

INTERNET & WEB

APPLICATION DEVELOPMENT

SWE 444

Fall Semester 2008-2009 (081)

Module 3 (IV): Client-Side Scripting (JavaScript)

Dr. El-Sayed El-Alfy
Computer Science Department
King Fahd University of Petroleum and Minerals
alfy@kfupm.edu.sa

Objectives/Outline

- **Objectives**
 - Understand the Document Object Model
 - Use JavaScript to manipulate DOM Objects

- **Outline**
 - DOM
 - Manipulation of DOM with JavaScript

KFUPM-081 © Dr. El-Alfy SWE 444 Internet & Web Application Development 3.2

The Document Object Model (DOM)

- XHTML elements are viewed as objects, and many of their attributes are treated as properties of those objects
- The elements in a document make up the page's DOM tree, which describes the relationships among elements
- Each element can be identified with a unique ID which can be used by scripting languages to manipulate various properties of the element.
 - Using JavaScript, the page content can be modified by creating new elements, deleting or modifying existing elements.
 - Example: the style of an element can be accessed using:

```
<elementID>.style.<propertyName>
```
- DOM is language independent, i.e. can be used by any programming language like Java, JavaScript, and VBScript.
- Not all web browsers implement all features included in the DOM specification.

DOM Tree

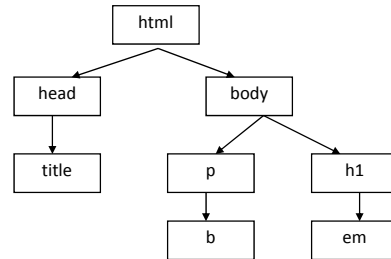
- Nodes in the DOM tree represent elements and are related to each other through child-parent relationships
 - A node may have multiple children, but only one parent
 - Nodes with the same parent node are referred to as siblings
 - The document node in a DOM tree is called the root node, because it has no parent

Example

```

<html>
<head>
<title>/</title>
</head>
<body>
<p>This is paragraph number
  <b>one</b></p>
<h1>heading 1 is the <em>largest</em>
  level</h1>
</body>

```



Visual Inspection of DOM Tree

- To see a visual representation of a document's DOM tree and information about each node, use
 - Firefox's DOM Inspector
 - IE Web Developer Toolbar

Example

```

1 <?xml version = "1.0" encoding = "utf-8"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
3 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
4
5 <!-- Fig. 12.1: domtree.html -->
6 <!-- Demonstration of a document's DOM tree. -->
7 <html xmlns = "http://www.w3.org/1999/xhtml">
8 <head>
9 <title>DOM Tree Demonstration</title>
10 </head>
11 <body>
12 <h1>An XHTML Page</h1>
13 <p>This page contains some basic XHTML elements. We use the Firefox
14 DOM Inspector and the IE Developer Toolbar to view the DOM tree
15 of the document, which contains a DOM node for every element in
16 the document.</p>
17 <p>Here's a list:</p>
18 <ul>
19 <li>One</li>
20 <li>Two</li>
21 <li>Three</li>
22 </ul>
23 </body>
24 </html>

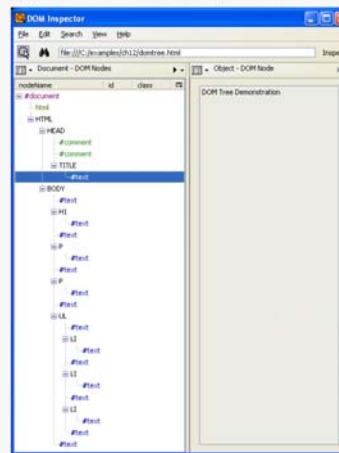
```

KFUPM-081 © Dr. El-Alfy SWE 444 Internet & Web Application Development

3.7

Example: Document's DOM Tree

b) The Firefox DOM inspector displays the document tree in the left panel. The right panel shows information about the currently selected node.



To get FireFox DOM Inspector, run FireFox installer and check the DOM Inspector in the Custom Setup Type

KFUPM-081 © Dr. El-Alfy SWE 444 Internet & Web Application Development

3.8

Example: Document's DOM Tree

c) The Internet Explorer Web Developer Toolbar displays much of the same information as the DOM inspector in Firefox in a panel at the bottom of the browser window.



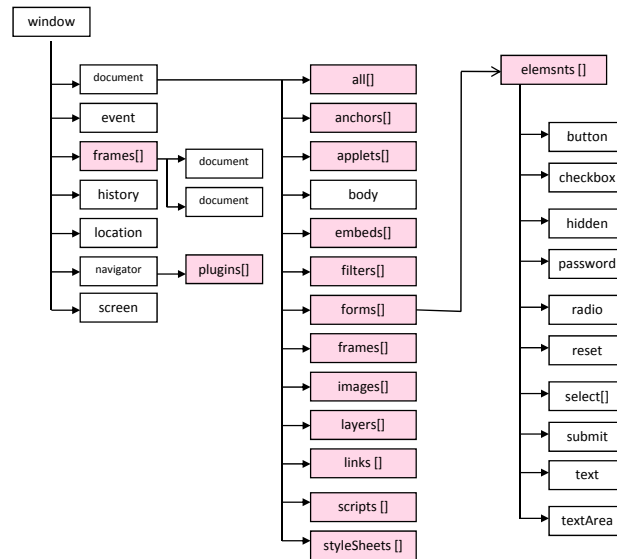
IE Web Developer Toolbar can be downloaded from Microsoft at go.microsoft.com/fwlink/?LinkId=92716

Collections

- Related objects on a page is referred to as a collection.
- There are several special collections, e.g.
 - The all collection is a collection of all elements in a document or a higher level element, e.g.
 - document.all // all elements under the document
 - body.all //all elements under the body
 - To refer to the direct elements within another element use the children collection:
 - html.children
- For complete list of objects and collections, visit msdn.microsoft.com/workshop/c-frame.htm
- Example
 - you can access all elements in a document using the following for loop in JavaScript:


```
for(var i=0; i<document.all.length; i++)
  document.write(document.all[i].tagName+"<br>")
```

W3C DOM



KFUPM-081 © Dr. El-Alfy SWE 444 Internet & Web Application Development

3.11

W3C DOM (cont.)

window	represents the browser window
document	represents the HTML document rendered in a window
body	provides access to the body element
history	keeps track of the sites visited by the browser
navigator	contains information about the web browser
location	contains the URL of the rendered document
event	contains information about the event that occurred
screen	contains information about the computer screen
all[]	provides access to all elements contained in an object
anchors[]	contains all the anchor elements on the page that have a name or id
applets[]	contains all applet elements in a page
embeds[]	contains all the embed elements
forms[]	contains all form elements
frames[]	contains all the windows that represent the frames
images[]	contains all the img elements
links[]	contains all the anchor elements with an href attribute
plugins[]	similar to embed[]
scripts[]	contains all the script elements
styleSheets[]	contains each style element in a document

KFUPM-081 © Dr. El-Alfy SWE 444 Internet & Web Application Development

3.12

The window object

- The window object is a “master” DOM object at the top of the DOM hierarchy
- Represents the browser window
- Provides access to the document object contained in the window,
- Also contains history and location objects
- An instance of a window object is created when a browser is launched
- Its properties become available for inspection/use as:
 - `window.property` or `self.property` or `only property`
- Useful properties:
 - `defaultStatus`: sets the default text to display in the status bar at the bottom of the window
 - `status`: sets/gets the text in the status bar
 - `name`: gets/sets the name of the window
 - `location`: gets URL of the document in the window
 - `screenLeft`, `screenTop`: gets/sets the left and top corner position of the window relative to left-top corner of the screen
 - `length`: number of frames in window
 - `frames`: an array of window objects, one for each frame
 - `parent`: since frames are window objects, sometimes parent window is needed
- Examples:
 - `window.document` : if frameless, accesses the top level document. If frames, accesses the top frame's document
 - `window.frame[1].document` : Access the document contained in the first frame
 - `frame[1].parent.document` : Access the document contained in the parent frame

The window object (cont.)

- Like other objects, the window object has methods, too, that become available when a new window is opened
- Some methods:
 - `alert()`, `confirm()` and `prompt()`
 - are actually methods of the window object, e.g.: `window.alert("Error!")`
 - `focus()`, `close()`, `resizeTo(h,v)`, `moveTo()`, `resizeBy()`, `moveBy()`, `print()`, `scrollTo()`, `scrollBy()`
 - `forward()`, `back()`, `home()`
 - `open()`
 - `window.open("URL", "titlebar=yes|no, toolbar=yes|no, menubar=yes|no, scrollbars=yes|no, status=yes|no, location=yes|no, resizable=yes|no, copyhistory=yes|no, width=pixels, height=pixels, alwaysRaised=yes|no")`
- Window timers:
 - two sets of window methods relate to setting up timing devices to control the automatic display of pages.
 - Delay timer
 - introduces a delay before showing a page;
 - established with `setTimeout()` and cleared with `clearTimeout()` methods.
 - `setTimeout()` causes the script to pause for a specified number of milliseconds.
 - `setTimeout("statement", milliseconds)`
 - Interval timer
 - defines a continuous interval during which activities are repeated.
 - established with `setInterval()` and cleared with `clearInterval()` methods.

Delay Timer Example

```
<script type="text/javascript">
var SlideWindow;
function SlideShow(){
    SlideWindow = open("slid1.jpg", "", "width=300,height=200");
    SlideWindow.moveTo(400,400);
    setTimeout("SlideWindow.location='slide2.jpg'", 2000);
    setTimeout("SlideWindow.location='slide3.jpg'", 4000);
    setTimeout("SlideWindow.location='slide4.jpg'", 6000);
    setTimeout("SlideWindow.location='slide5.jpg'", 8000);
    setTimeout("SlideWindow.location='slide6.jpg'", 10000);
    setTimeout("SlideWindow.close()", 12000);
}
</script>
<input type="button" value="Slide Show" onclick="SlideShow()"/>
```

Interval Timer Example

```
<script type="text/javascript">
var SlideCount;
var SlideWindow;
function SlideShow(){
    SlideCount = 1;
    SlideWindow = open("Slid1.jpg", "", "width=300,height=200");
    SlideWindow.moveTo(400,400);
    SlideTimer = setInterval("ShowNextSlide()",2000);
}
function ShowNextSlide(){
    SlideCount ++;
    if (SlideCount <= 5 && SlideWindow.closed != true) {
        SlideWindow.location = "Slide" + SlideCount + ".jpg";
    }
    else {
        clearInterval(SlideTimer);
        SlideWindow.close();
    }
}
</script>
<input type="button" value="Show Slides" onclick="SlideShow()"/>
```


The window object (cont.)

Event	Description
Onfocus	Occurs when the window gains the keyboard focus
Onblur	Occurs when the window loses the keyboard focus
Onload	Occurs when the loading the page is finished
Onunload	Occurs when the user leaves the page to another URL
Onmove	Occurs when the window location on the screen is changed
Onresize	Occurs when the window size is changed

The document object

- Typically represents the most accessed XHTML document rendered in a window
- The document object provides access to every element in the XHTML document and allows dynamic modification of the XHTML document.
 - Forms, tables, paragraphs, lists, images, etc.
- Contains several collections for accessing all elements of a given type.
- Frameless document: Access as `window.document` or `document`
- Document contained in a frame: `window.frame[x].document`, where `x` is the number or name of the frame
- Consult a reference for properties and methods

The document object (cont.)

➤ Identify elements in the document

`document.getElementById("id")` An XHTML element appearing in a document and identified by its assigned `id` value.

`document.all.id` Alternate reference to an XHTML element appearing in a document and identified by its assigned `id` value.

`id` Alternate reference to an XHTML element appearing in a document and identified by its assigned `id` value.

The document object (cont.)

Property	Description
<code>alinkColor</code>	returns the color of the activated links (that was initially set by the <code>alink</code> attribute of the <code><body></code> element)
<code>linkColor</code>	returns the color of unvisited links (that was initially set by the <code>link</code> attribute of the <code><body></code> element)
<code>vlinkColor</code>	returns the color of visited links (that was initially set by the <code>vlink</code> attribute of the <code><body></code> element)
<code>bgColor</code>	returns as sets the background color (that was initially set by the <code>bgcolor</code> attribute of the <code><body></code> element)
<code>fgColor</code>	returns the foreground color (that was initially set by the <code>text</code> attribute of the <code><body></code> element)
<code>anchors</code>	an array of anchor objects contained in the document
<code>forms</code>	an array of form objects contained in the document
<code>images</code>	an array of image objects contained in the document
<code>links</code>	an array of link objects contained in the document
<code>URL</code>	returns the actual URL of the current document
<code>title</code>	returns the string specified by the <code><title></code> element

The document object (cont.)

Method	Description
close()	closes the output stream to the document and displays any results that have not been displayed
open()	creates a new document in the current window then use write and writeln to add the content
write()	send output to the document
writeln()	send output to the document with a trailing newline character
getSelection()	returns the selected text in the document

The navigator object

- The navigator object at the top of the DOM hierarchy represents the browser.
 - has properties used to get information about the browser, version, OS platform etc
- Can be accessed as window.navigator or just navigator
- Useful properties:
 - appName: name of browser used (can be deceiving)
 - appVersion: version of browser used
 - platform: operating system in use
 - cookieEnabled: can the browser store cookies?

The location object

- Contains the URL of the rendered document.
- When this object is set to a new URL, the browser immediately navigates to the new location
- Can be accessed as `window.location` or just `location`
- Useful properties:
 - `href`: retrieves entire URL
 - `host`: retrieves just the domain name (ex: yahoo.com)
 - `pathname`: retrieves just the path inside the domain (page name is at end)
 - `hash`: retrieves the anchor

The location object (cont.)

- The location is extremely useful in setting up scripted links.
- Has the same effect as the `<a>` tag but with scripted control.
- Can create links using it along with other DHTML settings
- Example

```
<b>A scripted link:</b>
<span style="color:blue; text-decoration:underline;
  cursor:hand"
  onmouseover="this.style.fontWeight='bold' "
  onmouseout="this.style.fontWeight='normal' "
  onmousedown="this.style.color='red' "
  onmouseup="this.style.color='blue' "
  onclick="location='http://webcourses.kfupm.edu.sa' ">
  Link to WebCT (KFUPM's)
</span>
```

The history object

- Keeps track of the sites visited by the browser user.
- Can be accessed as window.history or just history
- The object provides a script programmer with the ability to move forward and backward through the visited sites.
- Useful properties:
 - next, previous (tells you the URL, but won't direct you there)
- Useful methods:
 - back: same as pressing the back arrow button
 - forward: same as pressing the forward arrow button
 - go(): go back or forward a given number of pages; to go back 3 pages:
 - history.go(-3);

Event Model

Event attribute	Description	Allowed with elements
Onblur	occurs when a form element loses focus by clicking or tabbing to another element	<a>, <area>, <button>, <input>, <select>, <textarea>
Onchange	occurs when a form element has been changed and lost user focus	<input>, <select>, <textarea>
Onclick	Indicates that the element has been clicked	Most elements
Ondblclick	Indicates that the element has been double-clicked	Most elements
Onfocus	occurs when the element gains focus	<a>, <area>, <button>, <input>, <select>, <textarea>
Onkeydown	occurs when a key is being pressed down	Most elements
Onkeypress	occurs when a key is being pressed and released	Most elements
Onkeyup	occurs when a key is being released	Most elements
Onload	occurs when a window or frame finishes loading a document	<body> <frameset>
Onmousedown	occurs when the mouse button is pressed	Most elements
Onmousemove	Indicates the mouse has moved	Most elements
Onmouseout	Indicates the mouse has moved away from the element	Most elements
Onmouseover	Indicates the mouse has moved over an element	Most elements
Onmouseup	Indicates the release of mouse button	Most elements
Onreset	occurs when the form is being reset by pressing the reset button	<form>
Onselect	occurs when the user selects text	<input>, <textarea>
Onsubmit	occurs when the form is being submitted by pressing the submit button	<form>
Onunload	occurs when the browser leaves the current document and unloads it from the window or frame	<body> <frameset>

Q & A



References

- Some useful links with examples and other resources:
 - *Internet and World Wide Web How to Program*, 4/e, H. M. Deitel, P. J. Deitel, and A. B. Goldberg, Pearson Education Inc., 2008. Chapters 6-13.
 - *Web Development and Design Foundations with XHTML*, 4/e, Pearson Education Inc. 2009. Chapter 14.
 - <http://devedge-temp.mozilla.org/library/manuals/2000/javascript/1.3/guide/intro.html>
 - <http://msconline.maconstate.edu/tutorials/JSHTML/default.htm>
 - <http://www.javascript.com>
 - Beginning JavaScript Tutorials <http://www.pageresource.com/jscript/index.html>
 - JavaScript Tutorial for the Total Non-Programmer <http://www.webteacher.com/javascript/>
 - More Beginning JavaScript Tutorials <http://echoecho.com/javascript.htm>
 - Core JavaScript 1.5 Reference Manual http://www.webreference.com/javascript/reference/core_ref
 - The JavaScript Source <http://javascript.internet.com>
- <http://devedge-temp.mozilla.org/library/manuals/2000/javascript/1.3/guide/intro.html>
- <http://msconline.maconstate.edu/tutorials/JSHTML/default.htm>
- <http://www.javascript.com>