Document Object model (DOM):-

What is Html DOM?

- A DOM is an application programming interface (API) for representing a document and accessing and manipulating various Html elements that make up that document.
- JavaScript enabled web browsers have always defined a document object model, for eg**.** The forms in the html document are accessible through the forms[] array of the document object.
- When the document is loaded in the browser, it creates a number of JavaScript objects with properties and capabilities based on the HTML in the document and other pertinent information.
- These objects exist in hierarchy that reflects the structure of HTML page itself.
- The predefined objects that are most commonly used are window and document objects.
- The window has methods that allows you to create new windows with open() and close() methods. It also allows you to create message boxes using alert(), confirm() and prompt().Each displays the text that is written within parentheses.
- The document object models the HTML page. It contains arrays which stores all the components constituting the contents of your web page such as images, links, and forms. You can access and call methods on these elements through the arrays.
- The object in this predefined hierarchy can be accessed and modified.
- To refer to specific properties, you must specify the property name and all its ancestors, spelling out the complete hierarchy until the document object.
- A period(.) is used in between each object and the name of its property.
- The object/property gets its name from the name attribute of the HTML tag.
- For eg. To refer a value property of the text field named txt1 in the form named myform in the current document , we can use

  document.myform.txt1.value

- The form elements can also be accessed through the fore mentioned forms array of the document object.
- To refer to value property of first field of form we can use

  document. Forms[0].elements[0].value

- Functions of an objects can also be accessed using the period notaion. To reset the 2nd form on web page we can use
  document.forms[1].reset();

- ## Java Script Object Model:

An object is a set of variables, functions that are in some way related. They are grouped together and given a name.
Objects may have:

1] Properties: A variable associated with an object. Most properties can be changed by the user. Properties are the values associated with an object.

document. write(txt.length);

└→ Length property

2] Methods: Functions associated with an object. Methods can be called by the user.

document. write(txt.toUppercase());

└→ method

3] Events: Events are the notification that the particular event has occurred. It can be used by programmer to trigger responses. By using Java script, we have the ability to create dynamic web pages. Events are the actions that can be detected by java script.

Every element on a web page has certain event which can trigger java script functions.

Examples of events:

- A mouse click
- A web page or an image loading
- Mousing over a hot spot on the web page
- Selecting an input field in an HTML form
- Submitting an HTML form
- A keystroke

Note: Events are normally used in combination with functions, and the function will not be executed before the event occurs!

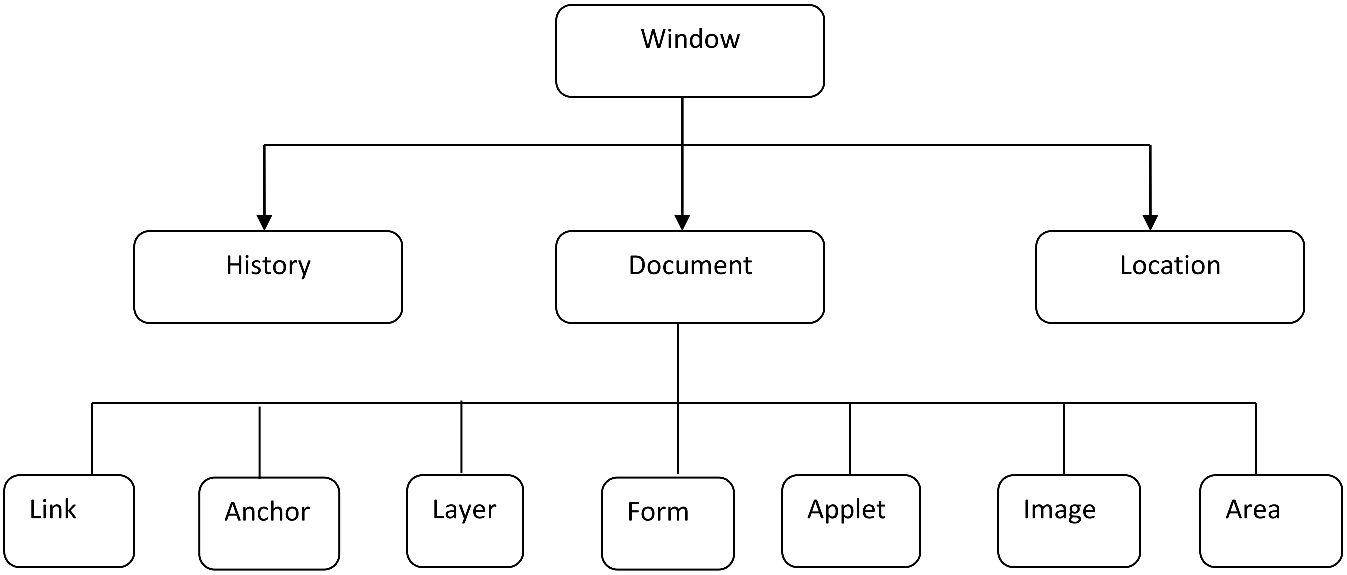
Event handlers

To allow you to run your bits of code when these events occur, JavaScript provides us with *event handlers*. All the event handlers in JavaScript start with the word `on`, and each event handler deals with a certain type of event. Here's a list of all the event handlers in JavaScript, along with the objects they apply to and the events that trigger them:

Event handler	Applies to:	Triggered when:
onAbort	Image	The loading of the image is cancelled.
onBlur	Button, Checkbox, FileUpload, Layer, Password, Radio, Reset, Select, Submit, Text, TextArea, Window	The object in question loses focus (e.g. by clicking outside it or pressing the TAB key).
onChange	FileUpload, Select, Text, TextArea	The data in the form element is changed by the user.
onClick	Button, Document, Checkbox, Link, Radio, Reset, Submit	The object is clicked on.
onDblClick	Document, Link	The object is double-clicked on.
onDragDrop	Window	An icon is dragged and dropped into the browser.
onError	Image, Window	A JavaScript error occurs.
onFocus	Button, Checkbox, FileUpload, Layer, Password, Radio, Reset, Select, Submit, Text, TextArea, Window	The object in question gains focus (e.g. by clicking on it or pressing the TAB key).
onKeyDown	Document, Image, Link, TextArea	The user presses a key.
onKeyPress	Document, Image, Link, TextArea	The user presses or holds down a key.
onKeyUp	Document, Image, Link, TextArea	The user releases a key.
onLoad	Image, Window	The whole page has finished loading.
onMouseDown	Button, Document, Link	The user presses a mouse button.
onMouseMove	None	The user moves the mouse.
onMouseOut	Image, Link	The user moves the mouse away from the object.
onMouseOver	Image, Link	The user moves the mouse over the object.
onMouseUp	Button, Document, Link	The user releases a mouse button.
onMove	Window	The user moves the browser window or frame.
onReset	Form	The user clicks the form's Reset button.
onResize	Window	The user resizes the browser window or frame.

onSelect	Text, Textarea	The user selects text within the field.
onSubmit	Form	The user clicks the form's Submit button.
onUnload	Window	The user leaves the page.

Java Script interacts with browsers through browsers object model. It is follows:



Window Object:-

The window object represents an open window in a browser.

If a document contain frames (<frame> or <iframe> tags), the browser creates one window object for the HTML document, and one additional window object for each frame.

Window Object Properties

Property	Description
closed	Returns a Boolean value indicating whether a window has been closed or not
defaultStatus	Sets or returns the default text in the statusbar of a window
frames	Returns an array of all the frames (including iframes) in the current window
length	Returns the number of frames (including iframes) in a window
name	Sets or returns the name of a window
opener	Returns a reference to the window that created the window
parent	Returns the parent window of the current window
self	Returns the current window
status	Sets the text in the statusbar of a window
top	Returns the topmost browser window

Window Object Methods

Method	Description
alert()	Displays an alert box with a message and an OK button
blur()	Removes focus from the current window
clearInterval()	Clears a timer set with setInterval()
clearTimeout()	Clears a timer set with setTimeout()
close()	Closes the current window
confirm()	Displays a dialog box with a message and an OK and a Cancel button
createPopup()	Creates a pop-up window

focus()	Sets focus to the current window
open()	Opens a new browser window
prompt()	Displays a dialog box that prompts the visitor for input
setInterval()	Calls a function or evaluates an expression at specified intervals (in milliseconds)
setTimeout()	Calls a function or evaluates an expression after a specified number of milliseconds
Eval(string)	Evaluates string as javascript code and returns the result.
Scroll(x,y)	Scrolls the window to the given x and y coordinates.

Window Object Events

Event	Description
OnBlur	Triggered when focus is removed from window
OnError	Triggered when error occur in window
OnFocus	Triggered when focus is applied to window
OnUnload	Triggered when user is exits from window
OnLoad	Triggered when web browser finishes loading a document into the window

Navigator Object

The navigator object contains information about the browser.

Navigator Object Properties

Property	Description
appCodeName	Returns the code name of the browser
appName	Returns the name of the browser
appVersion	Returns the version information of the browser
cookieEnabled	Determines whether cookies are enabled in the browser
platform	Returns for which platform the browser is compiled
userAgent	Returns the user-agent header sent by the browser to the server

Navigator Object Methods

Method	Description
javaEnabled()	Specifies whether or not the browser has Java enabled
taintEnabled()	Specifies whether or not the browser has data tainting enabled

History Object

The history object contains the URLs visited by the user (within a browser window).

The history object is part of the window object and is accessed through the window.history property.

History Object Properties

Property	**Description**
length	Returns the number of URLs in the history list
current	Contains the url of the current history entry
Previous	Contains the url of the current history stack entry
Next	Contains the url of the current history stack entry

History Object Methods

Method	**Description**
back()	Loads the previous URL in the history list
forward()	Loads the next URL in the history list
go(num)	Loads a specific URL from the history list. Goes forward num entries in the history stack if num>0, otherwise it goes backward –num entries
go(string)	Goes to the newest history entry whose title or url contains string as a substring.

Location Object

Location Object

The location object contains information about the current URL.

The location object is part of the window object and is accessed through the window.location property.

Location Object Properties

Property	Description
hash	Returns the anchor portion of a URL
host	Returns the hostname and port of a URL
hostname	Returns the hostname of a URL
href	Returns the entire URL
pathname	Returns the path name of a URL
port	Returns the port number the server uses for a URL
protocol	Returns the protocol of a URL
search	Returns the query portion of a URL

Location Object Methods

Method	Description
assign(string)	Loads a new document
reload()	Reloads the current document
replace()	Replaces the current document with a new one

- **Document Object**

Document Object

Each HTML document loaded into a browser window becomes a Document object.

The Document object provides access to all HTML elements in a page, from within a script.

Tip: The Document object is also part of the Window object, and can be accessed through the window. document property.

Document Object Collections

W3C: W3C Standard.

Collection	Description
anchors[]	Returns an array of all the anchors in the document
forms[]	Returns an array of all the forms in the document
images[]	Returns an array of all the images in the document
links[]	Returns an array of all the links in the document

Document Object Properties

Property	Description
cookie	Returns all name/value pairs of cookies in the document
documentMode	Returns the mode used by the browser to render the document
domain	Returns the domain name of the server that loaded the document
lastModified	Returns the date and time the document was last modified
readyState	Returns the (loading) status of the document
referrer	Returns the URL of the document that loaded the current document
title	Sets or returns the title of the document
URL	Returns the full URL of the document

- cookie - Used to identify the value of a cookie.

- domain - The domain name of the document server.
- embeds - An array containing all the plugins in a document.
- fgColor - The text color attribute set in the <body> tag.
- FileCreatedDate - Use this value to show when the loaded HTML file was created
- fileModifiedDate - Use this value to show the last change date of the loaded HTML file
- lastModified - The date the file was modified last.
- layers - An array containing all the layers in a document.
- linkColor - The color of HTML links in the document. It is specified in the <body> tag.
- location –Full url of the document.
- referrer - The Universal Resource Locator (URL) of the document that we got the link to the present document from.
- title - The name of the current document as described between the header TITLE tags.
- URL - The location of the current document.
- vlinkColor - The color of visited links as specified in the <body> tag
- alinkColor - The color of activated links as specified in the <body> tag
- anchors - The array of object corresponding to each named anchor in the document
- applets - The array of object corresponding to each java applet included in the document

Document Object Methods

Method	Description
close()	Closes the output stream previously opened with document.open()
getElementById()	Accesses the first element with the specified id
getElementsByName()	Accesses all elements with a specified name
getElementsByTagName()	Accesses all elements with a specified tagname

open()	Opens an output stream to collect the output from document.write() or document.writeln()
write()	Writes HTML expressions or JavaScript code to a document
writeln()	Same as write(), but adds a newline character after each statement

Document Object Events

- onafterupdate
- onbeforeupdate
- onClick
- ondblclick
- ondragstart
- onerrorupdate
- onhelp
- onkeydown
- onkeypress
- onkeyup
- onmousedown
- onmousemove
- onMouseOut
- onMouseOver
- onmouseup
- onreadystatechange
- onrowenter
- onrowexit
- onselectstart

Other objects in Document Object Model

1] Form Object

- The Form object represents an HTML form.
- For each <form> tag in an HTML document, a Form object is created.
- Forms are used to collect user input, and contain input elements like text fields, checkboxes, radio-buttons, submit buttons and more. A form can also contain select menus, textarea, fieldset, legend, and label elements.
- Forms are used to pass data to a server.
- A separate instance of the form is created for each form in a document.
- **A form object can be accessed or referred in two ways:**

    - **Using position in the form array**

      document.forms[index]

    - **Using name of the form**

      Document.formname

- **Accessing individual elements and their attributes:**

    - All the controls used in a form are referred to as its elements.
    - **By using array**

      document.form[index].element.[index].attribute

    - **By using element name**

      document.form1.txt1.value

Form Object Collection

Collection	Description
elements[]	Returns an array of all elements in a form

Form Object Properties

Property	Description
acceptCharset	Sets or returns the value of the accept-charset attribute in a form
action	Sets or returns the value of the action attribute in a form

enctype	Sets or returns the value of the enctype attribute in a form
length	Returns the number of elements in a form
method	Sets or returns the value of the method attribute in a form
name	Sets or returns the value of the name attribute in a form
target	Sets or returns the value of the target attribute in a form

Form Object Methods

Method	Description
reset()	Resets a form
submit()	Submits a form

Form Object Events

Event	The event occurs when...
onreset	The reset button is clicked
onsubmit	The submit button is clicked