Tutorial No. 07

Title: JavaScript Methods and Function, DOM Manipulation.

Batch: B1 Roll No.:16010420133 Tutorial:7

Aim: To implement javascript methods, functions and Event Handling to manipulate DOM

Resources needed: Notepad++, Web Browser

Theory:

JavaScript is a scripting language produced by Netscape for use within HTML Web pages. JavaScript is loosely based on Java and it is built into all the major modern browsers. JavaScript is a lightweight, interpreted programming language, Complementary to and integrated with Java, Complementary to and integrated with HTML, Open and cross-platform and is case sensitive.

Placing JavaScript in HTML document:

There is a flexibility given to include JavaScript code anywhere in an HTML document. But there are the following most preferred ways to include JavaScript in your HTML file:

1. Script in <head>...</head> section.
2. Script in <body>...</body> section.
3. Script in <body>...</body> and <head>...</head> sections.
4. Script in and external file and then include in <head>...</head> section.

An example of it is shown below:

```
<html>
<body>
<script language="javascript" type="text/javascript">
<!--
document.write("Hello World!")
//-->
</script>
</body>
</html>
```

Looping and Control statements in JavaScript:

- if statement syntax:

```
if (expression){
  Statement(s) to be executed if expression is true
}
```

- if else statement syntax:

```
if (expression){
   Statement(s) to be executed if expression is true
}else{
   Statement(s) to be executed if expression is false
}
```

- else if ladder syntax:

```
 if (expression 1){
    Statement(s) to be executed if expression 1 is true
 }else if (expression 2){
    Statement(s) to be executed if expression 2 is true
 }else if (expression 3){
    Statement(s) to be executed if expression 3 is true
 }else{
    Statement(s) to be executed if no expression is true
 }
```

- switch statement syntax:

```
 switch (expression)
 {
   case condition 1: statement(s)
                     break;
   case condition 2: statement(s)
                     break;
    ...
   case condition n: statement(s)
                     break;
   default: statement(s)
 }
```

- While Loop

```
  while (expression){
     Statement(s) to be executed if expression is true
  }
```

- do- while Loop

```
do{
    Statement(s) to be executed;
 } while (expression);
```

- for Loop

```
for(initialization; test condition; iteration statement){
Statement(s) to be executed if test condition is true
}

 for (variablename in object){
   Statement or block to execute
 }
```

Syntax for JavaScript functions:

```
function concatenate(first, last)
{
  var full;
  full = first + last;
  return full;
}
```

To invoke a function somewhere later in the script, you would simple need to write the name of that function.

Javascript Dialog boxes:

JavaScript supports three important types of dialog boxes. These dialog boxes can be used to raise and alert, or to get confirmation on any input or to have a kind of input from the users:

- Alert Dialog Box:
  An alert dialog box is mostly used to give a warning message to the users.
  ```
  alert("Warning Message");
  ```

- Confirmation Dialog Box:
  A confirmation dialog box is mostly used to take user's consent on any option. It displays a dialog box with two buttons: OK and Cancel.
  ```
  var retVal = confirm("Do you want to continue ?");
  ```
- Prompt Dialog Box:
  The prompt dialog box is very useful when you want to pop-up a text box to get user input. Thus it enables you to interact with the user. The user needs to fill in the field and then click OK.

  ```
  var retVal = prompt("Enter your name : ", "your name here");
  ```

In built objects in JavaScript:

A String object encapsulates a sequence of characters, enclosed in quotes

properties include :

- length : stores the number of characters in the string
- charAt(index) : returns the character stored at the given index (as in indices start at 0)
- substring(start, end) : returns the part of the string between the start (inclusive) and end (exclusive) indices
- toUpperCase() : returns copy of string with letters uppercase
- toLowerCase() : returns copy of string with letters lowercase

Arrays store a sequence of items, accessible via an index since JavaScript is loosely typed, elements do not have to be the same type. To create an array, allocate space using new (or can assign directly):

```
items = new Array(10);          // allocates space for 10 items
items = new Array();            // if no size given, will adjust dynamically
items = [0,0,0,0,0,0,0,0,0,0]; // can assign size & values []
```

To access an array element, use [] (as in C++/Java)

```
for (i = 0; i < 10; i++) {
    items[i] = 0;           // stores 0 at each index
}
```

The length property stores the number of items in the array.

The Date object can be used to access the date and time. To create a Date object, use new & supply year/month/day/… as desired

```
today = new Date();          // sets to current date & time
newYear = new Date(2002,0,1); //sets to Jan 1, 2002  12:00AM
```

Methods can access individual components of a date includes:

```
newYear.getYear()
newYear.getMonth()
newYear.getDay()
newYear.getHours()
newYear.getMinutes()
newYear.getSeconds()
newYear.getMilliseconds()
```

Document Object Model(DOM):

DOM Objects can be referenced using JavaScript

- by their id or name (this is the easiest way, but you need to make sure a name is unique in the hierarchy)

- by their numerical position in the hierarchy, by walking the array that contains them
- by their relation to parent, child, or sibling (parentNode, previousSibling, nextSibling, firstChild, lastChild or the childNodes

JavaScript and DOM:

JavaScript is used to manipulate the objects. For this id of an element is needed to be passed to method getElementById() of document object, which returns the element with the given id. And then we can alter its property.

For example,
if you want to find a <p> with the id of "cool", use:

```
getElementById("cool")
document.getElementById(item).style.backgroundColor =color;
```

to access the elements nested in <p> tag we can use,

```
document.getElementById(item).childNodes[1].style.backgroundColor =color;
```

Document Object:

innerHTML is a property of any document element that contains all of the html source and text within that element.

```
getElementById("cool").innerHTML ="new text string";
```

Methods:

document.write(…) : method that displays text in the page

document.URL : property that gives the location of the HTML document

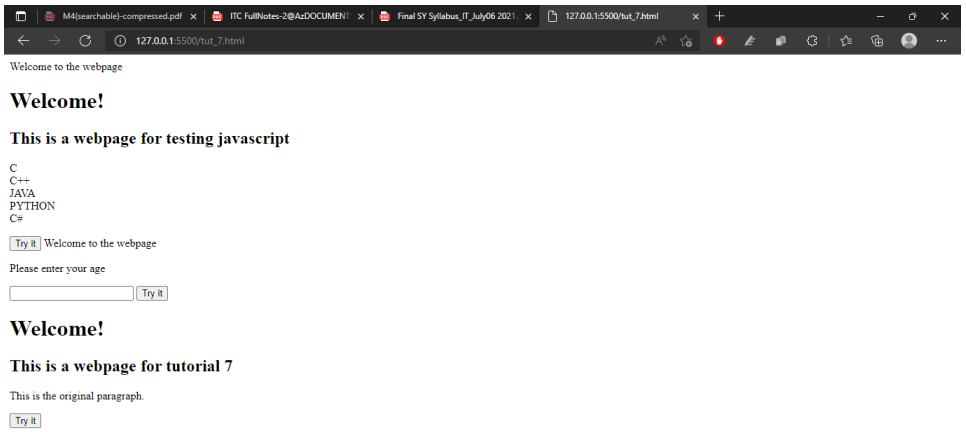
document.lastModified :property that gives the date & time the HTML document was last changed

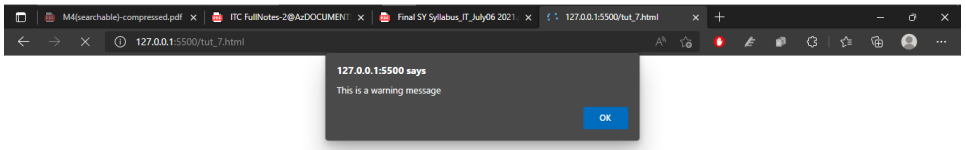
Activity:

1. Explore different methods of in-built JavaScript objects date, string, math, array etc.
2. Include at least two significant methods of some of these objects in your script
3. Extract elements of document using DOM and manipulate same using methods

---

Results: (Program printout with output)





Program:

```
<html>
<head>
<script language="javascript" type="text/javascript">
function myFunction() {
document.getElementById("demo").innerHTML = "Paragraph changed.";
}
document.write("Welcome to the webpage")
alert("This is a warning message");
</script>
</head>
<body>
<h1>Welcome!</h1>
<h2>This is a webpage for testing javascript</h2>
<p id="demo">This is the original paragraph.</p>
<button type="button" onclick="myFunction()">Try it</button>
</body>
</html>
<html>
<head>
<script language="javascript" type="text/javascript">
function myFunction() {
document.getElementById("demo").innerHTML = "Paragraph changed.";
}
```

```
document.write("Welcome to the webpage")
//alert("This is a warning message");
//var retVal = confirm("Do you want to continue ?");
//var retVal = prompt("Enter your name : ", "your name here");
</script>
</head>
<body>
<p>Please enter your age</p>
<input id="myInput" type="text">
<button onclick="myFunction()">Try it</button>
<p id="demo"></p>
<script>
function myFunction()
{
var age = document.getElementById("myInput").value;
var text;
if (letter === "19")
{
text = "You are eligible";
}
else if (age!='19')
{
text = "You are not eligible";
}
document.getElementById("demo").innerHTML = text;
}
let text1='';
const branches = ["C", "C++", "JAVA", "PYTHON","C#"];
for (let i = 0; i < branches.length; i++)
{
text1 += branches[i] + "<br>";
}
document.getElementById("demo").innerHTML = text1;
</script>
<h1>Welcome!</h1>
<h2>This is a webpage for tutorial 7</h2>
<p id="demo">This is the original paragraph.</p>
<button type="button" onclick="myFunction()">Try it</button>
</body>
</html>
```

Questions:

Q1) Explain with examples on how Javascripts help in creating dynamic HTML page.
Ans. JavaScript is a scripting language which is done on a client-side. The various browser supports JavaScript technology. DHTML uses the JavaScript technology for accessing, controlling, and manipulating the HTML elements. The statements in JavaScript are the commands which tell the browser for performing an action. An event is defined as changing the occurrence of an object. It is compulsory to add the events in the DHTML page. Without events, there will be no dynamic content on the HTML page. The event is a term in the HTML, which triggers the actions in the web browsers. Suppose, any user clicks an HTML element, then the JavaScript code associated with that element is executed. Actually, the event handlers catch the events performed by the user and then execute the code. Example of events: 1. Click a button.
2. Submitting a form.
3. An image loading or a web page loading, etc.

Q2) What is DOM? Explain.
Ans.
Every web page resides inside a browser window which can be considered as an object. A Document object represents the HTML document that is displayed in that window. The Document object has various properties that refer to other objects which allow access to and modification of document content. The way a document content is accessed and modified is called the Document Object Model, or DOM. The Objects are organized in a hierarchy. This hierarchical structure applies to the organization of objects in a Web document.
• Window object − Top of the hierarchy. It is the outmost element of the object hierarchy.
• Document object − Each HTML document that gets loaded into a window becomes a document object. The document contains the contents of the page.
• Form object − Everything enclosed in the ... tags sets the form object.
• Form control elements − The form object contains all the elements defined for that object such as text fields, buttons, radio buttons, and checkboxes.

Outcomes:

Apply JavaScript and JSON for web application development.

Conclusion: (Conclusion to be based on the outcomes achieved)

The various components of Javascript and Javascript DOM were explored and implemented.

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of faculty in-charge with date

References:

Books/ Journals/ Websites:

- "Web technologies: Black Book", Dreamtech Publications
- http://www.w3schools.com