Client Side Technologies

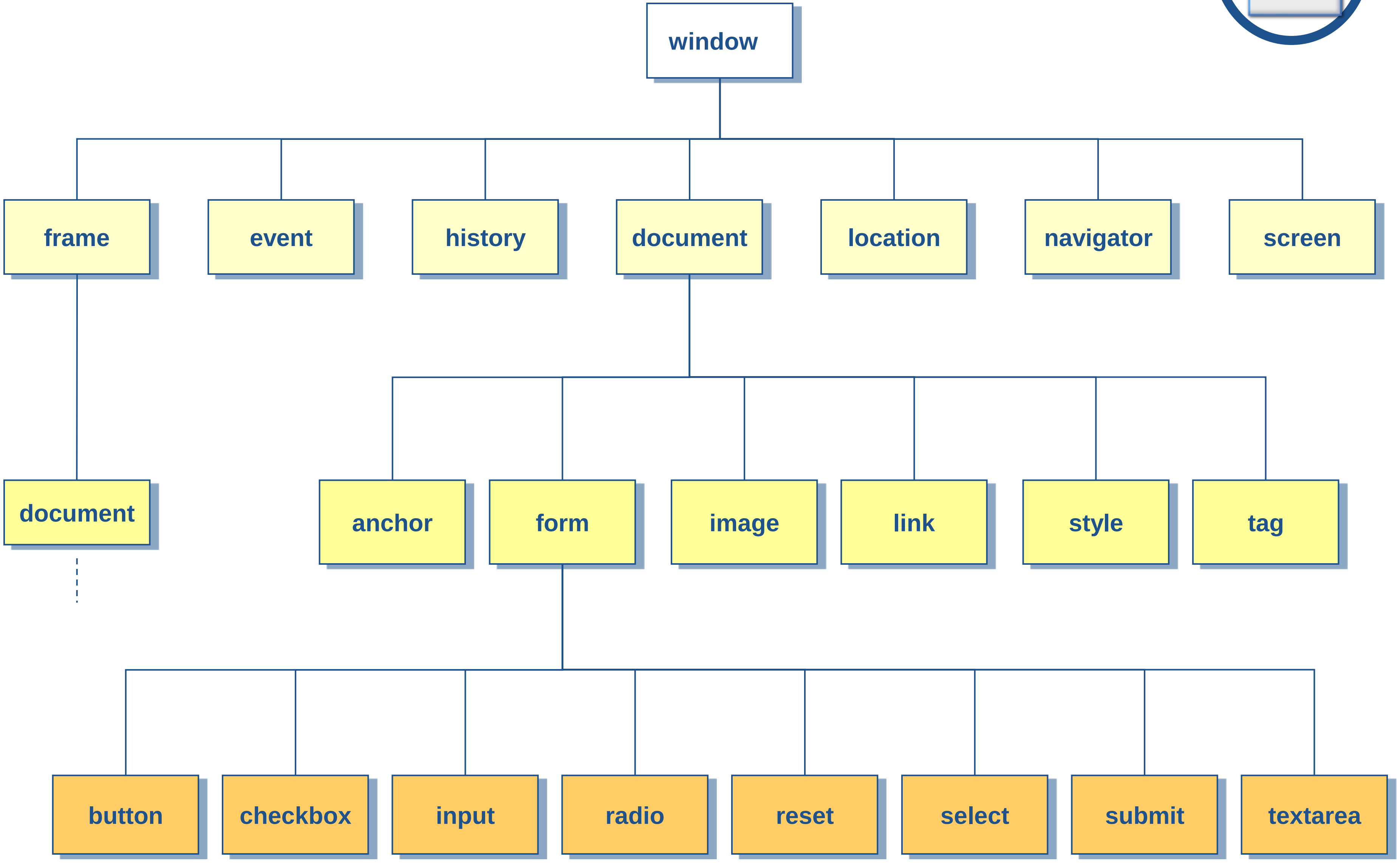
Browser Object Model (BOM)

ITI – Assiut Branch
Eng. Hany Saad

What is DOM?

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

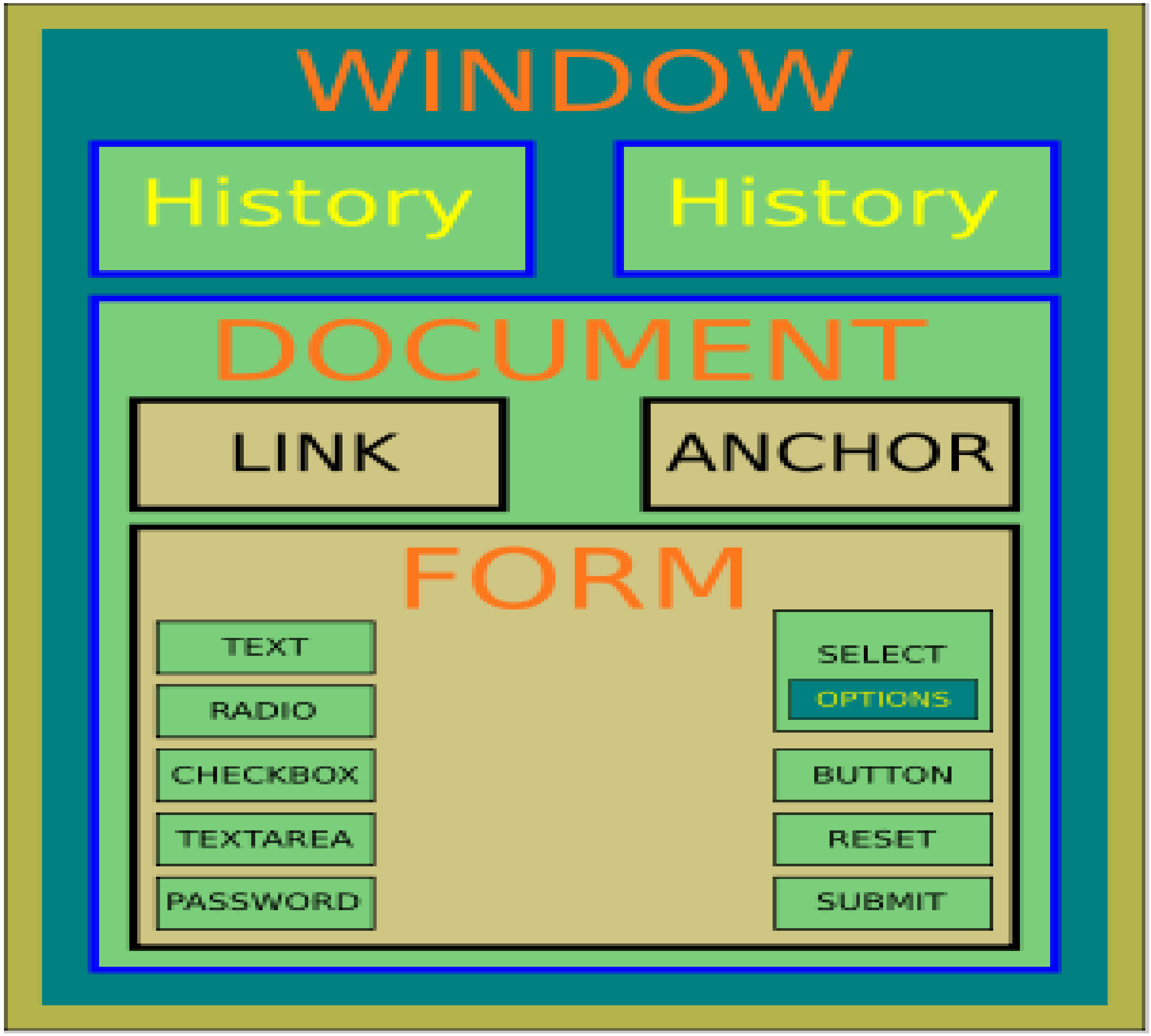
DOM Model


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

JavaScript Object 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.
  - **screen** : contains information about the visitor's screen.

JavaScript Object Hierarchy(Cont.)



What is BOM?

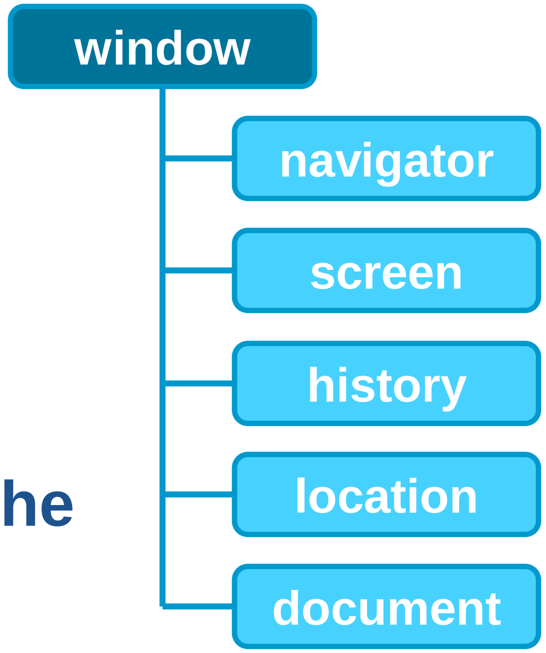
❑BOM Stands for Browser Object Model.

❑BOM covers objects which relate to the browser.

❑At the top of the BOM hierarchy is window object. Below that comes the

- navigator
- screen
- history
- location
-document objects (in no particular order).

❑Each object below the window is of equal status, they all relate directly to the window object.



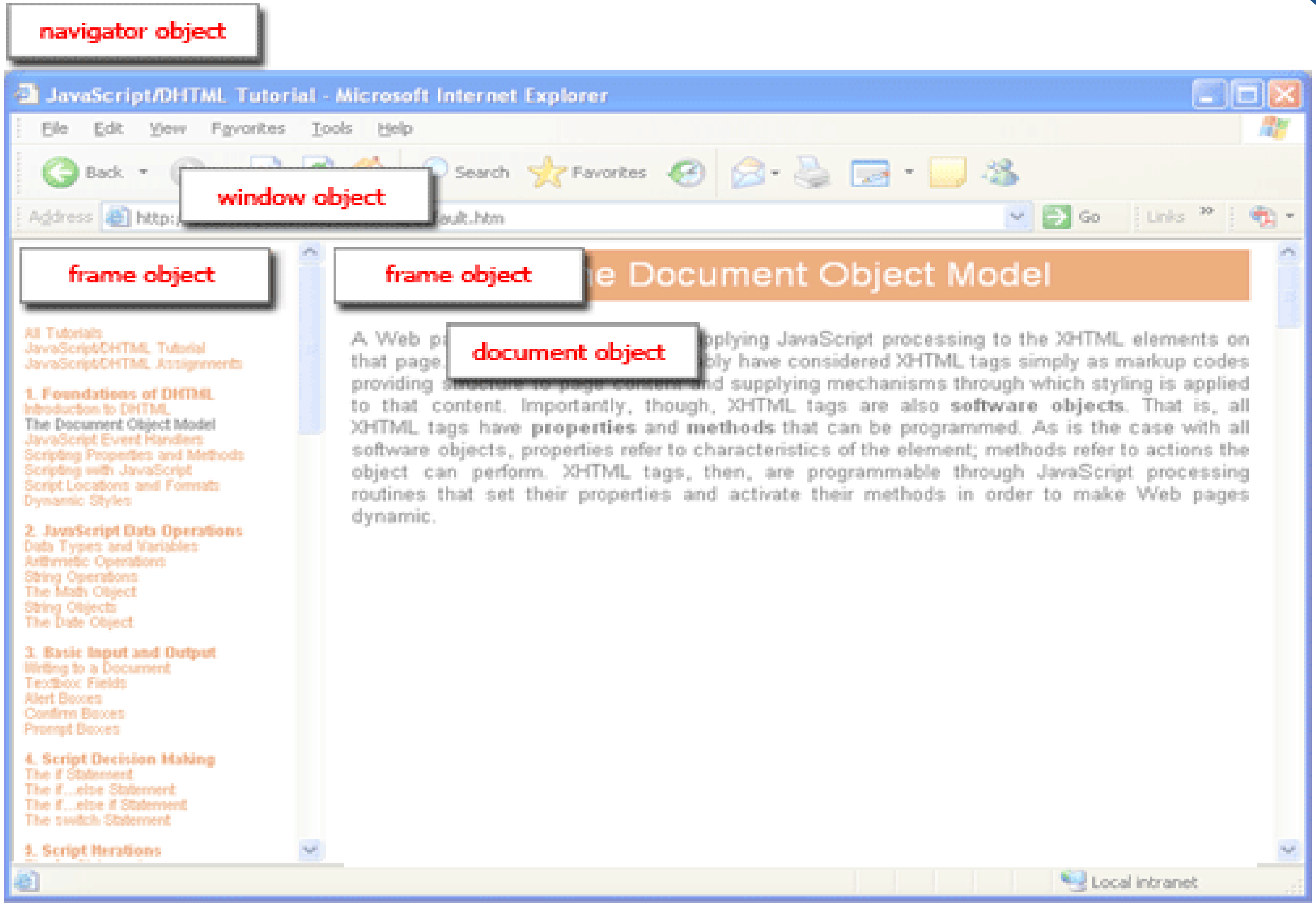
What is BOM? (Cont.)

- ❑Using the BOM, developers can move the window, change text in the status bar, and perform other actions that do not directly relate to the page content.

- ❑For some reason, the Browser Object Model is generally not referred to by its proper name. More often, it's usually wrapped up with the DOM.

- ❑In actuality, the DOM, which relates to all things pertaining to the document, resides *within* the BOM.

BOM Model



e Document Object Model

A Web p… applying JavaScript processing to the XHTML elements on that page. …bly have considered XHTML tags simply as markup codes providing … nd 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.

Window

- The top level object in the JavaScript object 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
innerHeight	Returns the inner Height of a window's content area	window.innerHeight
innerWidth	Returns the inner width of a window's content area	window.innerWidth
outerHeight	Returns the outer height of a window, including toolbars/scrollbars	window.outerHeight
outerWidth	Returns the outer width of a window, including toolbars/scrollbars	window.outerWidth

Window(Cont.)

❑ Properties:

Name	Description	Syntax
screenLeft **=** **screenX**	Returns the horizontal coordinate of the window relative to the screen	window.screenLeft
screenTop **=** **screenY**	Returns the vertical coordinate of the window relative to the screen	window.screenY
pageXOffset **=** **scrollX**	Returns the pixels the current document has been scrolled (horizontally) from the upper left corner of the window	window.pageXOffset
pageYOffset **=** **scrollY**	Returns the pixels the current document has been scrolled (vertically) from the upper left corner of the window	window.pageYOffset

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]
frameElement	Returns the <iframe> element in which the current window is inserted	window.frameElement;
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

- **More Properties:**

http://www.w3schools.com/jsref/obj_window.asp

Window(Cont.)

❑ **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 http://www.w3schools.com/jsref/met_win_open.asp	window.open(url, name, properties)
close()	close a specified window	window.close()

Window(Cont.)

❑ Methods (Cont.):

Name	Description	Syntax
focus()	Sets focus to the current window	window.focus();
blur()	Removes focus from the current window	window.blur()
getSelection()	Returns a Selection object representing the range of text selected by the user	window.getSelection();
stop()	Stops the window from loading	window.stop();
print()	Print the contents of the specified window.	window.print()

Window(Cont.)

❑ 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 by + or - horizontal and vertical pixels:	window.resizeBy(,)
scrollTo(h,v)	Scrolls the document in the current window or frame to horizontal and vertical pixel postions 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.)

❑ Methods(Cont.):

Name	Description	Syntax
setInterval(expression, interval)	Evaluates an expression at specified intervals	var t= window.setInterval(*funcName,500*)
clearInterval(interval_Obj_Name)	Used to clear a time interval set using the above method	Window.clearInterval(t)
setTimeout()	Used to execute an expression or function after a time interval (in millisecond).	window.setTimeOut(*exp*, *time_interval*)
clearTimeout()	Used to clear a timeout set using the above method	

Screen

❑The screen object provides information about the desktop outside the browser .

❑This object allows scripts to:

- Detect the browser’s usable area.
- Return information on the display screen's dimensions and color depth

❑Object Model reference:

[window.]screen

Screen(Cont.)

❑ **Properties:**

Name	Description
availHeight	Returns the height of the screen (excluding the Windows Taskbar)
availWidth	Returns the width of the screen (excluding the Windows Taskbar)
colorDepth	Returns the bit depth of the color palette for displaying images
height	Returns the total height of the screen
pixelDepth	Returns the color resolution (in bits per pixel) of the screen
width	Returns the total width of the screen

Navigator

- ❑ **The navigator object represents the browser application.**
- ❑ **This object allows scripts to see:**
  - browser type
  - browser version
- ❑ **Object Model reference:**

```
[window.]navigator
```

- ❑ **All of its properties are read-only.**

Navigator

❑ Properties:

Name	Description	Syntax
appName	get the name of the browser	navigator.appName
appVersion	get the version of the browser	navigator.appVersion
language	get the language of the browser	navigator.language
cookieEnabled	returns whether the browser allows cookies or not	navigator. cookieEnabled
platform	return the name of the OS	navigator.platform
onLine	Determines whether the browser is online	navigator.online
geolocation	Returns a Geolocation object that can be used to locate the user's position	navigator.geolocation

❑ Methods:

- javaEnabled()

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.

```
Var url=location.href;
```

- replace method loads the specified URL over the current history entry.

```
location.replace(URL)
```

Location (Cont.)

❑ **Properties:**

Name	Description
href	Sets or returns the entire URL
hash	Sets or returns the anchor part (#) of a URL
search	Sets or returns the querystring part of a URL

❑ **Methods:**

Name	Description
replace(URL)	Replaces the current document with a new one
assign(URL)	almost the same as replace method. The difference is that it creates an entry in the browser's history list, while replace() doesn’t
reload()	Reloads the current document

History

- The history Object is an Array of URLs.
- The history Object lets you send the user to somewhere in the history list from within a JavaScript program.
- **Object Model reference:**

[window.]history

- **Properties:**
  - length
- **Methods:**

back()	forward()	go(index)

Event object

❑ Event object:

- HTML DOM events allow JavaScript to register different event handlers on elements in an HTML document.
- Events are normally used in combination with functions, and the function will not be executed before the event occurs (such as when a user clicks a button).
- When an event occurs an object of Event is created to hold some additional information about the occurred event.
- Dom Events Reference:
  - http://www.w3schools.com/jsref/dom_obj_event.asp
  - https://developer.mozilla.org/en-US/docs/Web/API/Event
  - https://developer.mozilla.org/en-US/docs/Web/API/UIEvent
  - https://developer.mozilla.org/en-US/docs/Web/API/MouseEvent
  - https://developer.mozilla.org/en-US/docs/Web/API/KeyboardEvent
  - https://developer.mozilla.org/en-US/docs/Web/API/WheelEvent

Event object properties

❑ Event object properties

Property	Description
Target	The element that fired the event (IE old versions = srcElement)
type	Type of event
timeStamp	Returns the time in milliseconds (A Number, representing the number of milliseconds since midnight of January 1, 1970) at which the event was created
bubbles	Returns whether or not a specific event is a bubbling event
cancelable	Returns whether or not an event can have its default action prevented

MouseEvent object properties

❑ MouseEvent object properties

Property	Description
screenX	Returns the horizontal coordinate of the mouse pointer, relative to the screen, when the mouse event was triggered
screenY	Returns the vertical coordinate of the mouse pointer, relative to the screen, when the mouse event was triggered
clientX	Returns the horizontal coordinate of the mouse pointer, relative to the current window, when the mouse event was triggered
clientY	Returns the vertical coordinate of the mouse pointer, relative to the current window, when the mouse event was triggered
pageX **(New, not supported in all browsers)**	Returns the horizontal coordinate of the mouse pointer, relative to the document, when the mouse event was triggered
pageY **(New, not supported in all browsers)**	Returns the vertical coordinate of the mouse pointer, relative to the document, when the mouse event was triggered
offsetX **(New, not supported in all browsers)**	the horizontal coordinate of the mouse pointer relatively to the target element
offsetY **(New, not supported in all browsers)**	the vertical coordinate of the mouse pointer relatively to the target element

MouseEvent object properties (Cont.)

❑ MouseEvent object properties (Cont.)

Property	Description
altKey	True if the alt key was also pressed
ctrlKey	True if the alt key was also pressed
shiftKey	True if the alt key was also pressed
detail	Returns a number that indicates how many times the mouse was clicked
button	Any mouse buttons that are pressed . Possible values: 0 : Left mouse button 1 : Wheel button or middle button (if present) 2 : Right mouse button Note: Internet Explorer 8 and earlier has different return values: 1 : Left mouse button 2 : Right mouse button 4 : Wheel button or middle button (if present)
movementX / movementY	The X or Y coordinate of the mouse pointer relative to the position of the last mousemove event.

keyboardEvent object properties

❑ KeyboardEvent object properties (Cont.)

Property	Description
altKey	True if the alt key was also pressed
ctrlKey	True if the alt key was also pressed
shiftKey	True if the alt key was also pressed
code	Returns String with the code value of the key represented by the event.
key	Returns String representing the key value of the key represented by the event.
which **(Deprecated, use key instead)**	Returns a Number representing a system dependent numeric code identifying the value of the pressed key; this is usually the same as keyCode. (For IE 8 and ealier use keyCode property instead) **Both the which and keyCode properties are deprecated and provided for compatibility only.** **The latest version of the DOM Events Specification recommend using the key property instead.**

Event object methods

❑ Event object Methods

Property	Description
preventDefault()	Cancels the event if it is cancelable, meaning that the default action that belongs to the event will not occur. For example, this can be useful when: • Clicking on a "Submit" button, prevent it from submitting a form • Clicking on a link, prevent the link from following the URL Note: Not all events are cancelable. Use the cancelable property to find out if an event is cancelable.
stopPropagation()	Prevents further propagation of the current event.

Event - preventDefault

- Event handlers return value:

```
<a href="#" onclick="myFunc();return false"> click me </a>
```

This will make the browser ignore the action of href

- Another way that can also make the browser ignore the action of href is:

```
<a href="javascript: void(0)" onclick="alert('hi')" >click me</a>
```

- Or use preventDefault() method

```
<a href="#" onclick="myFunc();event.preventDefault()"> click me </a>
```

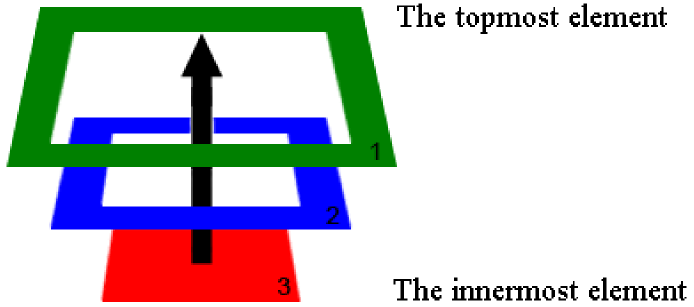
Event propagation (Bubbling):

❑ Event propagation (Bubbling):

- DOM elements can be nested inside each other. And somehow, the handler of the parent works even if you click on it’s child.
- The reason is event bubbling.
- After an event triggers on the deepest possible element, it then triggers on parents in nesting order.

```
<script>
        function tryme()
        {
            alert(“document")
        }
        document.onclick=tryme;
</script>
<body leftmargin=0 topmargin=0 >
<input type=text onclick="alert(‘input')">
```

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



Event propagation (Cont.)

❑ **Cancel Event Bubbling:**

```
<input type="text" onclick="handlerFun(Event)"/>
```

```
function handlerFun(e)
{
        //For most browsers (W3C-compliant browsers)
        e. stopPropagation();
        //For IE <9
         e.cancelBubble = true


}
```

Event handling

❑ Binding Events :

- Binding Event Handlers to Elements can be:
    1. Event handlers as tag attribute
    2. Event handlers as object property

Event handling (Cont.)

1. Event handlers as tag attribute:

```
<input type=button value="click me" name=b1 onclick="alert('you
        have made a click')">
OR
        <script>
        function showmsg()
        {
            alert("you have made a click")
        }
        </script>

<input type=button value="click me" onclick="showmsg()" />
```

Event handling (Cont.)

2. Event handlers as object property:

```
<body>
        <form>
            <input type="button" name='b1' value="Click ME" />
        </form>
</body>
<script>
        function showAlert ()
        {
            alert("you have clicked me")
        }
        document.forms[0].b1.onclick=showAlert;
        //or
        document.getElementByName("b1").onclick=showalert;
</script>
```

Note: there are no parentheses

addEventListener() method

- The addEventListener() method attaches an event handler to the specified element.
- The addEventListener() method attaches an event handler to an element without overwriting existing event handlers.
- The addEventListener() method makes it easier to control how the event reacts to bubbling.
- When using the addEventListener() method, the JavaScript is separated from the HTML markup, for better readability and allows you to add event listeners even when you do not control the HTML markup.

addEventListener() method (cont.)

- Syntax:

```
element.addEventListener(event, function, [useCapture]);
```

```
document.getElementById(“b1”).addEventListener("click", myFunction);


function myFunction() {
        alert (“Button Clicked");
}
```

- The first parameter is the type of the event (like "click" or "mousedown").
- The second parameter is the function we want to call when the event occurs.
- The third parameter (optional parameter): is a boolean value specifying whether to use event bubbling or event capturing. Possible values:
  - true - The event handler is executed in the capturing phase
  - false- Default, the event handler is executed in the bubbling phase

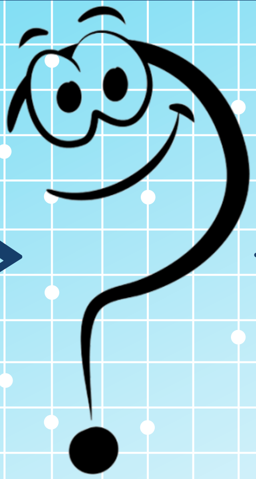
addEventListener() method (cont.)

- You can easily remove an event listener by using the removeEventListener() method.

```
element.removeEventListener("mousemove", myFunction);
```

- Note: The addEventListener() and removeEventListener() methods are not supported in IE 8 and earlier versions and Opera 6.0 and earlier versions. However, for these specific browser versions, you can use the attachEvent() method to attach an event handlers to the element, and the detachEvent() method to remove it.

```
element.attachEvent(event, function);
element.detachEvent(event, function);
```

<SCRIPT >  </SCRIPT>

<script>document.writeln("Thank You!")</script>