

JavaScript, Seventh

Chapter 5

The Browser Document Object Model

©Professor Tom DeDonno

Outline

- Designing a Lightbox Web Application
- Browser Object Model
- Window Objects
- History, Location, & Navigator Object
- Frames within Windows

©Professor Tom DeDonno

CIMW 160 – JavaScript – Chapter 4

2

Document Object Model

Chapter 5: Creating a Web App
Using Document Object Model
BOM: Browser Object Model

Learning Objectives

1. Understand the principles of building a web app.
2. Create nodes and append them to a web document.
3. Add, delete, copy, & move Web document element nodes.
4. Implement set intervals for repeating a timed command block
5. Create system dialog boxes that receive user input.
6. Open and configure a browser popup window.
7. Create an overlay that lies on top of a web document.
8. Work with the objects within the Browser Object Model.

©Professor Tom DeDonno

Web Application Background

Lightbox

- Modeless Dialog Panel Image Viewer
- Sample Web Application for Chapter 5

Web Application Components

- Strict Separation of:
 - HTML + CSS + JavaScript + Data
 - HTML Markup/ Container for App "Known Structure"
 - CSS is Presentation/Appearance "CSS Applied to Structure"
 - JavaScript – Action – Behavior (Operation) code—manages
 - Data (Integrated in Web Page, SQL Data, Flat Data)

Various Systems (Multi-tier Server) and/or
People Maintain Components

©Professor Tom DeDonno

CIMW 160 – JavaScript – Chapter 4

5

Figure 5-1 Light Box H11 Example

Large collection of image viewers.

Slider slides to next image, Lightbox pops modal dialog of Image



©Professor Tom DeDonno

CIMW 260 – JavaScript – Chapter 4

6

Figure 5-1 Light Box H11 Example

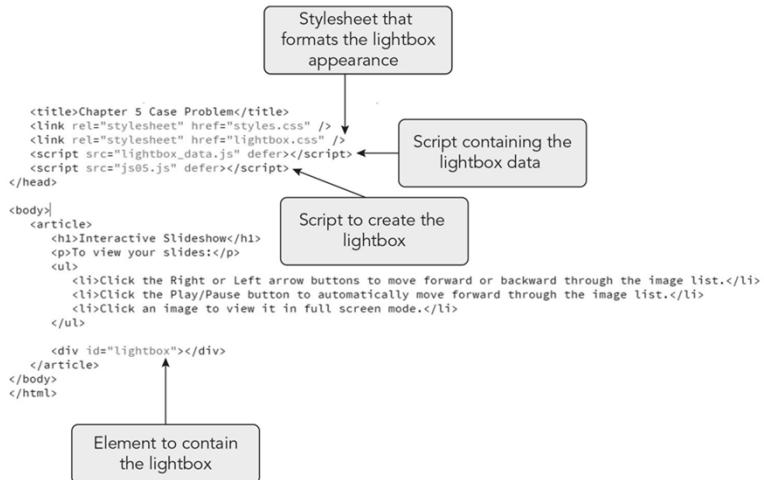
1. Page 164 Get, Copy, rename _txt Files
2. Goto your data folders/chapter05/chapter/js05 adn copy folder to your site root
3. Open js05_txt.html adn js05_txt.js and save as js05.html adn js05.js
4. In header, add links to local files lightbox.css, lightbox_data.js and js05.js, both JavaScript files are defer - we have been putting local js at end
- 5. Make sure all variables have been defined.
Replace imageCount with imgFiles.length**
6. Insert <div id="lightbox" ></div> before </article>
7. Analyze code, and Save

©Professor Tom DeDonno

CIMW 160 – JavaScript – Chapter 6

7

Lightbox HTML File



8

lightbox-data.js

```
// Title of the slideshow
let lightboxTitle = "My Western Vacation";

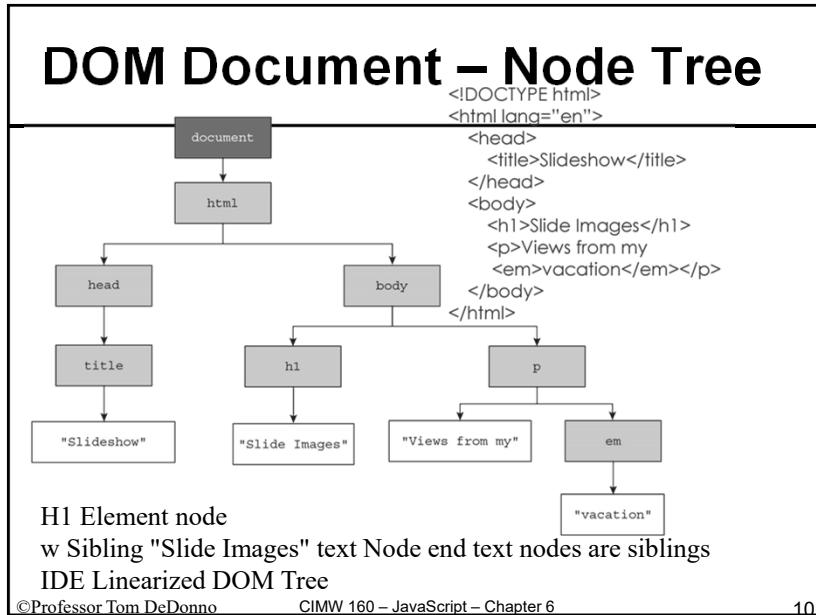
// Names of the image files shown in the slideshow
let imgFiles = ["photo01.jpg", "photo02.jpg", "photo03.jpg", "photo04.jpg",
    "photo05.jpg", "photo06.jpg", "photo07.jpg", "photo08.jpg",
    "photo09.jpg", "photo10.jpg", "photo11.jpg", "photo12.jpg"]

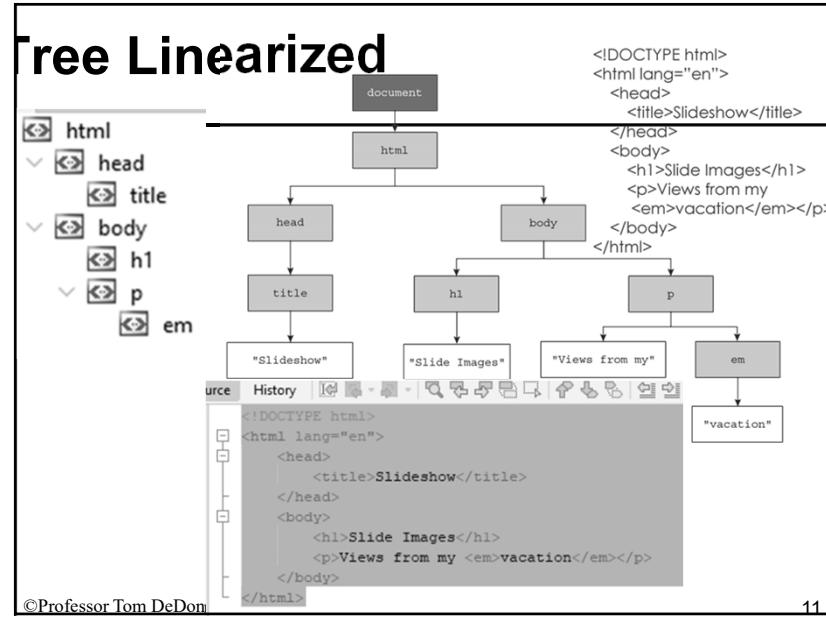
// Captions associated with each image
let imgCaptions = new Array(12);
imgCaptions[0] = "Sky Pond (Rocky Mountain National Park)";
imgCaptions[1] = "Buffalo on the Plains (South Dakota)";
imgCaptions[2] = "Garden of the Gods (Colorado Springs)";
imgCaptions[3] = "Elephant Head Wild Flower (Rocky Mountain National Park)";
imgCaptions[4] = "Double Rainbow (Colorado National Monument)";
imgCaptions[5] = "Moose in the Wild (Grand Lake, Colorado)";
imgCaptions[6] = "Camas Wild Flower (Rocky Mountain National Park)";
imgCaptions[7] = "Chasm Lake (Rocky Mountain National Park)";
imgCaptions[8] = "Teton Crest Trail (Grand Teton National Park)";
imgCaptions[9] = "The Notch Trail (Badlands National Park)";
imgCaptions[10] = "Sprague Lake (Rocky Mountain National Park)";
imgCaptions[11] = "Longs Peak Trail (Rocky Mountain National Park)";

// Count of images in the slideshow
let imgCount = imgFiles.length;
```

9

DOM Document – Node Tree





DOM - Nodes

- DOM Display is Built in (Linearized tree)
 - NetBeans Window → Navigator "DOM+"
 - Dreamweaver Window → DOM
 - Browser Inspect Inspector
- Class Window is BOM
- Class Document is DOM
- document.write rewrites DOM
 - Initially document.write, alert, console.log is easier
 - Best Practice Modify the DOM never rewrite it
 - Selecting Elements and/or Creating new nodes is difficult

js05_txt.html x Navigator

CSS

HTML

html

head

meta

meta

title

link

body

article

h1

p

ul

div id=lightbox

©Professor Tom DeDonno CIMW 160 – JavaScript – Chapter 6 12

DOM Node Types

Node: any an entity

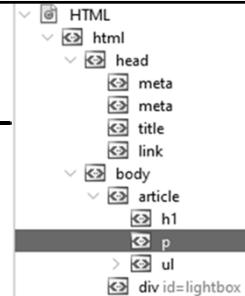
- element, attribute, comment, processing instruction, or text

Node tree: node hierarchical Nodes

Root node: top parent node

Often Navigate (Sizzle/DOM) using familial relationships

- Top Node Root Tree the Parent Note (HTML)
- Sizzle jQuery, CSS, JavaScript document.querySelector
 - html p { space descendant of html }
 - h1 + p { p + is sibling of H1}
 - html > body > article > p { p is a child > of article}
 - body>article:nth-child(2);



©Professor Tom DeDonno

CIMW 160 – JavaScript – Chapter 6

13

JavaScript Node Navigation

JavaScript Properties	Description – <u>element, text or comment</u>
<code>node.firstChild</code>	The first child of <code>node</code>
<code>node.lastChild</code>	The last child of <code>node</code>
<code>node.childNodes</code>	A node list of all direct children of <code>node</code>
<code>node.previousSibling</code>	sibling before <code>node</code> on the same level in the node tree
<code>node.nextSibling</code>	sibling after <code>node</code> on the same level in the node tree
<code>node.ownerDocument</code>	The root node of the document
<code>node.parentNode</code>	The parent of <code>node</code>

These properties pick next node

- Which can be HTML Element tag w Text or just Text

©Professor Tom DeDonno

CIMW 160 – JavaScript – Chapter 6

14

Navigation js05.html Nodes

- `pHC = document.getElementsByTagName("p");`
 - Returns an `HTMLCollection` –
 - `pHC[0]` is a node
- `p = document.querySelector("p");`
 - Returns node; `p === pHC[0]`
- Inspect Open node, or pause IDE will list properties/methods
- `p.parentNode` "parent" `<article>`
- `p.parentNode.parentNode` "GP" `body`
- `p.nextSibling` is text node
- `p.nextElementSibling` returns element ``

©Professor Tom DeDonno

CIMW 160 – JavaScript – Chapter 6



15

Navigation Element (Tag w Text)

Expression	Description
<code>n.Children</code>	A node list of all elements which are direct children of <code>node</code>
<code>n.firstElementChild</code>	The first element within <code>node</code>
<code>n.lastElementChild</code>	The last element within <code>node</code>
<code>n.previousElementSibling</code>	The sibling element immediately prior to <code>node</code> in the node tree
<code>n.nextElementSibling</code>	The sibling element immediately following <code>node</code> in the node tree
<code>n.parentElement</code>	The parent element of <code>node</code>

`p.nextSibling` is text node

`p.nextElementSibling` returns element ``

`p.parentElement.nextElementSibling`

"aunt" `<div>`

©Professor Tom DeDonno

CIMW 160 – JavaScript – Chapter 6



16

Selecting Nodes

- Use Sizzle w jQuery or querySelector
- `document.querySelector(sizzle);`
 - Returns first matching element node
- `document.querySelectorAll("p.review");`
 - Return nodelist of p tags of class review
- `document.getElementsByTagName("p").getElementsByClassName("review");`
 - Returns an HTML Collection
- HTML Collections contain only HTML element (tag+attr+text)
- NodeList can be element, tag, attribute or text

©Professor Tom DeDonno

CIMW 160 – JavaScript – Chapter 6

17

Creating Element

- Goal create `<h1>My Slideshow</h1>`:
- ```
/* Create HTML Element h1 */
let mainHeading = document.createElement("h1");
/* Then create Text Node or mainHeader.text(...)*/
let headingTxt =
 document.createTextNode("My Slideshow");
/* Then Append to h1 element */
mainHeading.appendChild(headingTxt);
/* to add to any web page */
document.getElementsByTagName(
"body")[0].appendChild(mainHeading);
```

©Professor Tom DeDonno

CIMW 160 – JavaScript – Chapter 6

18

## Add Attributes to Element Node

- Add id="main" and class "lightbox"

```
mainHeading.id = "main";
mainHeading.className = "lightbox";
```

- We now have at bottom

```
<h1 id="main" className="lightbox">
My Slideshow</h1>
```

- /\* To move it to the top \*/

```
b= document.getElementsByTagName("body")[0];
b.insertBefore(mainHeading, b.firstChild);
```

©Professor Tom DeDonno

CIMW 160 – JavaScript – Chapter 6

19

## Setting CSS Styles on Node

- HTMLElement.styles = "CSS Styles";

- element.styles.csspropertyname = "property value";

- <h1 id="main"> My SlideShow</h1>

```
mainHeading.style="font-size:1.5em;color:blue";
```

- Or create new element pointer

```
Let mainH1 = document.getElementById("main");
```

```
mainH1.style.fontSize = "1.5em";
```

```
mainH1.style.color = "blue";
```

- Actual CSS property name is font-size,

- but dash is minus operator, so use camel casing

©Professor Tom DeDonno

CIMW 160 – JavaScript – Chapter 6

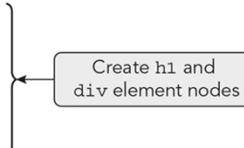
20

## js05 Figure 5-9

```
window.addEventListener("load", createLightbox);

function createLightbox() {
 // Lightbox Container
 let lightBox = document.getElementById("lightbox");

 // Parts of the lightbox
 let lbTitle = document.createElement("h1");
 let lbCounter = document.createElement("div");
 let lbPrev = document.createElement("div");
 let lbNext = document.createElement("div");
 let lbPlay = document.createElement("div");
 let lbImages = document.createElement("div");
```



- Repeating createElement("div") 5 Times
- For the 5 divs we will be using an associative array

©Professor Tom DeDonno

CIMW 160 – JavaScript – Chapter 6

21

## Figure 5-10

```
let lbImages = document.createElement("div");

// Design the lightbox title
lightBox.appendChild(lbTitle); ← Append the lightbox heading
lbTitle.id = "lbTitle";

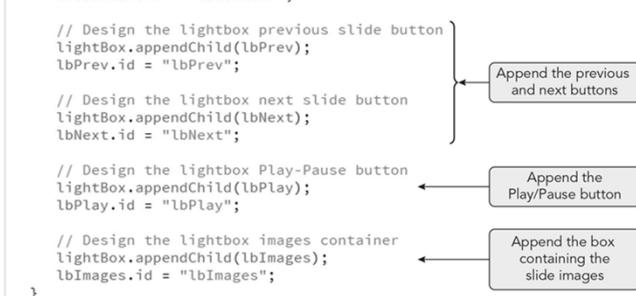
// Design the lightbox slide counter ← Append the image slide counter
lightBox.appendChild(lbCounter);
lbCounter.id = "lbCounter";

// Design the lightbox previous slide button ← Append the previous and next buttons
lightBox.appendChild(lbPrev);
lbPrev.id = "lbPrev";

// Design the lightbox next slide button ← Append the Play/Pause button
lightBox.appendChild(lbNext);
lbNext.id = "lbNext";

// Design the lightbox Play-Pause button ← Append the box containing the slide images
lightBox.appendChild(lbPlay);
lbPlay.id = "lbPlay";

// Design the lightbox images container
lightBox.appendChild(lbImages);
lbImages.id = "lbImages";
```



- Repetitive Task Create an associative array for divs
- var e = { "lbCount":null, "lbPre":null, ... };

©Professor Tom DeDonno

CIMW 160 – JavaScript – Chapter 6

22

## jsFiddle for js05

- Author creates elements
- appends
- Sets id
- We will use an associative Array to do this
- We may need HTML Elements Later on
- Use inspect Verify

©Professor Tom DeDonno

```

1 <div id="lightbox">
2 We need to creat inside elements
3 </div>

JavaScript
1 var e = { /* key: value */
2 "lbCount" : null, "lbPrev" : null,
3 "lbNext" : null, "lbPlay" : null,
4 "lbImages" : null };
5
6 function createLightBox() {
7
8 alert("In createLightBox");
9
10 let lightBox = document.getElementById("lightbox");
11 let lbTitle = document.createElement("h1");
12 lbTitle.id = "lbTitle";
13 lightBox.appendChild(lbTitle);
14
15 for(var key in e) {
16 e[key] = document.createElement("div");
17 console.log(key + ": " + e[key]);
18 lightBox.appendChild(e[key]);
19 e[key].id = key;
20 e[key].innerHTML = "Created:" + key;
21 }
22 }
23 window.addEventListener("load", createLightBox);

```

## Verify 5 divs Browser Inspect

Chrome inspect → Elements  
Firefox inspect → Inspector

Hover the cursor over an element to highlight it in the page

View the element hierarchy in the Elements tab

Nodes created and appended to the document

Selected element is highlighted within the web page

Styles applied to the selected element

## H11 Steps 12-15

- Page174 To add images rate imgFiles array, Open js05.js  
createLightBox add:

```
for(let i=0; i < imgFiles.length ; i++) {
 let image = document.createElement("img");
 image.src = imgFiles[i];
 image.alt = imgCaption[i];
 e["lblImages"].appendChild(image);
}
```

- Use IDE check for syntax errors or undefined variables
- Test system you should see 4 images;

## H11 js05 Add Title/Counter

- Page 17 set lbTitle, and lbCounter  
at end of createLightbox

```
lbTitle.textContent = lightboxTitle;
let currentImg = 1;
e["lbCount"].textContent =
 currentImg + "/" + imgFiles.length;
```

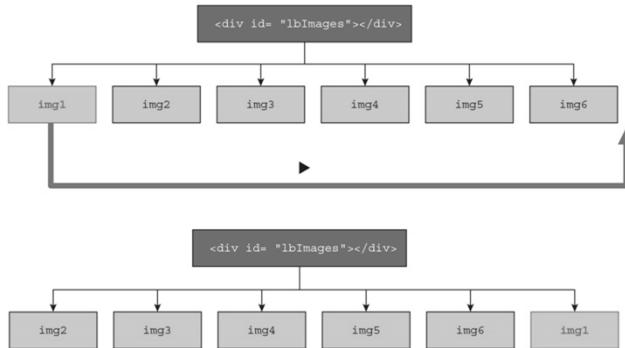
- Other functions will be using currentImg so move  
let currentImg = 1; //to before createLightbox
- use IDE and console to fix any syntax errors

## js05 Add Play Controls

- Set the entities for `lbPrev` `lbNext` and `lbPlay` using jsFiddle.
- Page 176 Save changes and verify operation with new HTML entities.

## Restructuring a Node Tree

- We have Many Images but only want 4 Visible,
- We need to move `firstChild`



## appendChild to Move

```
/* move 1st element to end of image */
e["lblImages"].appendChild(
e["lblImages"].firstElementChild);
++currentImg; //update current image count
```

©Professor Tom DeDonno

CIMW 160 – JavaScript – Chapter 6

29

## Js05 Add showNext

- Page 178 implement the next button, showNext() function – [jsFiddle](#)

```
function showNext() {
 e['lblImages'].appendChild(e['lblImages'].firstElementChild);
 (currentImg < imgFiles.length ? currentImg++ : currentImg = 1);
 e['lbCounter'].textContent = currentImg + " / " + imgFiles.length;
 console.log("in showNext " + currentImg);
}In createLightBox function, add handler for lbNext button.

e["lbNext"].onclick = showNext;
```

- Save all work and test (keep console inspect visible)

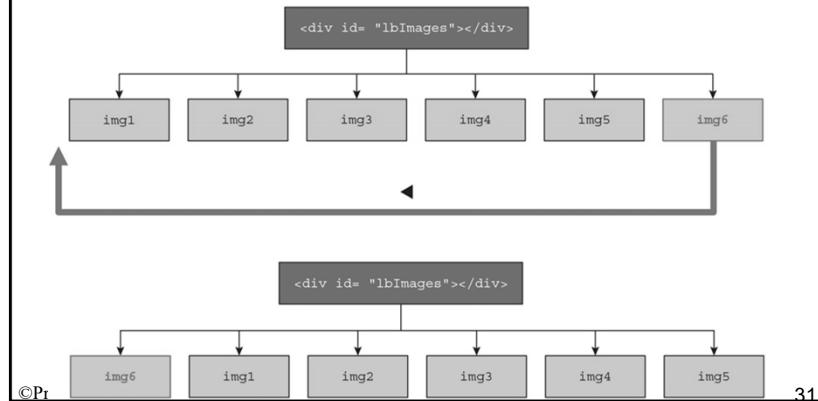
©Professor Tom DeDonno

CIMW 160 – JavaScript – Chapter 6

30

## For Previous go Backwards

We need to go backwards  
last child becomes first child



## Js05 Add showPrevious

```
node.insertBefore(newChild, child)
 - Inserts newChild before child

Can copy code for showNext, use insertBefore(
lastElementChild, firstElementChild) and
decrement currentImg

function showPrevious()
e['lbImages'].insertBefore(
e['lbImages'].lastElementChild,
e['lbImages'].firstElementChild);
//decrement the count watch for 0
(currentImg>1?currentImg--:
 currentImg = imgFiles.length);
e['lbCounter'].textContent =
currentImg + " / " + imgFiles.length;
```

Add listener for showPrevious

## Knowledge Check

```
let aHead = document.createElement("h1");
let aHeadText=document.createTextNode("About Us");
aHead.appendChild(aHeadText);
let aboutDiv = document.getElementById("about");
aboutDiv.appendChild(aHead);

What do first three lines do?
We are creating h1 tag with text about us
<h1>About Us</h1>

What last 2 lines do?
Selecting an element with id="about"
appending as child to aHead
<h1>About Us</h1><x id="about">...</x>
From code we can assume x is a div
```

©Professor Tom DeDonno

CIMW 160 – JavaScript – Chapter 6

33

## Clone a Node

- let image = document.createElement("img");
  - Creating an object in Memory <img>
- image points to that object's address
- With showNext, moving pointers
- X = node.clone( false ); //we copy node
- X =node.clone(true);  
//copy node and all descendants
- X and node point to different addresses

©Professor Tom DeDonno

CIMW 160 – JavaScript – Chapter 6

34

## IbPlay need a interval Timer

- Timed command:  
run command at specified time or repeats at set intervals
- `timeVar = window.setInterval(cmd, interval)`
  - Run cmd (command or function) at every interval
- Stopping timed command
  - `window.clearInterval(timeVar);`
- To run a command just once
  - `let timeVar = window.setTimeout(cmd, delay);`
- `window.clearTimeout(timeVar)`
- Window object is always present
  - `clearTimeout( timeVar );`

©Professor Tom DeDonno

CIMW 160 – JavaScript – Chapter 6

35

## js05.js set Play Button

When play clicked start a 1.5sec interval

When play clicked second time stop interval

```
//add code after onclick=showPrevious
let timelD = false; //undefined default
e["ibPlay"].onclick = function() {
 if(timelD) { //stop slideshow
 clearInterval(timelD);
 timelD = false;
 //undefined is falsely
 } else {
 showNext();
 //every 1.5seconds call showNext
 timelD = setInterval(showNext, 1500)
 }
}; //end onclick
```

©Professor Tom DeDonno

CIMW 160 – JavaScript – Chapter 6

36

## Modal Window Dialog Boxes

- Modal Dialog Box controls the underlying window
- `window.alert( "message" )`
  - Popup window w message
- Confirmation Window
  - `response = confirm( "message" );`
  - Window is always present so its optional
  - Click Ok response=true; cancel false
- Prompt Window for User Input string
  - `response = prompt( Message, default)`
- `window.open( url... )` with 5 second timer.

©Professor Tom DeDonno

CIMW 160 – JavaScript – Chapter 6

37

## Sample Dialog Boxes Fig 5-24



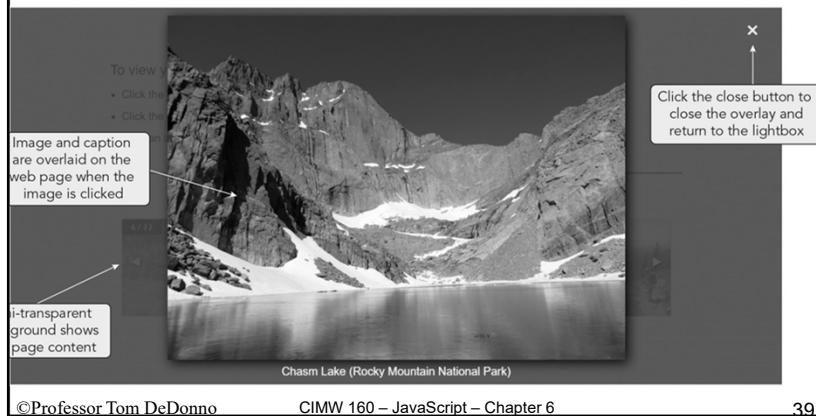
©Professor Tom DeDonno

CIMW 160 – JavaScript – Chapter 6

38

## Add Overlay Popup

- Add a overlay popup window



39

## Create Overlay Page 189

```
function createOverlay() {
 let overlay = document.createElement("div");
 overlay.id = "lbOverlay";

 // Append the figure box to the overlay
 let figureBox = document.createElement("figure");
 overlay.appendChild(figureBox);

 // Append the overlay to the page body
 document.body.appendChild(overlay);
}
```

## Js05 call createOverlay when image is clicked

- In your createLightBox function
- Find in loop adding images
- Add image.onclick = createOverlay;

```
// Add images from the imgFiles array to the container
for (let i = 0; i < imgCount; i++) {
 let image = document.createElement("img");
 image.src = imgFiles[i];
 image.alt = imgCaptions[i];
 image.onclick = createOverlay;
 lbImages.appendChild(image);
}
```

©Professor Tom DeDonno

CIMW 160 – JavaScript – Chapter 6

41

## Copy Image Clicked Information

- Modifying createOverlay w this object

```
function createOverlay() {
 let overlay = document.createElement("div");
 overlay.id = "lbOverlay";

 // Add the figure box to the overlay
 let figureBox = document.createElement("figure");
 overlay.appendChild(figureBox);

 // Add the image to the figure box
 let overlayImage = this.cloneNode("true");
 figureBox.appendChild(overlayImage);

 // Add the caption to the figure box
 let overlayCaption = document.createElement("figcaption");
 overlayCaption.textContent = this.alt;
 figureBox.appendChild(overlayCaption);

 document.body.appendChild(overlay);
}
```

Copy the image that called the function

Append the copied image to the figure box

Display the value of the alt attribute as the figure caption

Append the caption to the figure box

## Js05 Page 191 Remove Modal

- In createOverlay() add close button after alt  

```
let closeBox = document.createElement("div");
closeBox.id = "lbOverlayClose";
closeBox.innerHTML = "×";
```
- Next remove overlay when X is clicked  

```
closeBox.onclick = function()
{ document.body.removeChild(overlay);};
```
- Append close button to overlay  

```
overlay.appendChild(closeBox);
```

©Professor Tom DeDonno

CIMW 160 – JavaScript – Chapter 6

43

**BOM**

©Professor Tom DeDonno

CIMW 160 – JavaScript – Chapter 644

# Browser Object Model

AKA Client-side Object Model

Policy Guideline for Describing a Browser Model

- JavaScript Specifications Mechanism Meets the Model

Contains Objects For All Aspects of Browser

window object aka Global Object:

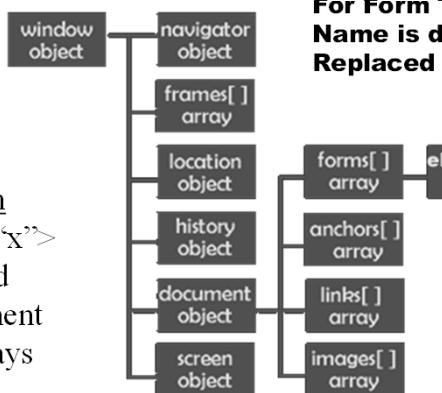
- Top level or Root Object
- Represents Browsers Window Frame
- Created Automatically By Web Browser
- Contains document (Browser Body), history, frames[], etc.

# Browser Object Model

Inheritance  
Example

Anchors.htm

Note <a id="X">  
Is deprecated  
Lists Document  
Objects/Arrays



**For Form Tag**  
**Name is deprecated**  
**Replaced with form ID**

Default  
First  
Image  
is 0..n

## Browser Object Model Hierarchy

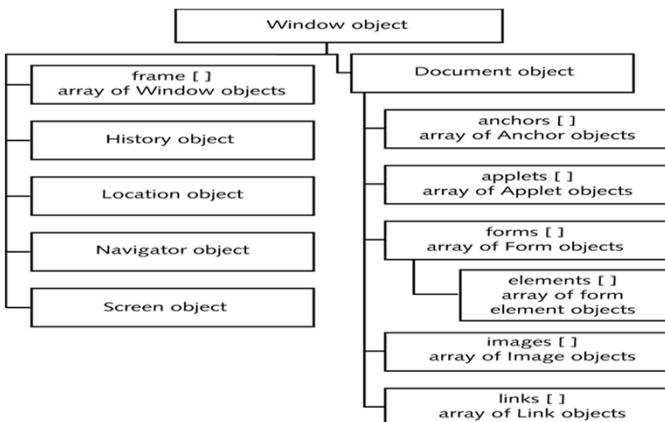


Figure 4-1 Browser object model

©Professor Tom DeDonno

CIMW 260 – JavaScript – Chapter 4

47

## document Object

**Document Body**  
– DOM (Document Object Model)  
– Object:document Class Document

**Methods:** `write()`, `document.writeln()`

### Properties

- `document.forms[ "formID" ]`
- `document.links[]` Array of Links on Page
- `document.images[]` Array of Images on Page

©Professor Tom DeDonno

CIMW 260 – JavaScript – Chapter 4

48

- stop

## Referencing Objects

- Start at window and Work down
  - window is Optional
- document.formName.elementName.value
  - Using Deprecated Name Property
- document.forms[id].elements['elementName'].value
  - Using id Property (Integer, Variable, Literal String)
  - Use id="formID" is preferable unless you have one form forms[0]
  - Id by default starts at 0 Automatic Ordering

## **Forms.htm Simple Adding Machine**

- Form Element 'data' Text Field
- Buttons enter, clear and sum
- Read Only <textarea name='out' ..>
- Enter button Append to textarea
- Clear clear textarea
- Sum Eval textarea (add + in append)

©Professor Tom DeDonno

## **Window Object**

- Class Definition Window
  - Properties Effect Browser Window
  - Can Create New Browser Windows
- Self Property Refers Current Window
  - self.alert() == window.alert() == alert()
  - windowMethods.htm
    - Alert, prompt returns string, confirm return true or false
- Properties, Methods, and Events

©Professor Tom DeDonno

CIMW 260 – JavaScript – Chapter 4

52

## Window Properties

Table 4-1 Window object properties (continued)

Property	Description
parent	Returns the parent frame that contains the current frame
self	Returns a self-reference to the Window object; identical to the window property
status	Specifies temporary text that is written to the status bar
top	Returns the topmost Window object that contains the current frame
window	Returns a self-reference to the Window object; identical to the self property

©Professor Tom DeDonno

CIMW 260 – JavaScript – Chapter 4

53

## Win dow Properties.htm

- Create a 2<sup>nd</sup> Window & Display Properties
- Not Working in IE8 devPHP use IE(new Project)

Table 4-1 Window object properties

Property	Description
closed	Returns a Boolean value that indicates whether a window has been closed
defaultStatus	Sets the default text that is written to the status bar
document	Returns a reference to the Document object
frames[ ]	Returns an array listing the Frame objects in a window
history	Returns a reference to the History object
location	Returns a reference to the Location object
name	Returns the name of the window
navigator	Returns a reference to the Navigator object
opener	Refers to the window that opened the current window

©Professor Tom

## Window: Methods

Table 4-2 Window object methods

Method	Description
<code>alert()</code>	Displays a simple message dialog box with an OK button
<code>blur()</code>	Removes focus from a window
<code>clearInterval()</code>	Cancels an interval that was set with <code>setInterval()</code>
<code>clearTimeout()</code>	Cancels a timeout that was set with <code>setTimeout()</code>
<code>close()</code>	Closes a Web browser window
<code>confirm()</code>	Displays a confirmation dialog box with OK and Cancel buttons
<code>focus()</code>	Makes a Window object the active window
<code>moveBy()</code>	Moves the window relative to the current position
<code>moveTo()</code>	Moves the window to an absolute position
<code>open()</code>	Opens a new Web browser window
<code>print()</code>	Prints the document displayed in the window or frame
<code>prompt()</code>	Displays a dialog box prompting a user to enter information
<code>resizeBy()</code>	Resizes a window by a specified amount
<code>resizeTo()</code>	Resizes a window to a specified size
<code>scrollBy()</code>	Scrolls the window by a specified amount
<code>scrollTo()</code>	Scrolls the window to a specified position
<code>setInterval()</code>	Repeatedly executes a function after a specified number of milliseconds have elapsed
<code>setTimeout()</code>	Executes a function once after a specified number of milliseconds have elapsed

©Professor Tom DeDonno

CIMW 260 – JavaScript – Chapter 4

55

## Window Events

- User Generated Events
- Events = EventHandler
  - Javascript code For handling Event
  - Form Elements Using Events
    - Anchor `<a href="..." onMouseOver="...">`
      - Anchor has a default onClick Handler (Open Page)
    - `<input type="button" onClick="EventHandler">`
    - `<input type="text" onDoubleClick="EventHandler">`

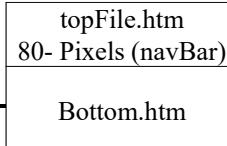
©Professor Tom DeDonno

CIMW 260 – JavaScript – Chapter 4

56

## Frameset.htm

```
<HTML> <HEAD>
<TITLE>A Page With Frames</TITLE> </HEAD>
<FRAMESET ROWS="80,*">
 <FRAME SRC="topFile.htm" NAME="topFrame">
 <FRAME SRC="bottom.htm" NAME="mainFrame">
<NOFRAMES> Now A Days All Browser Support Frames
</NOFRAMES>
</FRAMESET>
</HTML>
```



Lots of People Argue You Should Not Use Frames  
Linking Someone else Work, Which Page, etc.

©Professor Tom DeDonno

CIMW 260 – JavaScript – Chapter 4

57

## Examples

- file.htm opens The File Up in the...
  - \_blank open in New Tab
  - window.open( URL ) open in New Window
- Frameset.htm need to add form example
  - Default, \_self is the current Frame
  - \_blank Opens in a New Browser Tab
  - \_top (Window that Created Frames)
  - \_parent Parent Window

HTML iframe, HTML5 canvas

©Professor Tom DeDonno

CIMW 260 – JavaScript – Chapter 4

58

## Mouse Events

- mouseover: Mouse passes Over Element
- mouseout: Mouse moves off an element
- mousedown: on element click mouse btn
- mouseup event: you release mouse button
- [mouseEvents.htm](#)

## Browser Object Model Quiz

- ❶ Four Targets For Opening A New Web Page?
- ❷ The \_\_\_\_\_ Object is the 1<sup>st</sup> Browser Model Object?
- ❸ With HTML <form> tag use \_\_\_\_\_ or forms[ index ].
- ❹ Input Elements use the Field \_\_\_\_\_ to represent data?
- ❺ The \_\_\_\_\_ object represents the entire Web Browser Window. The \_\_\_\_\_ object corresponds to the DOM model & is a child of the top level \_\_\_\_\_ object.

# Window

open & close methods

## Window.open & Close

- Used to Open New Browser Window
  - Creates a New Browser Window Object
- [automatically opens
  - New self window](url)
- Can Use Browsers
  - File → new Window
  - File → new Tab (in Real Browsers)

## Window.open method

- Strict DTD You must use open() method
- window.open( *url*, WindowName [,features] [,replace] );
- URL is the address  
replace if True replace current page in browsers history
- WindowName is Window Object Identifier
- Window.open(  
“cim.saddleback.edu”, “”, “height=200,width=300” );
  - Open Up a Window with no status, navigation,etc of size 300x200
- Features “toolbar=1,status=0”
  - Window Has Toolbar and But No Status Bar
- focus() makes the window Active Object

©Professor Tom DeDonno

CIMW 260 – JavaScript – Chapter 4

63

## Window.open Arguments

Table 4-3 Arguments of the Window object's open( ) method

Argument	Description
URL	Represents the Web address or filename to be opened
name	Assigns a value to the name property of the new Window object
options	Represents a string that allows you to customize the new Web browser window's appearance
replace	A Boolean value that determines whether the URL should create a new entry in the Web browser's history list or replace the entry

©Professor Tom DeDonno

CIMW 260 – JavaScript – Chapter 4

64

## Window.open features

Table 4-4 Common options of the Window object's open() method

Name	Description
directories	Includes directory buttons
height	Sets the window's height
location	Includes the URL Location text box
menubar	Includes the menu bar

Menubar = {1 | 0 } display Menu Bar Default is Yes  
location enter URL in the browser edit bar

## Window.open features

Table 4-4 Common options of the Window object's open() method (continued)

Name	Description
resizable	Determines if the new window can be resized
scrollbars	Includes scroll bars
status	Includes the status bar
toolbar	Includes the Standard toolbar
width	Sets the window's width

### Detailed List of Features-Option [javascripter.net/faq/openinga.htm](http://javascripter.net/faq/openinga.htm)

## windowOpen.htm

- Open( URL, Identifier, Option String, Boolean fReplace)
- myHome = open( “<http://saddleback.edu>”)
- myHome.close( );
- Options are Comma separated
  - Attribute Name Value
  - Height=200; //200 Pixel Height
  - Toolbar= 0; //No Toolbar

©Professor Tom DeDonno

CIMW 260 – JavaScript – Chapter 4

67

## Window - Quiz

- What does Window Object Represent?
- //What does this Code Do?

```
myX = window.open(“”, “”, “width=100, height=100”)
myX.document.write(“Hello”);
myX.document.bgColor=“blue”;
```
- To create a Windows with no Toolbar or Bottom Status Bar use option “toolbar=? ,status=?” ?

©Professor Tom DeDonno

CIMW 260 – JavaScript – Chapter 4

68

# JavaScript Timers

Single Timeout: setTimeOut() &  
clearTimeOut()

Interval: setInterval() clearInterval()

```
<meta http-equiv="refresh"
content="0; url=" />
```

## setTimeout/clearTimeout

- Window Methods
- Var id = setTimeout( “function”, millisec)
- clearTimeout( id )
- Single Event at X millisecs After setTimeOut
- Give PHP Students  
5 Seconds to View Homework
- countDown.htm

## countDown.htm

```
<script type="text/javascript">
var timeLength = 5;
var myAssignment; //assignment window
function winClose() { myAssignment.close(); }

function myOpen(str)
{
 alert("Do not forgot " + timeLength + " Seconds Only ");
 myAssignment = open(str, "Assignment", "toolbar = 0, width=400, height=500"
);

//why is function in quotes?
self.setTimeout("winClose();", timeLength*1000);
}

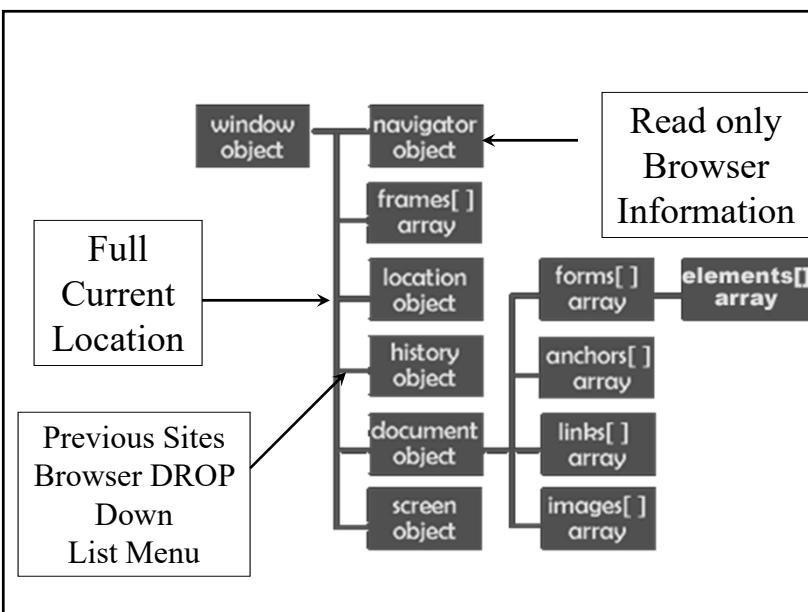
</script>
</head> <body> ...
<input name="Input" type="button" value="Open Assignments"
onClick="myOpen(
 \"/http://saddleback.edu/faculty/tdedonno/cim225/assignment.htm\"
);"
©Professor Tom DeDonno
```

## **setInterval/clearInterval**

- Window Methods
- Execute Code at Specified Interval
- Id = setInterval( ‘function’, millisec)
- clearInterval( id )
- Write a Client-Side Clock
  - Including Seconds & H:M:S
  - Reports Time & Seconds Left In Class
  - myInterval.htm

# Other Window Objects

history  
location  
navigator



# History Object

## List of Pages Visited In Current Session

Each Window Has Own History

- Tabs Back Button Correspond to Active Window
- Top Address Drop Down List All Sites

## Security

- Cannot Display URL History
- But you Can Navigate
  - Methods: back(), forward(), go(delta#)
- Browser has Details
  - So Engine Could Allow Security Leak

# History Navigation

Table 4-5 Methods of the History object

Method	Description
back()	Produces the same result as clicking a Web browser's Back button
forward()	Produces the same result as clicking a Web browser's Forward button
go()	Opens a specific document in the history list

**history.go( -2 ) Go Back 2 In History**

**history.go( location );**

**Example history.htm**

## Location Object

- Object Describing Current Location – Full URL
- Protocol://hostname:port pathname#anchor

Property	Description
hash	Anchor name
host	Hostname and Port String
hostname	Server Name – domain name
href	Entire URL
pathname	URLs Path
port	Port Number 80 for http
protocol	http for the Web
search	URL query portion after ?

©Professor Tom DeDonno

CIMW 260 – JavaScript – Chapter 4

77

## Location Methods

Table 4-7 Methods of the Location object

Method	Description
assign()	Loads a new Web page
reload()	Causes the page that currently appears in the Web browser to open again
replace()	Replaces the currently loaded URL with a different one

**Setting location.href Property Will Move You  
Reload() reloads the location.href Property**

**Replace(URL) replaces the current history with URL  
Element with the Specified URL**

©Professor Tom DeDonno

CIMW 260 – JavaScript – Chapter 4

78

## Location.htm

- Create JavaScript Array With Each Location Property
- On Click Step Through Array Writing Output to a TextArea
- Add Submit with Current File  
`<form action="" ...>`
- Add Textbox to Enter URL  
Add Button `location.href=URL`

## **Navigator Object**

- Property Information About Browser
- Adopted Netscape Navigator
- Browser Have Different Plugins/Properties
- userAgent Property Identifies Browser
- 
- Useful to Determine Browser Capabilities
  - E.g., `is javaEnabled();`
- Can Set Some Preferences “Netscape Only”
  - `preference( prefName [,setValue] );`

# Navigator Properties

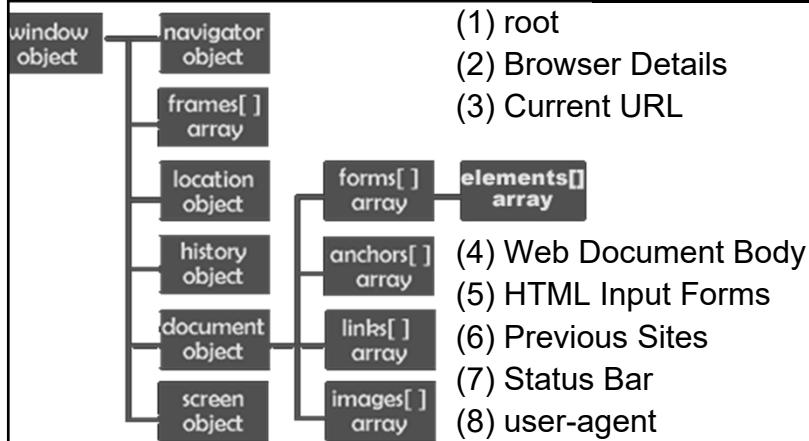
Table 4-8 Properties of the Navigator object

Properties	Description
appCodeName	The Web browser code name
appName	The Web browser name
appVersion	The Web browser version
platform	The operating system in use on the client computer
userAgent	The string stored in the HTTP user-agent request header, which contains information about the browser, the platform name, and compatibility

**with( object )**  
**{can Use [Property,method] skip object }**  
**navigator.htm (Similar to Code on Page 198)**

# Summary Quiz

## Identify Objects..



# Referring to Frames & Windows

target, parent, top

## FrameSet2.html Review

- Create a Three Frame Web Page
- leftFrame is navBar.htm
  - User can...
    - Set Specific Targets Locations
    - Select Links & Button to Open Pages
    - Print out Frame Information,  
Need to use parent Object

## Frameset2.html

120px Width leftFrame navBar.htm	topFile.htm 80- Px Height
	mainFrame bottomFile.htm

Three Frame System

DW Insert Panel Layout Last Icon Frames

Left with Nested Top/Bottom Frames

```
frameset cols="120,*" frameborder="Yes" >
frame src="navBar.htm" name="leftFrame" >
frameset rows="80,*" >
<frame src="topFile.htm" name="topFrame" >
<frame src="bottomFile.htm" name="mainFrame" >
/frameset>
/frameset>
```

©Professor Tom DeDonno

## Target Property

- Determines Where Document Should Open
- `<a href="URL" target="frameName">`
  - Will Find FrameName in Other Tabs
  - Will not Open if FrameName not Present
- Default Target is Self
  - Other Targets=[\_blank,\_top,\_parent,\_self]
- How do set Default Target for All Links on a Page

©Professor Tom DeDonno      CIMW 260 – JavaScript – Chapter 4

## HTML base tag

- <base target="topFrame">
  - Sets default target topFrame
- <base href="http://saddleback.edu/">
  - All Relative URL prefix base href
  - <a href="faculty/tdedonno">
  - Try to Use Relative Links
  - But System A all Data on System B

## parent Property

- Object parent : holder of Current frames
- window.frames.length is 0 (FrameSet)
  - Window object Selected Frame
- To refer to a frame in a FrameSet
  - Like Forms 1<sup>st</sup> Frame is 0 parent.frames[0]
- parent.frames[ 0 ]
  - Window Object for First Frame
  - parent.frames[0].document : Document Object

## navBar.htm Select Targets

- leftFrame of Frameset2.html
- Drop Down List Box To Select Targets

```
function newTarget(t)
{
 alert("inside onChange eventHandler: " + t);
 baseTarget = t;

 //Set all links to new baseTarget
 var link, l = 0;
 while (link = document.links[l++]) link.target = baseTarget;

 (document.getElementById("baseTarget")).value = baseTarget;
}
```

©Professor Tom DeDonno

## Drop Down List Box

- <select> Tag
  - Each Element <option>...</option>
- onChange Event
  - List Box is Losing Focus & New Element Selected
  - Requires an Empty Default Selection
- <http://www.comptechdoc.org/independent/web/cgi/javamanual/javaselect.html>
- this.options[ this.selectedIndex ].value
  - Inside Select This is Pointer to Options
  - <select onChange=  
"newTarget(  
this.options[ this.selectedIndex ].value);" >

©Professor Tom DeDonno

## Document Methods

- **getElementsByTagName( tag )**
  - Returns Array of All Tag Elements
  - `(document.getElementsByTagName( "base" ))[0].name`
    - Return name of first <base> tag in the Document
- **getElementByID( idName )**
  - Return pointer to Element with id
  - `(document.getElementById( "baseTarget" )).value = baseTarget;`
    - Set element <tag id="baseTarget" to Value of Variable baseTarget

©Professor Tom DeDonno

## FrameSet2.html Objective Review

- Create a Three Frame Web Page
- leftFrame is navBar.htm
  - User can...
    - Set Specific Targets Locations
    - Select Links & Button to Open Pages
    - Print out Frame Information

©Professor Tom DeDonno

## Summary BOM Objects

- Window: Browser Window – Top Level
  - Methods: open, close, setInterval; Properties: status
  - Pop-up Windows: alert, confirm, prompt
- Document: document Body – Follows DOM
  - Write, writeln
  - Forms: HTML Form Input
    - Elements: <input or <textarea
- Location: Current Web Page Locations
- History: URL history List
- Navigator: Web Browser Properties

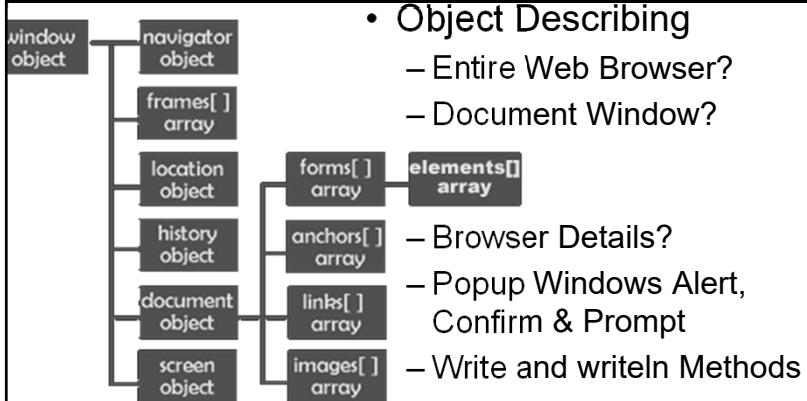
©Professor Tom DeDonno

CIMW 260 – JavaScript – Chapter 4

93

## Summary Quiz

- Model Used by JavaScript Describing Entire Browser?



- Object Describing

- Entire Web Browser?
- Document Window?
- Browser Details?
- Popup Windows Alert, Confirm & Prompt
- Write and writeln Methods
- User-agent
- Back button

©Professor Tom DeDonno

CIMW 260 – JavaScript – Chapter 4

94

## Window Has Frames (Panes)

Target = \_self or Default Current Frame

on <a href=>  
default handler Open \_self Window

\_target: blank (New Browser Window)

\_parent: Parent of Frames or Window

\_top: Top Window or Frameset

File.htm illustrates Targets, no framesets