onKeyPress, onKeyUp

Getting any HTML element by its ID:

```
<body>
    <input type='text' id='MyID' value='Value' />
    <script>
        var MyID = document.getElementById("MyID");
        MyID.value = "Value changed";
    </script>
</body>
```

```
<div id='test'>Change Me</div>
<script>
    document.getElementById('test').innerHTML = "<b>New HTML</b>";
</script>
```

CSS: background-color: red;

JS: document.getElementById(‘MyID’).style.backgroundColor = “red”;

There is an inbuilt function for escaping string. This is useful for escaping strings correctly. Escaped string is useful for URL.

escape(str);

Similarly, there is an inbuilt method to unescape the string.

unescape(str);

Array Declaration:

To start any new array, we should first define it.

```
var myArr = new Array();
myArr[0] = "Apurva";
myArr[1] = "Kumar";
myArr[2] = "Sinha";
var arr = new Array("Apurva","Kumar","Sinha");
var ar = ["Apurva","Kumar","Sinha"];
```

Multi-dimensional array declaration:

```
val mulArr = [["Apurva","Kumar","Sinha"],[1,2,3]];
alert(mulArr[0,0]+" : "+mulArr[1,0]); //This will alert "Apurva : 1"
```

Loops:

```
for(i=0; i<5; i++){
    document.write(i+"<br />");
}

for(v in arr){
    document.write(v+"<br />");
}
```

Date operations:

```
var d = new Date();
document.write(d.getDay());
//Sunday to Saturday: 0 to 6

document.write(d.getTime());
//Returns the time from Midnight Jan 1, 1970 to right not in milli seconds
```

Mathematical functions:

Math.round(num);   Math.pow(num,expo);   Math.PI;   Math.ceil(num);

Math.floor(num);   Math.min(arr);   Math.max(arr);

Defining a new regular expression:

var myRegExp = new RegExp(pattern, modifiers);

var myRegExp = /pattern/modifiers;

Example:

```
var myRegEx = /JS/gi;
var result = myRegEx.test("Welcome to js demo");
alert(result);
//'JS' is a pattern
//'g' stands for global and 'i' stands for case-insensitive match

var regEx = /sentence/g;
var str = "Change this sentence sentence";
document.getElementById("id1").innerHTML = str.replace(regEx, "string");
var strArr = str.match(regEx); //Match method will create an array of all
the matches
var matches = str.length;
```

We can get the user's browser information using the **navigator** keyword.

```
navigator.appName      //gives the name of the browser
navigator.appVersion   //gives the version of the browser
navigator.platform     //gives the user's platform like win32
```

var interval = setInterval(function(){funcName()}, timeInMillis); executes the function repeatedly after a time interval specified in the method parameter.

var delayedAlert = setTimeout(function(){alert("Delayed");},5000); executes the function just once after the specific time interval

Objects just like arrays can hold multiple values as well as other objects.

```
var myWallet = {
    cards:
            {
            visa: 1,
            masterCard: 0
            },
    cash:
            {
            USD: 1000,
            CAD: 100
            },
    id: 123
    }
```

AJAX

AJAX (Asynchronous JavaScript and XML) is a group of interrelated web development techniques used on the client-side to create asynchronous web application. With Ajax, web applications can send data to, and retrieve data from a server asynchronously (in the background) without interfering with the display and behaviour of the existing page. Data can be retrieved using XMLHttpRequest object.

Reference: https://www.udemy.com/thecompletewebdeveloper