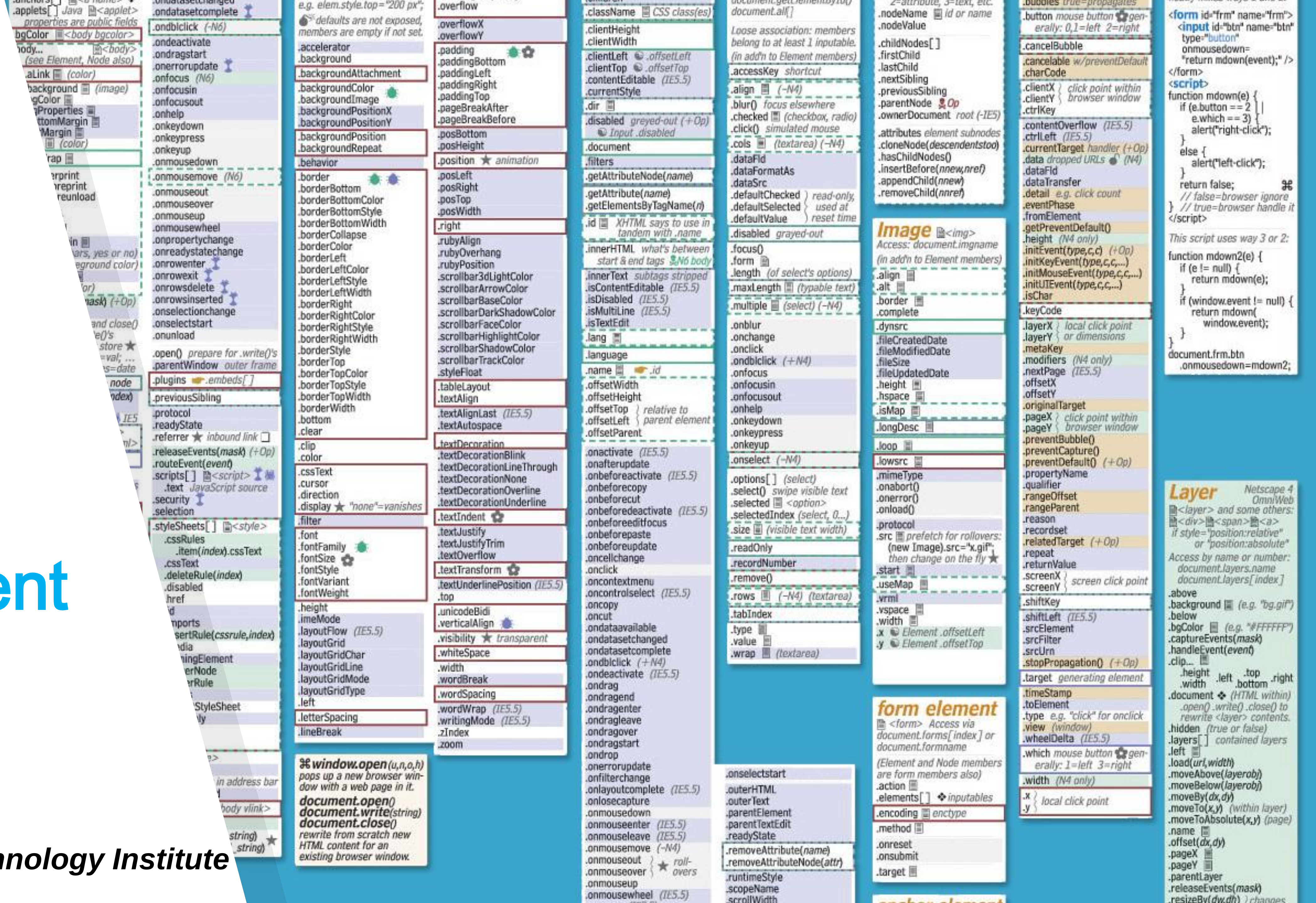
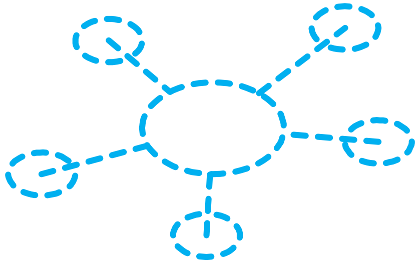


Document Object Model

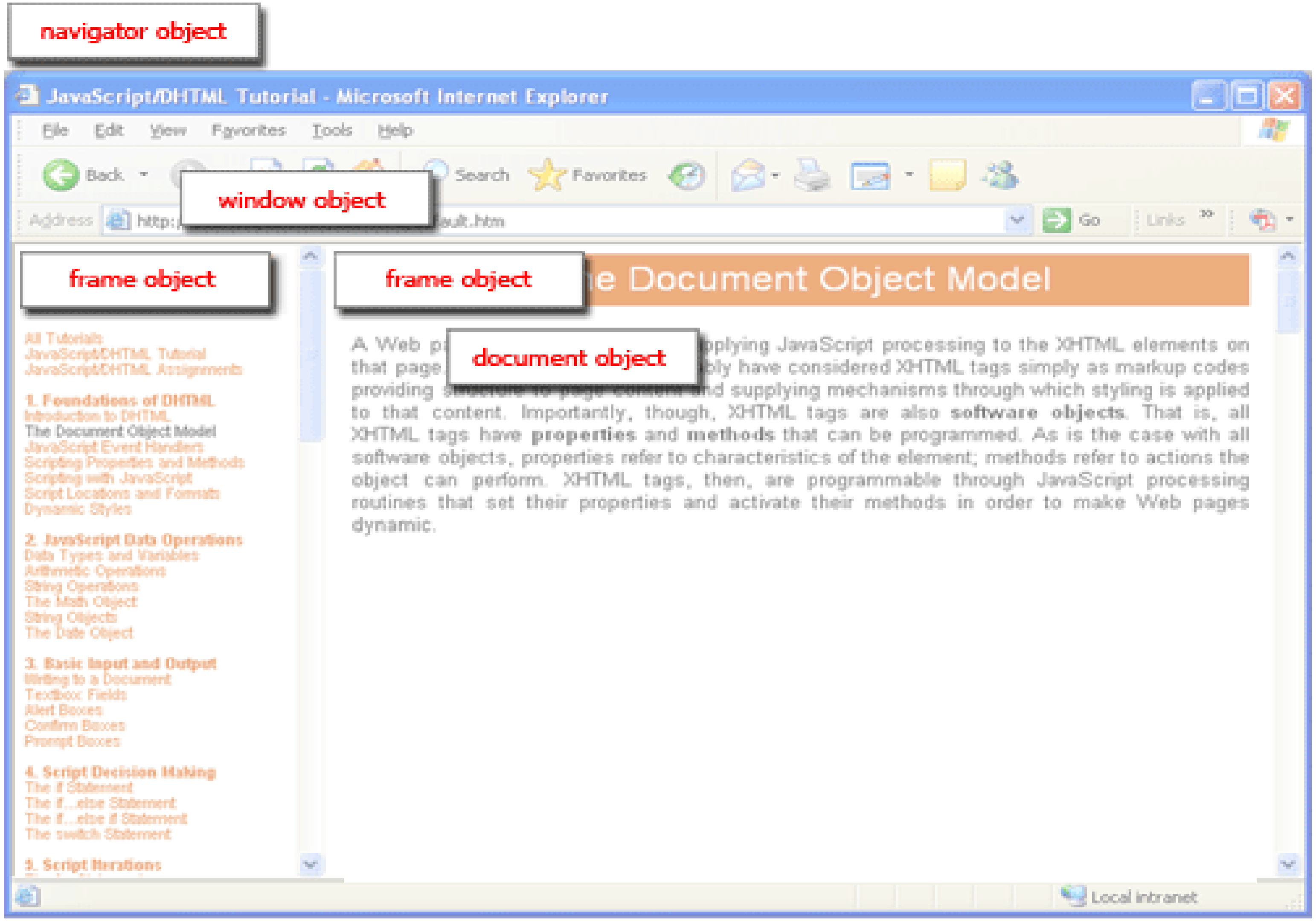
Eman Fathi
Information Technology Institute

WHAT IS DOM?

- Stands for Document Object Model.
- W3C standard.
- Defines a standard way to access and manipulate HTML documents.
- Platform independent.
- Language independent



THE DOM MODEL CONT.



All Tutorials
JavaScript/DHTML Tutorial
JavaScript/DHTML Assignments

1. Foundations of DHTML
Introduction to DHTML
The Document Object Model
JavaScript Event Handlers
Scripting Properties and Methods
Scripting with JavaScript
Script Locations and Formats
Dynamic Styles

2. JavaScript Data Operations
Data Types and Variables
Arithmetic Operations
String Operations
The Math Object
String Objects
The Date Object

3. Basic Input and Output
Writing to a Document
TextBox Fields
Alert Boxes
Confirm Boxes
Prompt Boxes

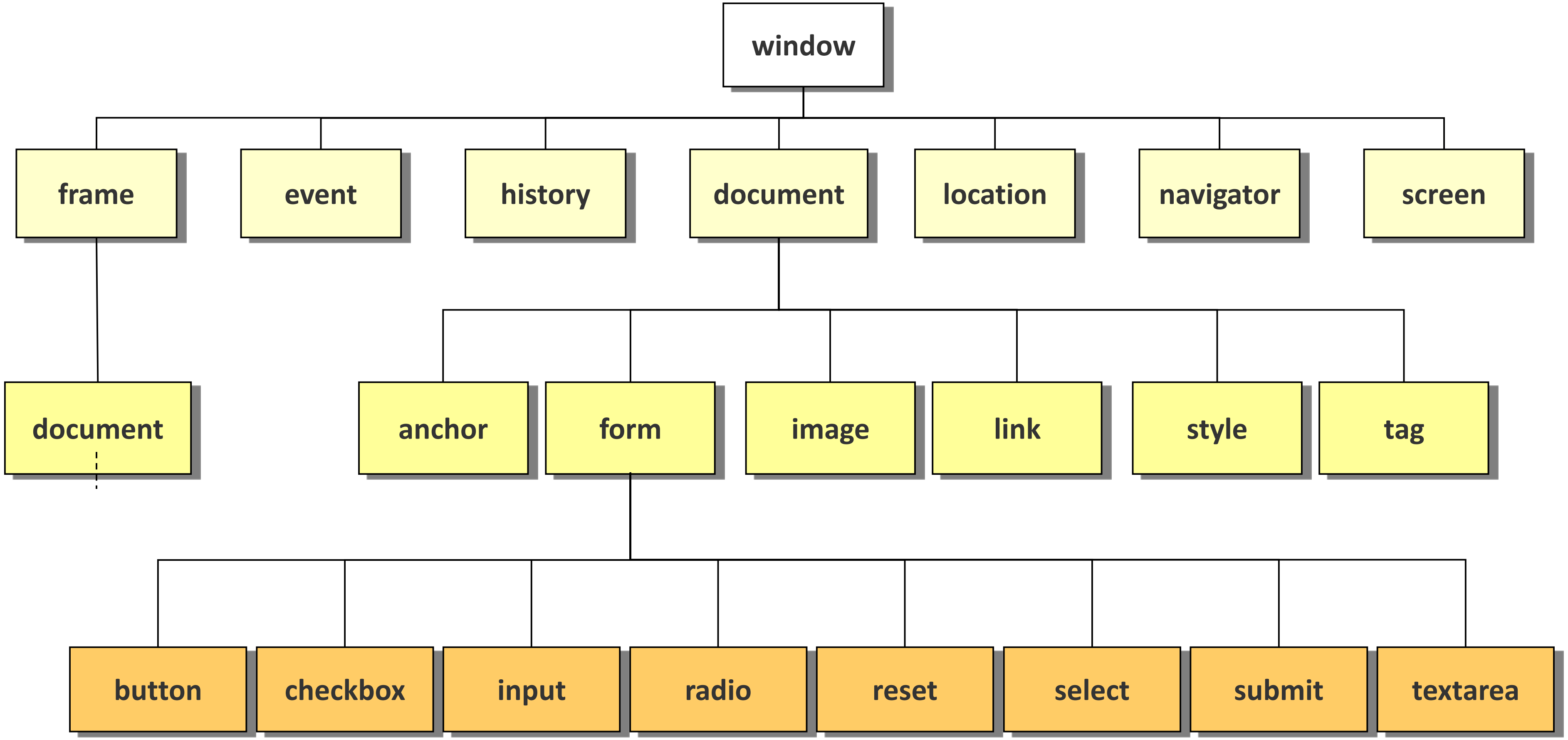
4. Script Decision Making
The if Statement
The if...else Statement
The if...else if Statement
The switch Statement

5. Script Iterations

e Document Object Model

A Web p pplying JavaScript processing to the XHTML elements on that page. bly have considered XHTML tags simply as markup codes providing structure to page content and supplying mechanisms through which styling is applied to that content. Importantly, though, XHTML tags are also **software objects**. That is, all XHTML tags have **properties** and **methods** that can be programmed. As is the case with all software objects, properties refer to characteristics of the element; methods refer to actions the object can perform. XHTML tags, then, are programmable through JavaScript processing routines that set their properties and activate their methods in order to make Web pages dynamic.

DOM TREE


window
frame
event
history
document
location
navigator
screen
document
anchor
form
image
link
style
tag
button
checkbox
input
radio
reset
select
submit
textarea

JAVASCRIPT OBJECTS HIERARCHY

Every page has the following objects:

- **window***:* the top-level object; has properties that apply to the entire window.
- **navigator***:* has properties related to the name and version of the Navigator being used.
- **document***:* contains properties based on the content of the document, such as title, background color, links, and forms.
- **location***:* has properties based on the current URL.
- **history***:* contains properties representing URLs the client has previously requested.

WINDOW

DEMO

- The top level object in the JavaScript client hierarchy.
- The Window object represents a browser window.
- Window object has a set of properties & methods.

- Object Model Reference:

  window

  To reference its properties & methods:

  - [window.]property
  - [window.]method

WINDOW

CONT.

Properties		
Name	**Description**	**Syntax**
name	Return or set a window's name	window.name

statusbar	Sets whether or not the browser's statusbar should be visible	Window.statusbar=0 // Not visible
		Window.statusbar=1// visible

closed	Specifies whether a window has been closed	window.closed
opener	Return details of the source window	window.opener
screenX screenY	read-only property returns the distance of the left border of the browser from the left side of the screen	window.screenX window.screenY
scrollX scrollY	Returns the number of pixels the document has been scrolled horizontally.	window.scrollX Window.scrollY

WINDOW

CONT.

Properties-cont		
Name	**Description**	**Syntax**
document	Reference to the current document object.	window.document
frames	An array referencing all of the frames in the current window.	window.frames[i]
history	Reference to the History object of JavaScript	window.history
navigator	Reference to the browser application	window.navigator
location	Reference to the Location object of JavaScript	window.location

WINDOW

CONT.

DEMO

Methods		
Name	**Description**	**Syntax**
alert()	Displays an alert box with a message and an OK button	window.alert(“Hello”)
confirm()	Displays a dialog box with a message and an OK, returning true, and a Cancel, returning false	Window.confrim(“Do you want to exit”)
prompt()	Displays a dialog box that prompts the user for input	name=prompt("Please enter your name","")
open()	Opens a new browser window	window.open(, ,)
close()	close a specified window	window.close()

WINDOW

CONT.

DEMO

Methods-cont		
Name	**Description**	**Syntax**
moveTo(h,v)	Moves the window to horizontal and vertical position relative top-left of screen:	window. moveTo(,)
moveBy(h,v)	Moves the window by + or - horizontal and vertical pixels:	window.moveBy(,)
resizeTo(h,v)	Changes the size of the window to horizontal and vertical number of pixels:	window.resizeTo(,)
resizeBy(h,v)	Changes the size of the window + or – current size	window.resizeBy(,)
scrollTo(h,v)	Scrolls the document in the current window or frame to horizontal and vertical pixel positions from top of document:	window.scrollTo(,)
scrollBy(h,v)	Scrolls the document in the current window or frame by + or - horizontal and vertical pixel from current position:	window.scrollBy(,)

WINDOW

CONT.

DEMO

OPEN METHOD

Return Value: A reference to the newly created window, or null if the call failed.

```
var myWindow = window.open("", "", "width=200,height=100");

var myWindow = window.open("", "MsgWindow", "width=200,height=100");

myWindow.document.write("<p>This is 'MsgWindow'. I am 200px wide and 
100px tall!</p>");
```

WINDOW

CONT.

DEMO

resizable yes | no | 1 | 0	Whether or not the window is resizable. IE only
scrollbars yes | no | 1 | 0	Whether or not to display scroll bars. IE, Firefox & Opera only
menubar yes | no | 1 | 0	Whether or not to display the menu bar
status yes | no | 1 | 0	Whether or not to add a status bar
titlebar yes | no | 1 | 0	Whether or not to display the title bar. Ignored unless the calling application is an HTML Application or a trusted dialog box
toolbar yes | no | 1 | 0	Whether or not to display the browser toolbar. IE and Firefox only
top=pixels	The top position of the window. Negative values not allowed
width=pixels	The width of the window. Min. value is 100
height=pixels	The height of the window. Min. value is 100

WINDOW TIMERS

WINDOW

CONT.

DEMO

Methods-cont		
Name	**Description**	**Syntax**
setInterval()	Evaluates an expression at specified intervals	intervalID = window.setInterval (*exp,time_interval*)
setTimeout()	Evaluates an expression after a specified number of milliseconds	timeoutID = window.setTimeOut (*exp, time_interval*)
clearTimeout()	Clears the delay set by setTimeout()	window.clearTimeout(timeoutID)
clearInterval()	Cancels repeated action which was set up using setInterval()	window.clearInterval(intervalID)

20 min
Break.

NAVIGATOR

- The navigator object represents the browser application.

- This object allows scripts to see:
  1. browser type
  2. browser version

- Object Model reference:
  [window.]**navigator**

- All of its properties are read-only.

NAVIGATOR CONT.

- **Proterties**
  - appName
  - appVersion
  - Platform
  - browserLanguage

- **Methods**
  - javaEnabled()

FRAME

- Created by using the HTML **<FRAME>** tag in a Window that contains the **<FRAMESET>** tag.
- A frame is an independent window within a parent window.
- Each frame has its own URL and is treated, *with a few exceptions*, as a Window object by JavaScript.
- Frame object has all the methods and properties of a window object.
- window.frames[i] → An array containing references to all the named child frames in the current window.
- To access a specific frame in a window:
  - window.**frames**[0]
  - window.**frames**[“frame_name”]

LOCATION

- The Location object is part of a Window object.
- The location Object refers to the current URL.

- Object Model Reference:
  [window.]**location**

- ***href*** is the defult property of the location object.
  location="documentURL"
  location.**href**="documentURL"

- ***replace*** method loads the specified URL over the current history entry.
  location.replace(URL)

HISTORY

- The history Object lets you send the user to somewhere in the history list from within a JavaScript program.
- The history Object is an Array of URLs. Like all Arrays, it has a **length** Property.

- Object Model reference:
  [window.]**history**

- Properties
  - length

- Methods
  - back()
  - forward()
  - go()

DOM EVENTS

EVENTS

- Events are actions that respond to user's specific actions.
- Events are controlled in JavaScript using event handlers that indicate what actions the browser takes in response to an event.
- Event handlers are created as **attributes** added to the HTML tags in which the event is triggered.
- An Event handler adopts the **event** name and appends the word "***on***" in front of it.

  < tag onevent = "JavaScript commands;">

- Thus the "click" event becomes the *onClick* event handler

EVENTS CONT.

Mouse Events	
Event handler	**Description**
onMouseDown	when pressing any of the mouse buttons.
onMouseMove	when the user moves the mouse pointer within an element.
onMouseOut	when moving mouse pointer out of an element.
onMouseUp	when the user release any mouse button pressed
onMouseOver	when the user moves the mouse pointer over an element.
onClick	when clicking the left mouse button on an element.
onDblClick	when Double-clicking the left mouse button on an element.
onDragStart	When the user has begun to select an element

EVENTS CONT.

Keyboard Events	
Event handler	**Description**
onKeyDown	When User presses a key
onKeyPress	When User holds down a key
onKeyUp	When User a key

EVENTS CONT.

Other Events	
Event handler	**Description**
onAbort	The User interrupted the transfer of an image
onBlur	when loosing focus
onFocus	when setting focus
onChange	when the element has lost the focus and the content of the element has changed
onLoad	a document or other external element has completed downloading all the data into the browser
onUnload	a document is about to be unloaded from the window
onError	When an error has occurred in a script.
onMove	when moving the browser window

EVENTS CONT.

Other Events	
Event handler	**Description**
OnReset	When the user clicks the form reset button
onSubmit	When the user clicks the form submit button
onScroll	When the user adjusts an element’s scrollbar
onResize	When the user resizes a browser window
onHelp	When the user presses the F1 key
onselect	When selecting text in an input or a textarea element
onStart	When A marquee element loop begins
onFinish	When a marquee object finishes looping
onSelectStart	When the user is beginning to select an element

BINDING EVENTS

DEMO

- Binding Event Handlers to Elements can be:
  - Event handlers as tag attribute
  - event handlers as object property

EVENT HANDLERS AS OBJECT PROPERTY

```
<script>
function showAlert ()
{
alert(“you have clicked me”)
}
document.forms[0].b1.onclick = showAlert
</script>
```

Note: there are no parentheses

EVENT HANDLERS AS TAG ATTRIBUTE

- <a href=“1.htm” onclick=“myFunc();return false”>

This will make the browser ignore the action of href

EVENT OBJECT

Properties of the event Object in IE

- shiftKey , altKey , ctrlKey
- button “Mouse Button”
- clientX, clientY
- fromElement
- cancelBubble

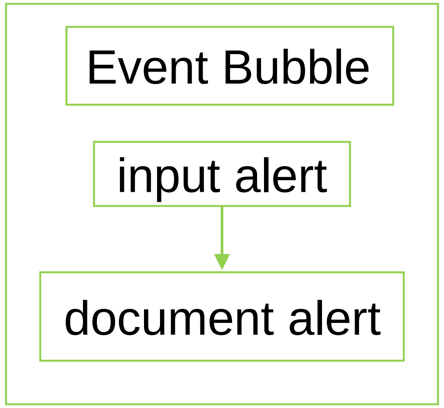
EVENT PROPAGATION

```
<script>
        function tryme()
        {
                alert(“document")
        }

        document.onclick = tryme
</script>

<body leftmargin=0 topmargin=0 >

<input type=text onclick="alert(‘input')">
```



Event Bubble

input alert

document alert

→ **if we made a click in the textbox**

EVENT BUBBLING

- Event Bubbling

→ Events in Internet Explorer bubble up from the target element through an element hierarchy.

- To cancel the Bubbling.

    event.stopPropagation()

Thank You