

**Government Arts and Science College (Women),
Sathankulam – 628704**

Class : II B.Sc. (Computer Science)

Semester : IV

Subject : Web Technology

Topic : Unit IV

Faculty : Mrs. A. Angeline Nancy Sophia M.Sc., M.Phil.,
Department of Computer Science

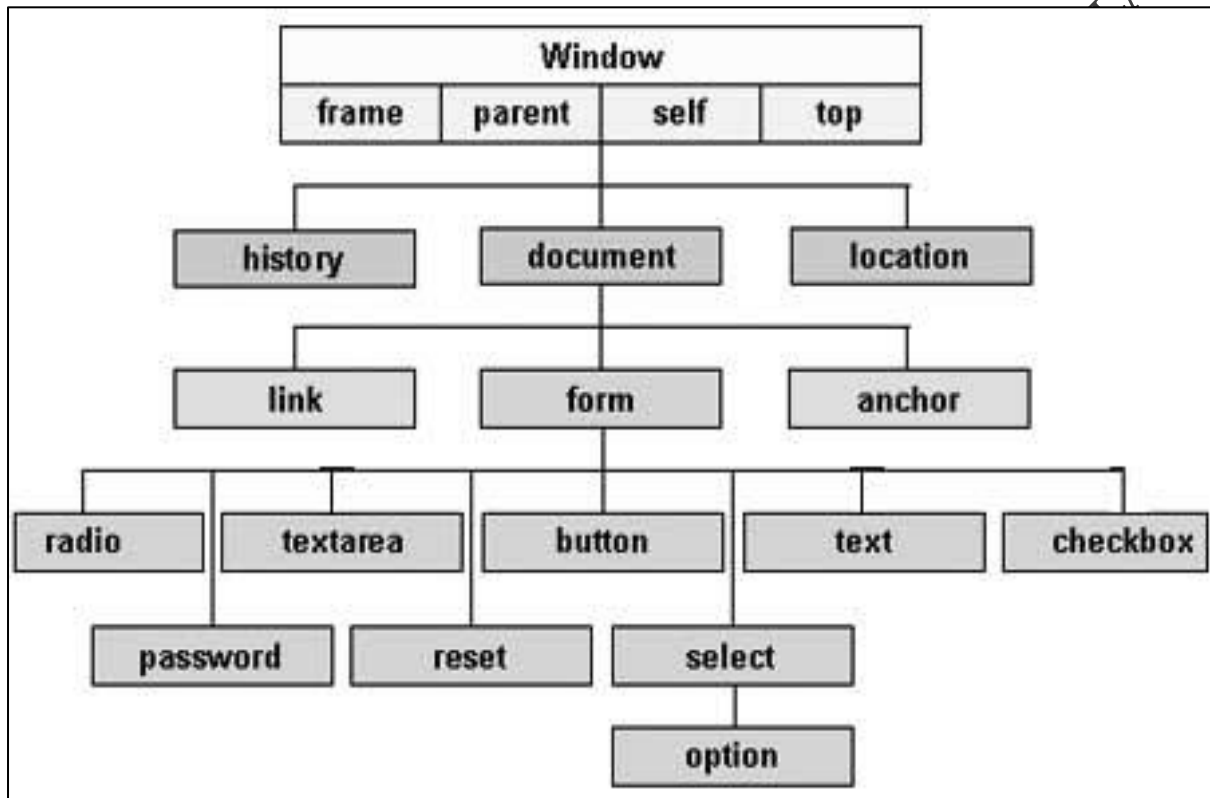
DEPARTMENT OF COMPUTER SCIENCE, GASC(W) SATHANKULAM.

DOCUMENT OBJECT MODEL

An object is a set of variables, functions etc that encapsulate data and function.

objects may have properties or variables associated with an object, methods which are functions associated with an object and events which notify that a particular event has occurred.

The objects are arranged into a hierarchy known as Document Object Model (DOM)



WINDOW OBJECT

Window is the fundamental object in the browser. It represents the browser window in which the document appears.

Properties

- Status – the contents of the status bar.
`window.status = "Hi";` will display the string "Hi" on the status bar.
- Location – the location and the URL of the document loaded into the window.
`alert(window.location);` will display an alert message containing the location and the URL of the document.
- Length – the number of frames into which the current window is divided.
`alert(window.length);` will display an alert message indicating the number of frames in the current window.

- Parent – the parent window, if the current window is a sub-window in a frameset.

```
var parentWindow = window.parent;
```

```
alert (parentWindow.length);
```

used to report the number of frames if any in the parent window.

- Top – the top-level window of which all other windows are sub-windows.

```
var topWindow = window.top;
```

```
alert (topWindow.length);
```

used to report the number of frames if any in the top-level window

Methods

- alert() – displays an alert dialog box, containing text and an OK button.

```
alert("Hi");
```

will display a dialog box containing the message "Hi".

- confirm() – displays a confirm dialog box, containing text entered by the user and an OK button and a Cancel button. The method returns true or false.

```
var response = confirm ("surely want to delete file?");
```

```
alert(response);
```

will display the message along with an OK and Cancel button. If the user click ok the variable will contain true value or if the user clicks cancel button the variable will contain false.

- prompt() – displays a message box into which the user can type text, an Ok button and a cancel button. The method returns a text string.

```
var filename = prompt("select file", "file.txt");
```

```
alert(filename);
```

will display a dialog box containing the message "Select file" along with an Ok and Cancel button and an area into which the user can type. This area will contain a string "file.txt" but can be overwritten with a new name.

- open() – opens a browser window and loads either an existing page or a new document into it.

```
open(URL_string, name_string, parameter_string)
```

```
var parameters = "height=100, width=200";
```

```
newWindow = open("NewDoc.html", "newDocument",  
parameters);
```

will open a window 100 pixels high by 200 pixels wide. An HTML document called "NewDoc.html" will be loaded into the window.

- close() – closes a window. If no window is specified, closes the current window.

```
window_name.close();
```

`newWindow.close();` will close the new window opened.

Events

- `onLoad()` – message is sent each time when a document is loaded into a window. Usually placed within the `<body>` tag.

```
<body onLoad = "displayWelcome()">
```

the function `displayWelcome()` to execute automatically every time the document is loaded or refreshed.

- `onUnload()` - message is sent each time when a document is closed or replaced with another document. Usually placed within the `<body>` tag.

```
<body onUnload = "displayclose()">
```

the function `displayclose()` to execute automatically every time the document is closed or refreshed.

DOCUMENT OBJECT

The document object represents the HTML document displayed in a browser window.

Properties

- `bgColor` – set the color of the background
`document.bgColor = "green";`
- `fgColor` – set the color of the text
`document.bgColor = "blue";`
- `linkColor` – set color for the un-visited links.
`document.linkColor = "brown";`
- `alinkColor` – set color for the active link
`document.alinkColor = "lightred";`
- `vlinkColor` – set color for visited links.
`document.vlinkColor = "darkred";`
- `title` - set the title of the document as displayed at the top of the browser window.

```
document.title = "This title is displayed now";
```

- `forms` – an array containing all the forms in the document. It accepts an index number.

`forms[index-number]` index number is the number of a particular form. First form will always have index-number 0.

Methods

- `write()` – allows a string of text to be written to the document.

```
document.write("<h1>Hello</h1>");
document.write("<p>Welcome to this new page </p>");
document.write("<p>To return to the recent page,");
document.write("<a href='NewDoc.html'>Click here </a> </p>");
```

will replace the existing page with the HTML code contained within brackets of the document.write() methods.

FORMS OBJECT

When a form is created in an HTML document using the <form> and </form> tags a form object is created automatically with properties, methods and events that relate to the form itself and to the individual elements within the form.

Properties

- name – name of the form as defined when the form is created

```
alert(document.forms[2].name);
```
- method – the method used to submit the information in the form

```
alert(document.forms[2].method);
```
- action - the action to be taken when the form is submitted

```
<form action = "mailto:sales@bigb.com">
alert(document.forms[2].action);
```

will display the action property of the example for.
- length – number of elements in the form.

```
alert(document.forms[2].length);
```

will display the number of elements in the form.
- elements – array of elements in the form referenced by index number starting from 0

```
alert(document.forms[2].elements[0].name);
```

will display the name of the first element in the form.

Form Methods

submit() – submits the form data to the destination specified in the action attribute.

Form Events

onSubmit() – message is sent each time a form is submitted. Usually placed within the <form> tags.

```
<form onSubmit = "displayOK()">
```

which cause the function displayOK() to execute automatically every time the form is submitted.

TEXT-BOXES AND TEXT-AREAS

Properties

- name – name of the text-boxes or text-area as defined in the HTML <input> tag when the form is created.

```
<input type=text name="textBox1">
```

- value – the text typed into a text-box by the user.

```
alert(document.forms[2].textBox1.value);
```

Events

- onFocus() – event signal generated when a user clicks in a text-box or text-area.

```
<input type = text name="textBox2" onFocus = "alertonFocus()">
```

the function called alertonFocus() displays an alert box

- onBlur() – event signal generated when a user clicks outside a text-box or text-area.

```
<input type = text name="textBox3" onBlur = "alertonBlur()" >
```

the function called alertonBur() displays an alert box

BUTTONS, RADIO-BUTTONS AND CHECKBOXES

Properties

- name – name of the button, radio-button or checkbox as defined in the HTML <input> tag when the form is created.

```
<input type=button name="button1">
```

- value – the value given to the button when it is created.

```
document.forms[2].button1.value="New value,  
new label";
```

- checked – this property is used with radio-buttons and checkboxes. It indicates whether or not the button has been selected by the user.

```
if(document.forms[2].checkbox1.checked==true)
```

```
{  
    alert("Checked");
```

```
}
```

```
else
```

```
{  
    alert("Not checked");
```

```
};
```

SELECT OBJECT**Properties**

- name
- value
- selectedIndex – returns an integer indicating which option has been selected from a group by the user.

```
alert(document.forms[2].selectbox1.selectedIndex);
```

DATE OBJECT

The date object allows us to obtain the current date and time.

```
var myDateObject = new Date;
```

this creates an object called myDateObject that contains information about the date and time.

Once an instance of the object is created, any methods below can be used to obtain information from it.

Methods	Description
getFullYear()	Returns the current year as a four-digit number
getMonth()	Returns current month as an integer from 0-11
getDate()	Returns the day of the month as an integer between 1 and 31
getDay()	Returns the day of the week as an integer between 0 and 6. starting from Sunday
getHours()	Returns the hour of the day as an integer between 0 and 23
getMinutes()	Returns the number of minutes as an integer
getSeconds()	Returns the number of seconds as an integer

```
<html>
<head>
<title>An example for using date object</title>
<script language="Javascript">
    var myDateObject = new Date;
    var currentday = myDateObject.getDay();
    alert(currentday);
</script>
```

</head>

<body>

</body>

</html>

MATH OBJECT

it allows to perform various mathematical operations

Methods	Descriptions
sqrt(x)	Returns the square root of x
log(x)	Returns natural logarithm of x
max(x,y)	Returns larger value
min(x)	Returns smaller value
round(x)	Returns the value x to nearest integer
ceil(x)	Returns the value x rounded up to next integer
floor(x)	Returns the value x rounded down to the next integer
abs(x)	Returns absolute value of x
pow(x,y)	Returns the value of x raised to the power of y
sin(x)	Returns sine value
cos(x)	Returns cosine value
tan(x)	Returns tan value

STRING OBJECT

JavaScript includes string object that allows us to manipulate strings.

```
var myString = new String("Hello World");
```

or

```
var myString = "Hello World";
```

Methods	Descriptions
charAt()	Returns a character at a specified position
indexOf()	Searches a string if it contains specified character if does it returns the position of the character. If the character occurs more than once it returns the first occurrence of that character
lastIndexOf()	Searches a string if it contains specified character if does it returns the position of the character if the character occurs more than once it returns the last occurrence of that character
substring()	Returns the portion of a string between two specified positions.
substr()	Returns a portion of a string, starting from a specified position and continuing for a specified number of characters
charCodeAt()	Returns ASCII value of the character
fromCharCode()	Returns the characters specified by ASCII values
toString()	Converts number into a string
italics()	Converts the string to italics
bold()	Converts the string to bold
sup()	Superscripts the text
sub()	Subscripts the text

ARRAYS

An array is a set of variables that are grouped together and given a single name.

```
var myArray = new Array("Sarah", "Patrick", "Jane", "Tim");
```

Or

```
Var myArray = ["Sarah", "Patrick", "Jane", "Tim"];
```

An empty array can be created as follows:

```
var myArray = new Array();
```

Or

```
var myArray = new Array(5); to create an array of specific size.
```

The number of elements in the array can be determined using the length property

```
alert(myArray.length);
```

The entire content of array can be viewed using the valueOf() method.

```
alert(myArray.valueOf());
```

The value of particular element in the array can be obtained using its position in the array as an index.

```
var indexNumber = prompt("Please enter a number between 0 and 4", "");  
alert("Element" + indexNumber + "=" + myArray[indexNumber]);
```

DEPARTMENT OF COMPUTER SCIENCE, GASC(W) SATHANKULAM.

Example 1:**Write a script to add two integers****Solution:**

```
<html>
<head>
<title> To add two integers </title>
<script language= "javascript">
var no, no1, no2, no3, result;
no = window.prompt("Enter the first number" + "0");
no1 = window.prompt("Enter the second number" + "0");
no2 = parseInt(no);
no3 = parseInt(no1);
result = no2+ no3;
document.writeln("<h3> First Number =" + no2 + "(/h3><br>");
document.writeln("<h3> Second Number =" + no3 + "(/h3><br><br>");
document.write("<h2> Result =" + result + "</h2>");
</script>
</head>
<body>
</body>
</html>
```

Example 2:**Write a script that reads five integers and determines the largest and the smallest integers in the group.****Solution:**

```
<html>
<head>
<title> To find smallest & largest </title>
<script language= "javascript">
var no, no1, large, small, count;
count=0;
large=0;
small=999999;
while(count<5)
```

```

{
no = window.prompt("Enter the first number" + "0");
no1 = parseInt(no);
document.writeln("<h3> Number" + count + "=" + no1 + "</h3><br>");
If (no1>large)
large=no1;
if (no1<small)
small=no1;
count++;
}
document.writeln("<h2> The largest Number =" + large + "</h2>");
document.write("<h2> The smallest Number =" + small + "</h2>");
</script>
</head>
<body>
</body>
</html>

```

Example 3:

Write a script that reads integers and determines the square root of the integer using squareroot method of Math object.

Solution:

```

<html>
<head>
<title> To find square root of an integer </title>
<script language= "javascript">
var no, root;
no = window.prompt("Enter the number" + "0");
no1 = parseInt(no);
root = Math.sqrt(no1);
document.writeln("<h2> The Number =" + no1 + "</h2>");
document.write("<h2> The square root =" + root + "</h2>");
</script>
</head>
<body>
</body>

```

```
</html>
```

Example 4:

Write a function distance which calculates the distance between two points (x1,y1)and(x2,y2) incorporate this function into a script that enables the user to enter the coordinates of the points through an HTML form.

Solution:

```
<html>
<head>
<title> Distance calculation <title>
<script language = "javascript">
Function calculate ( )
{
Var a1, a2, a3, a5, r1, r2;
a1=parseInt(document.myform.x1.value);
a2=parseInt(document.myform.y1.value);
a3=parseInt(document.myform.x2.value);
a5=parseInt(document.myform.y2.value);
r1=a1-a3;
r1=Math.abs(r1);
r2=a2-a5;
r2=Math.abs(r2);
document.writeln("<h2> Distance =" +r1+" "+r2+"</h2>");
}
</script>
</head>
<body>
<h1 align =center> Finding distance between the coordinates </h1>
<form name= "myform">
Coordinate 1<br>
X1
<input type= "text " name= "x1" size=20>
<br>
Y1
```

```

<input type= "text" name= "y1" size=25>
<br>
Coordinate 2<br>
x2
<input type= "text" name= "x2" size=20>
<br>
Y2
<input type= "text" name= "x2" size=25>
<br>
<br>
<br>
<input type= "button" value= "calculator " onclick= "calculate()">
<input type="reset" value = "clear" onclick= "clear">
</form>
</body>
</html>

```

Example 5:

Write a script to find the number of occurrences of given a number in a set of stored numbers

Solution:

```

<html>
<head>
<title>Number of Occurrences </title>
<script language= "javascript">
Function find( )
{
Var a1,count;
Var set= [12, 35 ,78, 12 ,67, 50, 35, 67];
Count=0;
a1=parseInt(document.myform.no.value);
for (var i=0; i<set. length; i++)
{
if(a1==set[i])
count++;
}

```

```
document .writeln( "<h2> The element in an array </h2><br>");
for (var i=0;i<set. length ; i++);
{
document .write ("<p> + set[i] + "</p>");
}
document .writeln("<br><br><h3> Number of occurrences of "+ a1+ "is"
+ count + "</h3>");
}
</script ></head>
<body>
<h1 align = center>Finding the frequency of a number </h1>
<form name = "my form ">
    Enter a number<br>
<input type = " text" name "no" size=20>
<br><br><br>
<input type = "button" value="Find"() onclick="find">
<input type= " button" value ="Clear"() onclick= "clear">
</form>
</body>
</html>
```

DEPARTMENT OF COMPUTER SCIENCE, GASC(W) SATHANKULAM.