

Tutorial No. 07

**Title: JavaScript Methods and Function , DOM
Manipulation.**

Batch: B2**Roll No.:1914078****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 crossplatform 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 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 (expression){
 Statement(s) to be executed if expression is true
}

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

- 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 simply 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 an 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

methods include

`charAt(index)` : returns the character stored at the given index
(as in C++/Java, 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:

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

Results: (Program printout with output)**CODE:**

```

<!DOCTYPE html>
<html lang="en">
<head>
  <title>TUT 7</title>
  <!-- Use of bootstrap -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta1/dist/css/bootstrap.min.css"
rel="stylesheet"
  integrity="sha384-
giJF6kkoqNQ00vy+HMDP7azOuL0xtbflcaT9wjK Hr8RbDVddVHyTfAAsrekwK mP1"
crossorigin="anonymous">

  <!-- Use of css -->
  <style>
    .main {
      border: 2px solid black;
      padding-left: 80px;
      padding-right: 80px;
      border-radius: 30px;
    }
    .con {
      height: 150vh;
    }
    h2 {
      color: rgb(133, 12, 82);
    }
  </style>
</head>
<body>
  <div class="con d-flex flex-column justify-content-center align-items-center">
    <div class="main py-5">
      <center>
        <h1 class="pb-3" id="heading">Sign Up</h1>
      </center>
      <div class="d-flex">
        <form id="myForm">
          

          <div class="mb-3 d-flex">

```

```

<div class="me-3" style="width:50%;">
  <label class="form-label" for="fname">First name:</label>
  <input class="form-control" type="text" id="fname" class="textar"
name="fname" placeholder="Enter First Name" autofocus>
</div>
<div style="width:50%;">
  <label class="form-label" for="lname">Last name:</label>
  <input class="form-control" type="text" id="lname" class="textar"
name="lname" placeholder="Enter Last Name" autofocus>
</div>
</div>

<div class="mb-3">
  <label class="form-label">Gender:</label>
  <select class="form-select">
    <option value="M">Male</option>
    <option value="F">Female</option>
    <option value="O">Other</option>
  </select>
</div>

<div class="d-flex mb-3">
  <div class="me-3" style="width:50%;">
    <label class="form-label" for="start">Date Of Birth:</label>
    <input class="form-control" type="date" id="start" required name="trip-
start" value="2020-02-18" min="2020-02-18" max="2020-06-31">
  </div>
  <div style="width:50%;">
    <label class="form-label">Time of Birth:</label>
    <input class="form-control" type="time" id="time" name="time" required
autocomplete="off">
  </div>
</div>

<div class="d-flex mb-3">
  <div class="me-3" style="width:50%;">
    <label class="form-label">Phone number:</label>
    <input class="form-control" type="tel" id="phone" name="phone">
  </div>
</div>

<div class="d-flex mb-3">
  <div style="width:50%;">

```

```

        <label class="form-label" for="pwd">Enter the OTP:</label>
        <input class="form-control" type="password" id="pwd" name="pwd">
    </div>
</div>

<a href="terms.html" id="showPage" onerror="error()" style="text-align:
center;">(click here to read terms and conditions.)</a>

<div class="mb-3">
    <input class="form-check-input" type="checkbox" name="accept"
value="accept">
    <label class="form-check-label" for="accept"> I accept the terms and
conditions of the hotel.</label>
</div>

<input class="btn btn-primary" type="submit" id="submitbtn" value="Sign
Up">
    <input class="btn btn-danger" type="reset" id="resetbtn" value="Reset">
</form>
</div>
</div>
</div>
</body>
<script>

```

```

//Hyperlink /another page opening(web page loading)
document.getElementById('showPage').onclick = function showP() {
    alert('Redirecting');
}

```

```

document.getElementById("heading").innerHTML = "Sign Up to Google!";

```

```

// onAbort
function abort(e){
    e.target.style.text="no image found";
}

```

```

// onBlur
// onFocus
var x = document.getElementById("myForm");
x.addEventListener("focus", myFocusFunction, true);
x.addEventListener("blur", myBlurFunction, true);

```



```
function myFocusFunction() {
    document.getElementById("phone").style.background = "yellow";
}

function myBlurFunction() {
    document.getElementById("phone").style.background = "";
}

// onChange
document.getElementById("fname").onchange = function s() {
    myFunction();
}

function myFunction() {
    var x = document.getElementById("fname");
    x.value = x.value.toUpperCase();
}

// onClick
document.getElementById("submitbtn").onclick = function formSubmit(){
    alert("button was clicked !!");
}

// onDbclick
document.getElementById("resetbtn").ondblclick = function formreset(){
    alert("Form has been reset!!");
}

// onError
function error(e){
    e.target.style.text="no image found";
}

// onKeyPress
// onKeyDown
// onKeyUp
document.getElementById("pwd").onkeydown = function password(e){
    e.target.style.background="red";
}
document.getElementById("pwd").onkeyup = function password(e){
    e.target.style.background="";
}
```

```
// onLoad
document.getElementsByTagName("form").addEventListener("load", alert("Page has been
loaded!!"));

// onMouseDown
// onMouse
document.getElementById("submitbtn").onmousedown = function formSubmit(e){
    e.target.style.background = "rgb(255, 184, 184)";
    e.target.style.border="3px solid maroon";
}

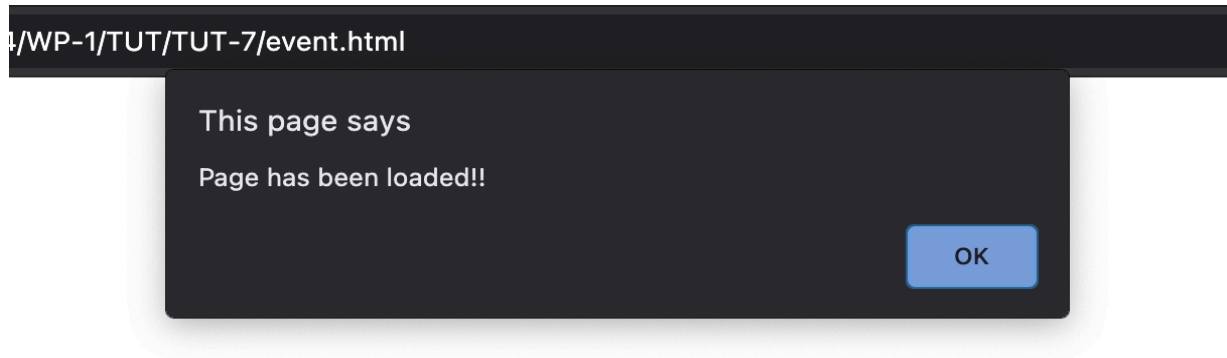
document.getElementById("submitbtn").onmouseover = function formSubmit(e){
    e.target.style.removeProperty('border');
    e.target.style.removeProperty('background');
}

// mouseEventHandler
var textar = document.getElementsByClassName("textar");

for(var i=0; i < textar.length; i++){
    textar[i].onmouseover = mouseEventHandler;
    textar[i].onmouseout = mouseEventHandler;
}
function mouseEventHandler(e){
    if(e.type == "mouseover"){
        e.target.style.background = "rgb(255, 184, 184)";
        e.target.style.border="3px solid maroon";
    }
    else {
        e.target.style.removeProperty('border');
        e.target.style.removeProperty('background');
    }
}
</script>
</html>
```


OUTPUT:

1. Whenever you load a page, you get an alert :



2. This is how the form looks when loaded:

Sign Up to Google!



First name:

Last name:

Gender:

Male

Date Of Birth:

Time of Birth:

Phone number:


Enter the OTP:

[\(click here to read terms and conditions.\)](#)
☐ I accept the terms and conditions of the hotel.

Sign Up

Reset

3. As soon as you enter the name, the name is transformed to uppercase:



First name:

Last name:

Gender:

4. When you focus on the phone number its background turns to yellow , when you blur it returns to white :

Phone number:

Enter the OTP:

Phone number:

Enter the OTP:

5. Whenever you press a key (keyDown)then the background color of password area changes to red and on releasing the key(i.e keyUp) the background color of password area goes back to white.

Phone number:

8879108369

Enter the OTP:

.....|

After releasing the key :

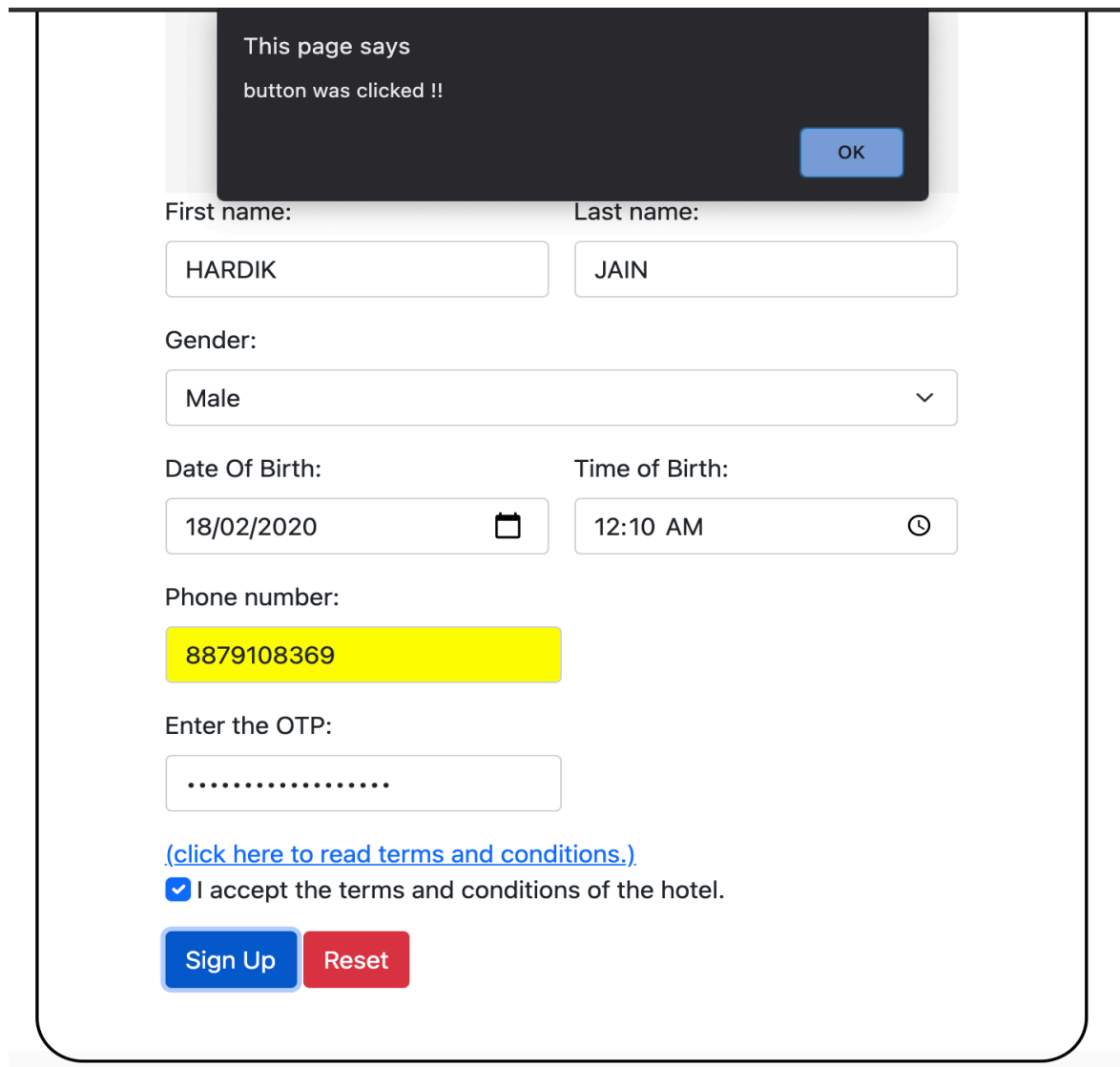
Phone number:

8879108369

Enter the OTP:

.....

6. After filling the form, when you hit on the submit button an alert button shows up :



This page says
button was clicked !!

OK

First name: Last name:

HARDIK JAIN

Gender:

Male

Date Of Birth: Time of Birth:

18/02/2020 12:10 AM

Phone number:

8879108369

Enter the OTP:

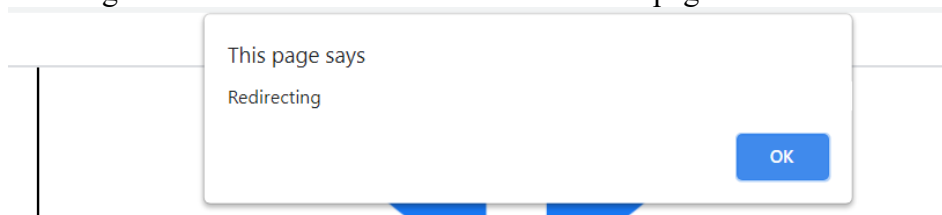
.....

[\(click here to read terms and conditions.\)](#)

☒ I accept the terms and conditions of the hotel.

Sign Up Reset

7. Clicking on the link throws alert and takes to new page:



This page says
Redirecting

OK

Terms and Conditions

- nadfndsomspdmppd
- nnidninodsnodln
- nskndvlfxnvkjbd

8. If you click the reset button twice (i.e doubleclick it) then another alert is thrown:

This page says
Form has been reset!!

OK

First name:

Enter First Name

Last name:

Enter Last Name

Gender:

Male

Date Of Birth:

18/02/2020

Time of Birth:

--:-- --

Phone number:

Enter the OTP:

[\(click here to read terms and conditions.\)](#)
☐ I accept the terms and conditions of the hotel.

Sign Up

Reset

(Autonomous College Affiliated to University of Mumbai)

Questions:

Q1) Explain with example how Javascripts help in creating dynamic HTML page.

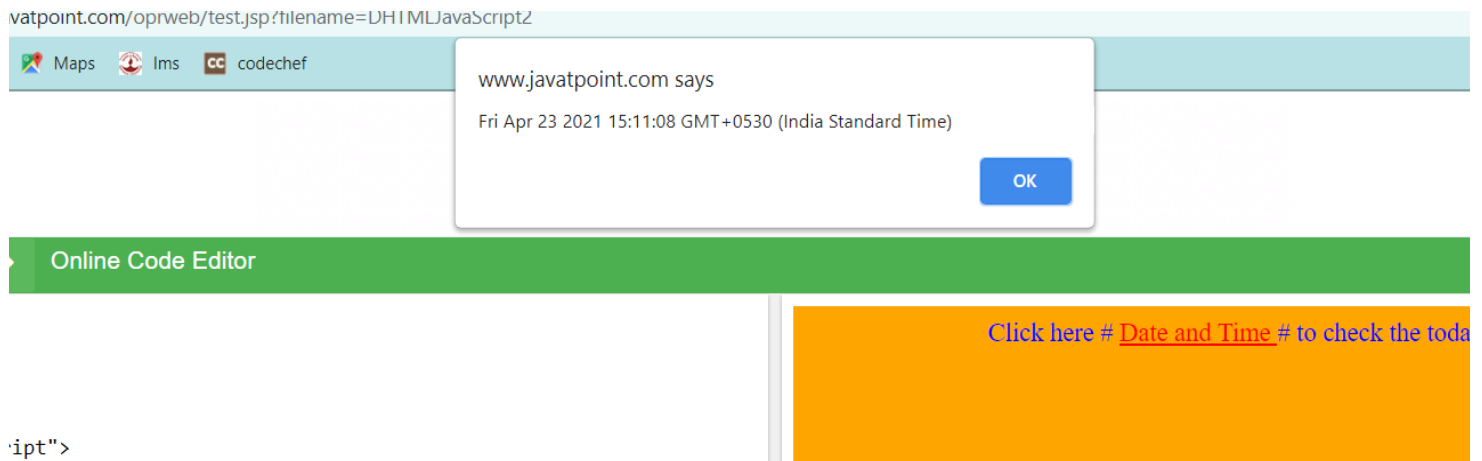
Ans. Following are the uses of DHTML (Dynamic HTML):

- It is used for designing the animated and interactive web pages that are developed in real-time. It helps users by animating the text and images in their documents.
- It also allows the page authors for including the drop-down menus or rollover buttons.
- It is also used to create various browser-based action games.
- It is also used to add the ticker on various websites, which needs to refresh their content automatically.
- It allows the users to alter the font, size, color, and content of a web page.
- It provides the facility for using the events, methods, and properties. And, also provides the feature of code reusability.
- It also provides the feature in browsers for data binding.
- users can easily change the tags and their properties.
- The web page functionality is enhanced because it uses low-bandwidth effect.

Example : The following example shows the current date and time with the JavaScript and HTML event (OnClick). In this example, we type the JavaScript code in the <head> tag.

```
<html>
<head>
<title>
DHTML with JavaScript
</title>
<script type="text/javascript">
function dateandtime()
{
    alert(Date());
}
</script>
</head>
<body bgcolor="orange">
<font size="4" color="blue">
<center> <p>
Click here # <a href="#" onClick="dateandtime();"> Date and Time </a>
# to check the today's date and time.
</p> </center>
</font>
</body>
</html>
```

Click here # [Date and Time](#) # to check the today's date and time.



In the above code, we displayed the current date and time with the help of JavaScript in DHTML. In the body tag, we used the anchor tag, which refers to the page itself. When you click on the link, then the function of JavaScript is called.

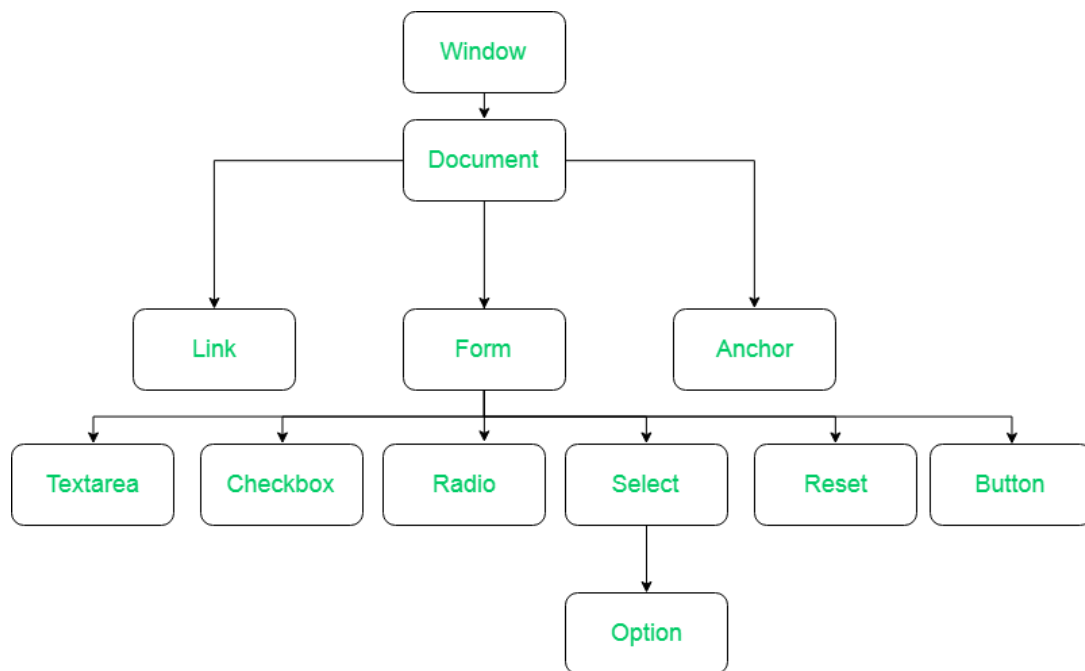
In the JavaScript function, we use the alert() method in which we type the date() function. The date function shows the date and time in the alert dialog box on the web page.

Q2) What is DOM? Explain

Ans. The Document Object Model (DOM) is a programming interface for HTML and XML(Extensible markup language) documents. It defines the logical structure of documents and the way a document is accessed and manipulated. It is called as a Logical structure because DOM doesn't specify any relationship between objects.

DOM is a way to represent the webpage in the structured hierarchical way so that it will become easier for programmers and users to glide through the document. With DOM, we can easily access and manipulate tags, IDs, classes, Attributes or Elements using commands or methods provided by Document object.

The properties of document object that can be accessed and modified by the document object:



Outcomes:

CO3: Apply JavaScript for Web Application development.

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

In this experiment, we learnt how to extract elements of document using DOM and explored different methods of inbuilt JavaScript objects from a webpage.

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>
-