JavaScript

Chapter 4
Manipulating the Browser Object Model

Lecturer: Thiều Quang Trung

Objectives

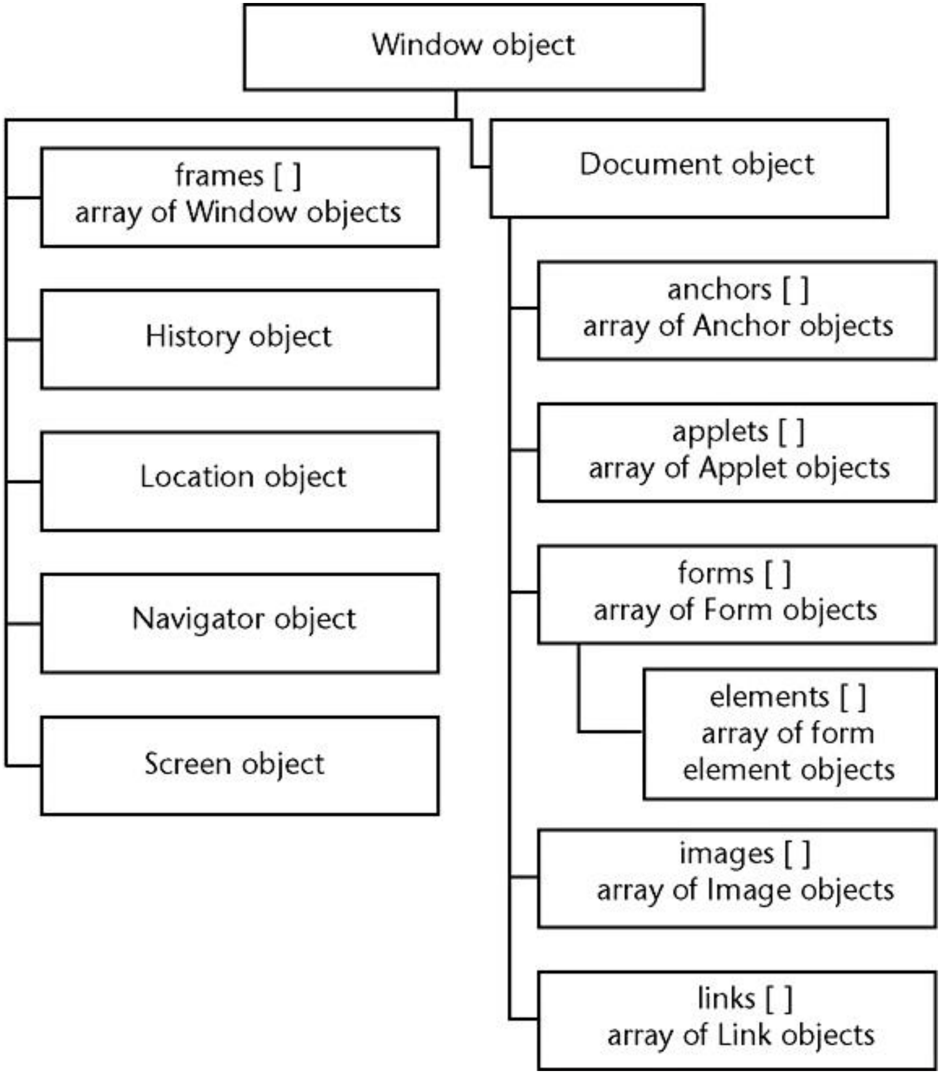
- Study the browser object model
- Work with the `Window` object
- Study the `History`, `Location`, and `Navigator` objects
- Use JavaScript to refer to windows and frames

Understanding the Browser Object Model

- **Browser object model (BOM)** or **client-side object model**
  - Hierarchy of objects
  - Each provides programmatic access
    - To a different aspect of the Web browser window or the Web page
- **`Window` object**
  - Represents a Web browser window or an individual frame within a window
  - Called the **global object** because all other objects in the browser object model are contained within it

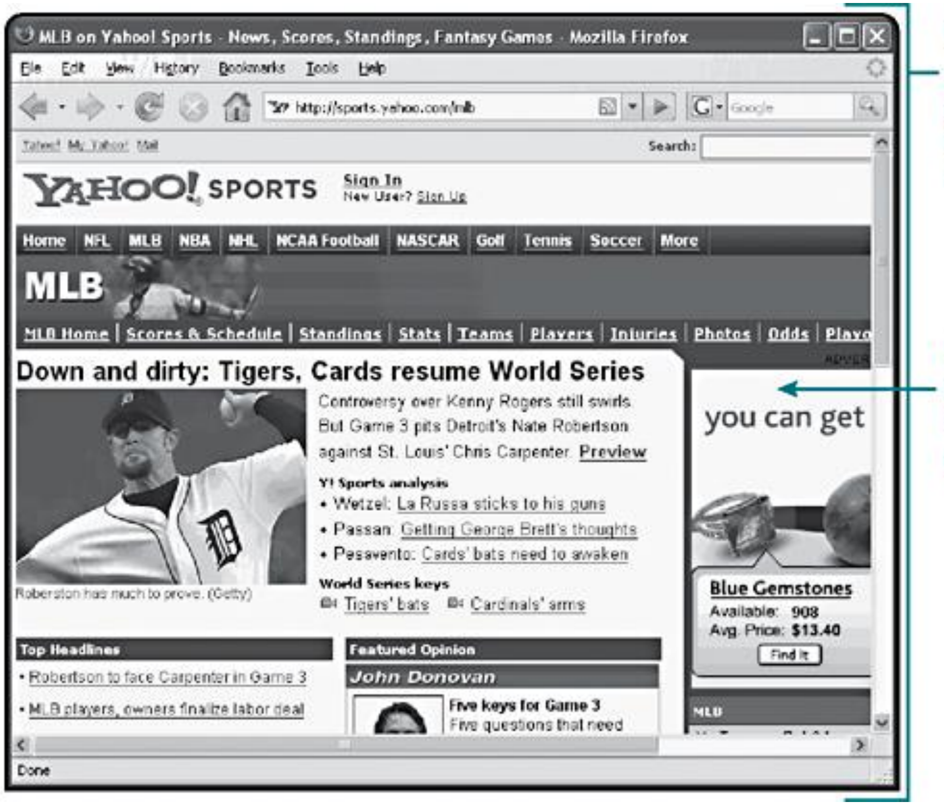
Understanding the Browser Object Model (continued)

Figure 4-1
Browser object model


Window object
frames [ ] array of Window objects
History object
Location object
Navigator object
Screen object
Document object
anchors [ ] array of Anchor objects
applets [ ] array of Applet objects
forms [ ] array of Form objects
elements [ ] array of form element objects
images [ ] array of Image objects
links [ ] array of Link objects

Understanding the Browser Object Model (continued)

Figure 4-2
Window object and Document object


MLB on Yahoo! Sports - News, Scores, Standings, Fantasy Games - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://sports.yahoo.com/mlb
Google
Yahoo! My Yahoo! Mail
Search:
YAHOO! SPORTS
Sign In
New User? Sign Up
Home NFL MLB NBA NHL NCAA Football NASCAR Golf Tennis Soccer More
MLB
MLB Home | Scores & Schedule | Standings | Stats | Teams | Players | Injuries | Photos | Odds | Playo
you can get
Blue Gemstones
Available: 908
Avg. Price: $13.40
Find It
MLB
Done

Down and dirty: Tigers, Cards resume World Series

Controversy over Kenny Rogers still swirls But Game 3 pits Detroit's Nate Robertson against St. Louis' Chris Carpenter. Preview

Roberston has much to prove. (Getty)

Y! Sports analysis
- Wetzel: La Russa sticks to his guns
- Passan: Getting George Brett's thoughts
- Pesavento: Cards' bats need to awaken

World Series keys
Tigers' bats Cardinals' arms

Top Headlines
- Robertson to face Carpenter in Game 3
- MLB players, owners finalize labor deal

Featured Opinion
John Donovan
Five keys for Game 3
Five questions that need

Web browser window represented by the Window object

Web page represented by the Document object

The `Document` Object

- `Document` object
  - Represents the Web page displayed in a browser
  - Has methods such as
    - `write()` and `writeln()`
  - Contains all elements on a Web page
    - Including forms created with the `<form>` element

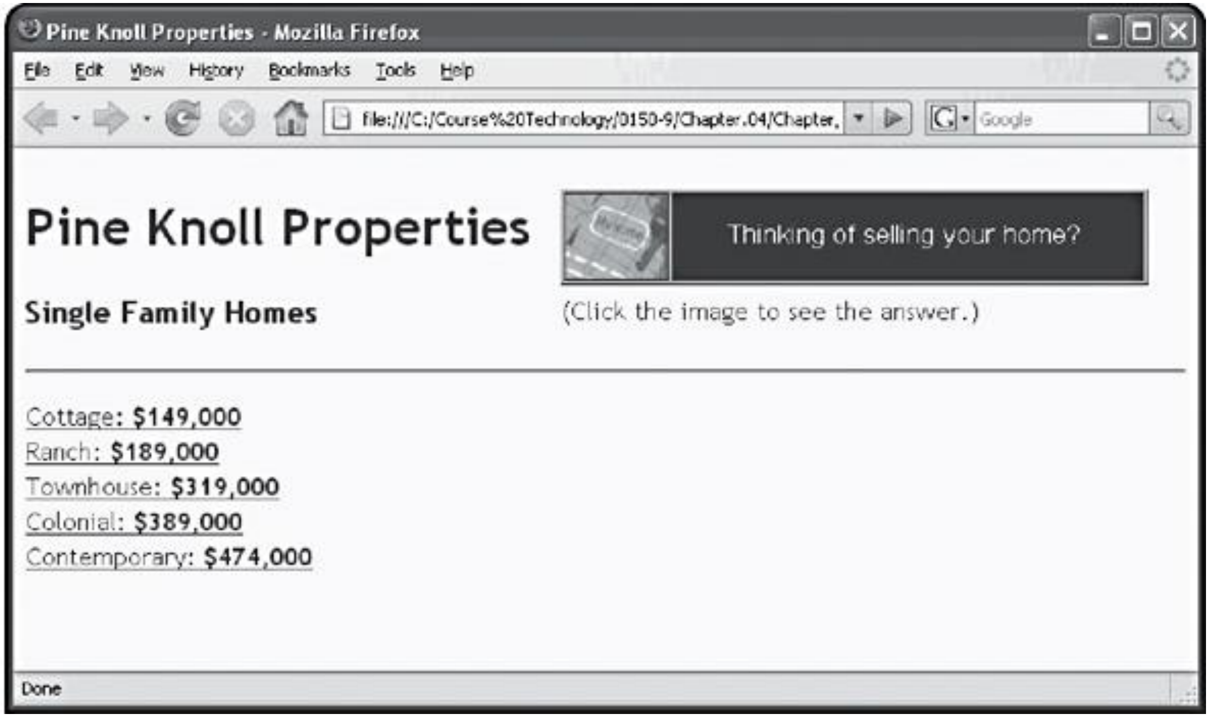
Referencing JavaScript Objects

- Some of the objects in the browser object model represent arrays
  - Such as `frame[]`, `forms[]`, or `images[]`
- To refer to a JavaScript object in code
  - You must refer to all of the objects that contain it, with the object names, separated by periods
- Example: Pine Knoll Properties Web site
  - Six prewritten Web pages: PineKnollProperties.html, colonial.html, contemporary.html, cottage.html, ranch.html, and townhouse.html

Referencing JavaScript Objects (continued)

Figure 4-3

Pine Knoll Properties Web page with an advertisement



Pine Knoll Properties - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///C:/Course%20Technology/0150-9/Chapter.04/Chapter.
Google
Done

Pine Knoll Properties

Thinking of selling your home?

(Click the image to see the answer.)

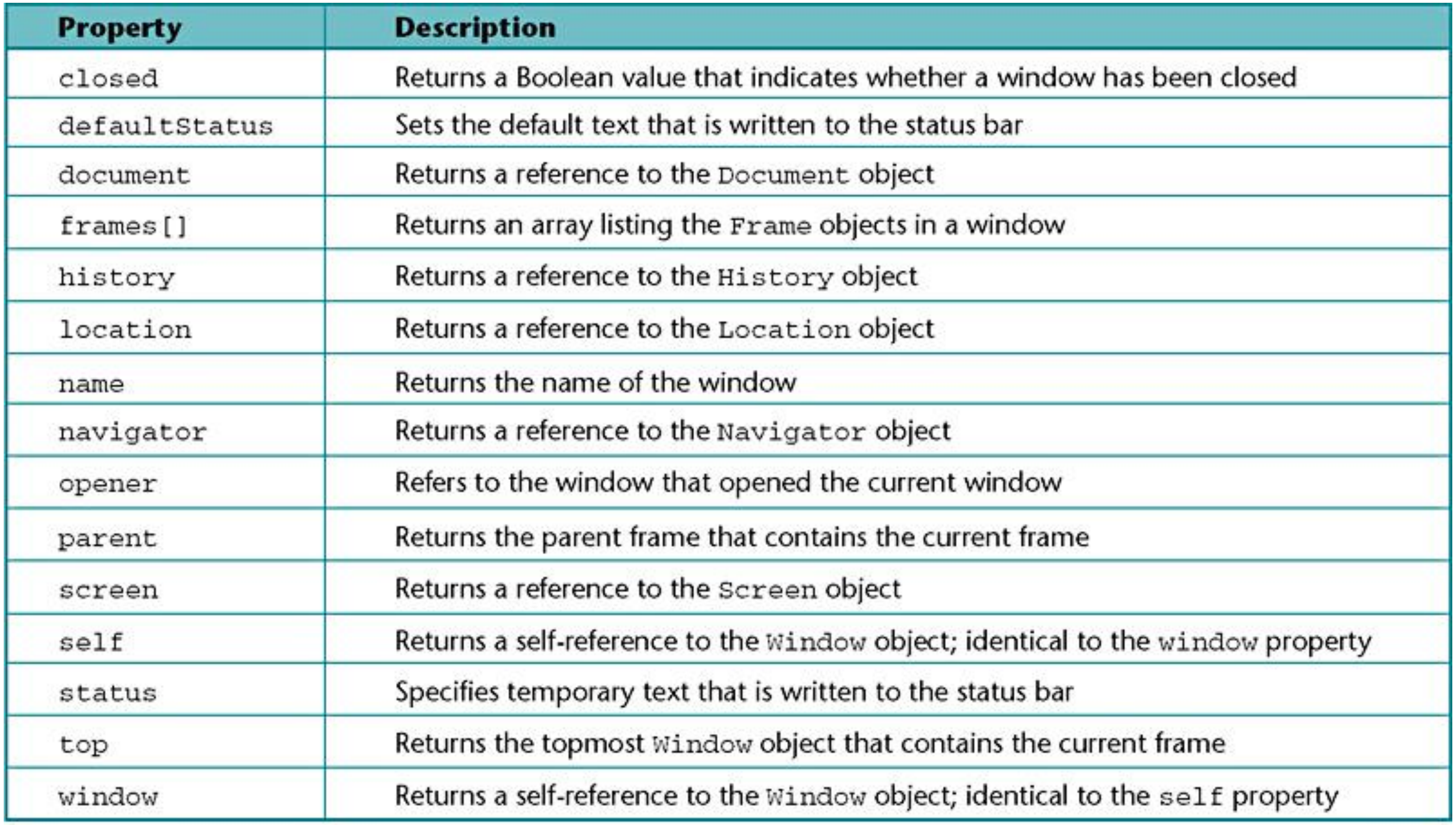
Single Family Homes

Cottage: **$149,000**
Ranch: **$189,000**
Townhouse: **$319,000**
Colonial: **$389,000**
Contemporary: **$474,000**

The Window Object

- Window object
  - Includes several properties that contain information about the Web browser window
  - Contains methods that allow you to manipulate the Web browser window itself
- **self property**
  - Refers to the current Window object
  - Using the self property is identical to using the window property to refer to the Window object
- Web browser assumes you are referring to the global object

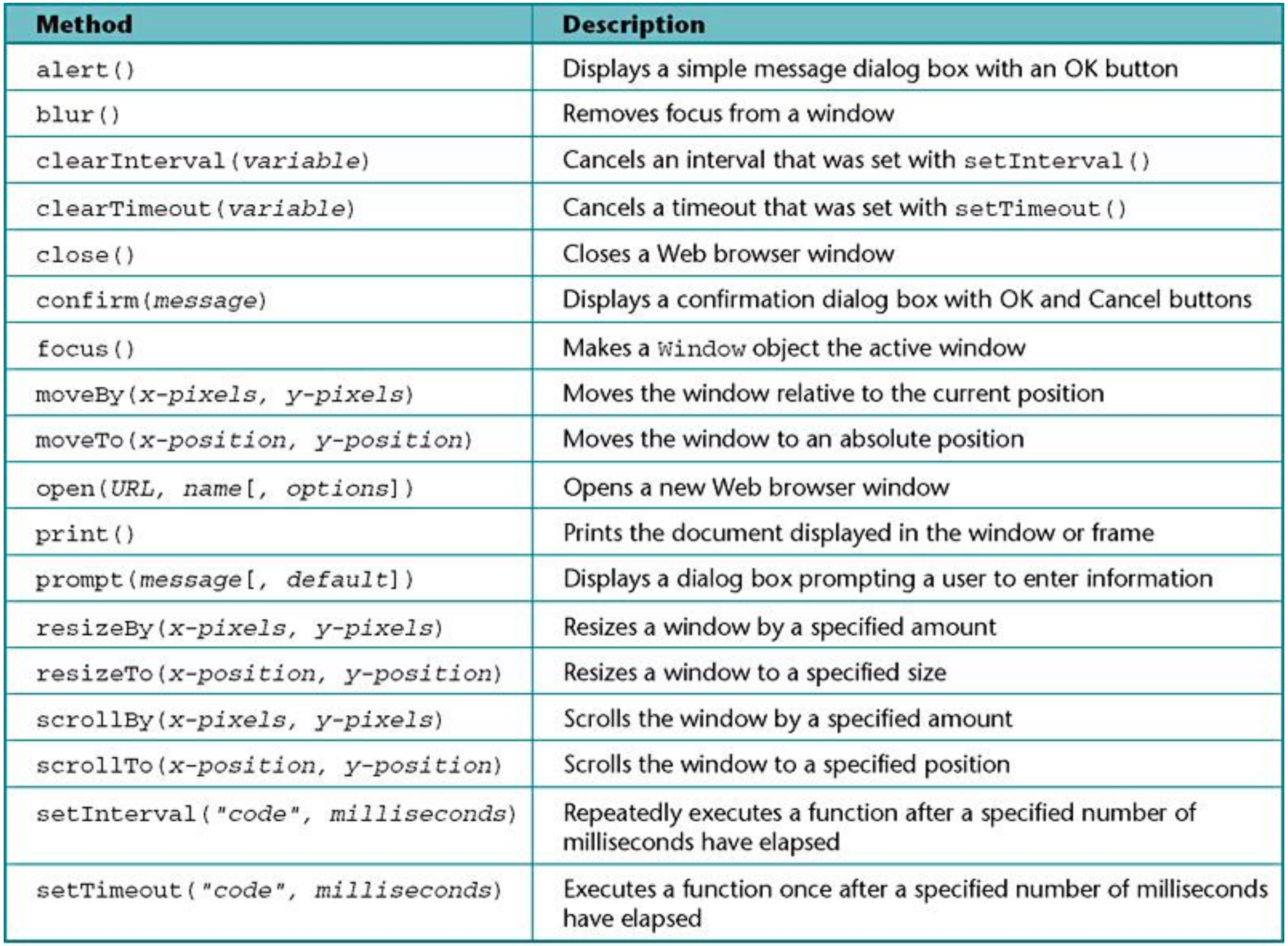
The `Window` Object (continued)



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
`parent`	Returns the parent frame that contains the current frame
`screen`	Returns a reference to the `Screen` object
`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

Table 4-1 Window object properties

The Window Object (continued)



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

Table 4-2 Window object methods

Windows and Events

- The `click` and `dblclick` Events
  - The `click` event is often used for anchor element
  - Web browser handles execution of the `onclick` event handler automatically
  - You can override an anchor element’s automatic `onclick` event handler
    - Add to the `<a>` element an `onclick` event handler that executes custom code
  - The `dblclick` event works like the `click` event

Windows and Events (continued)

- The `mouseover` and `mouseout` Events
  - Use the `mouseover` and `mouseout` events to create rollover effects
  - **Rollover** is an effect that occurs when your mouse moves over an element
  - `mouseover` event occurs when the mouse passes over an element
  - `mouseout` event occurs when the mouse moves off an element
  - One common use is to change the text that appears in a Web browser status bar

Windows and Events (continued)

- `defaultStatus` property
  - Specifies the default text that appears in the status bar whenever the mouse is not positioned over a link
- Example: Pine Knoll Properties Web site
  - Add the `defaultStatus` property
- Common use of rollovers is to replace (or swap) an image on a Web page
- The `mousedown` and `mouseup` events
  - `mousedown` event occurs when you point to an element and hold the mouse button down

Windows and Events (continued)

- The `mousedown` and `mouseup` events (continued)
  - `mouseup` event occurs when you release the mouse button
- Example: Pine Knoll Properties Web site
  - Modify the `<img>` element in the PineKnollProperties.html document
    - So the second image in the banner displays when you hold the mouse over it

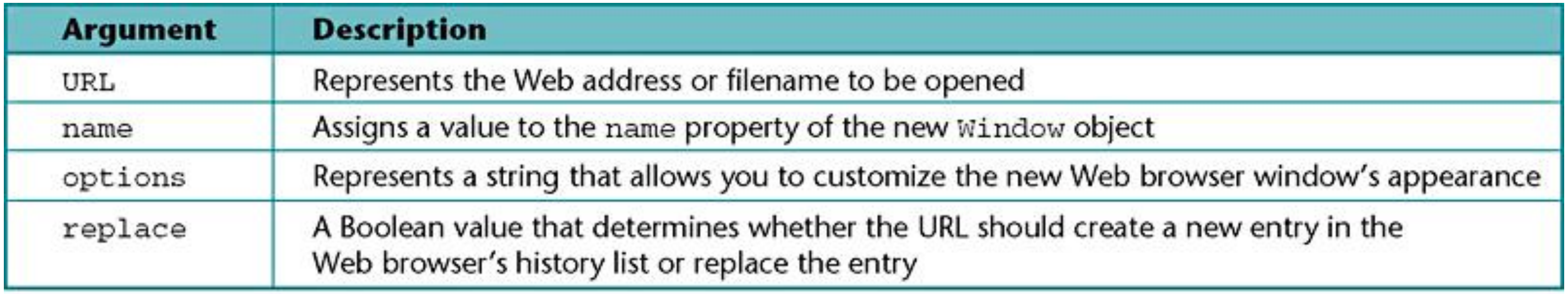
Opening and Closing Windows

- When a new Web browser window is opened
  - A new `Window` object is created to represent the new window
- Be familiar with how to open a link in a new window by using the `<a>` element’s target attribute
- Example: Pine Knoll Properties Web site
  - Open links in new windows
- Opening a Window
  - **`open()` method** of the `Window` object
    - Opens new windows in the strict DTD

Opening and Closing Windows (continued)

- Opening a Window (continued)
  - Syntax

```
window.open(url, name, options, replace);
```

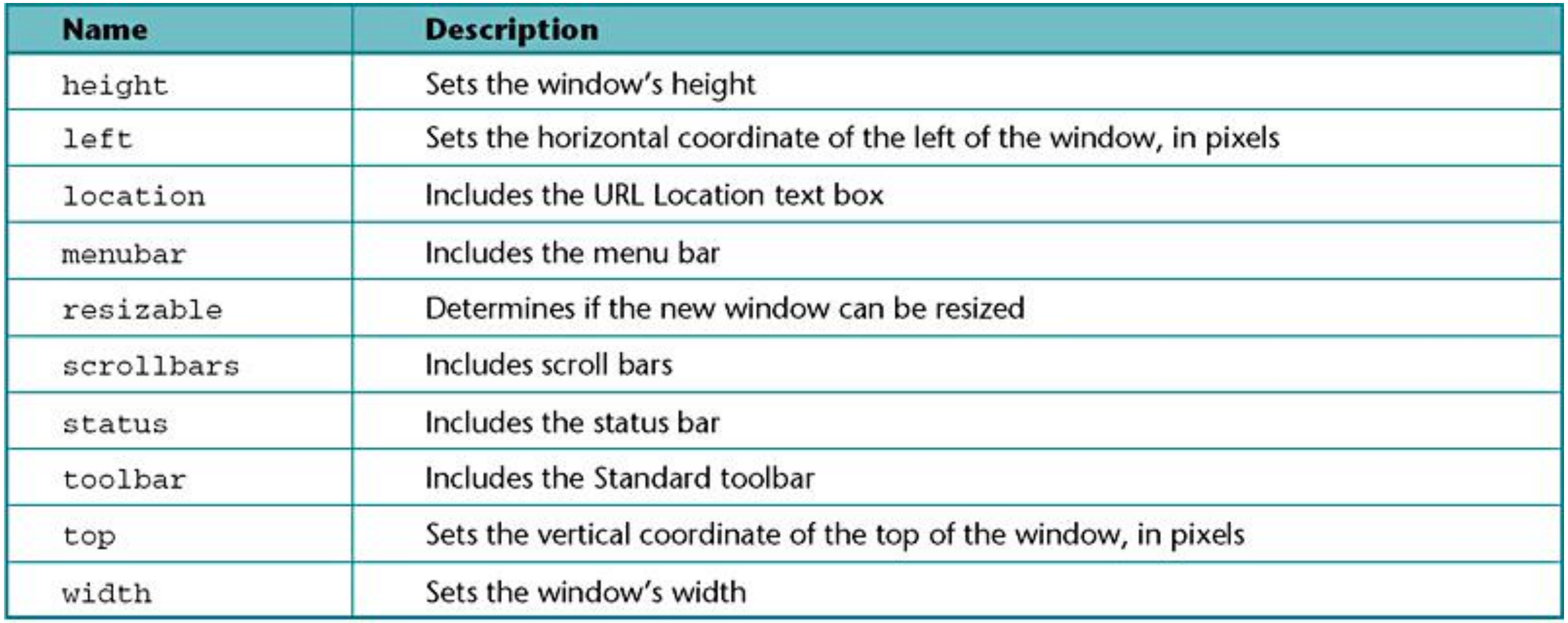


Argument	Description
`URL`	Represents the Web address or filename to be opened
`name`	Assigns a value to the `name` property of the new `Window` object
`options`	Represents a string that allows you to customize the new Web browser window's appearance
`replace`	A Boolean value that determines whether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

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

  - You can customize its appearance using the `options` argument

Opening and Closing Windows (continued)



Name	Description
`height`	Sets the window's height
`left`	Sets the horizontal coordinate of the left of the window, in pixels
`location`	Includes the URL Location text box
`menubar`	Includes the menu bar
`resizable`	Determines if the new window can be resized
`scrollbars`	Includes scroll bars
`status`	Includes the status bar
`toolbar`	Includes the Standard toolbar
`top`	Sets the vertical coordinate of the top of the window, in pixels
`width`	Sets the window's width

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

Opening and Closing Windows (continued)

- Opening a Window (continued)
  - Example: Pine Knoll Properties Web page
    - Links use the `window.open()` method instead of the `target` attribute to open the URLs in a separate page
  - A `Window` object's name property can be used only to specify a target window with a link
    - And cannot be used in JavaScript code
  - Assign created window to a variable
    - If you want to control it
  - **`focus()` method**
    - Makes a window the active window

Opening and Closing Windows (continued)

- Opening a Window (continued)
  - Example: Pine Knoll Properties Web page
    - Add a `focus()` method to `showProperty()`
- Closing a Window
  - **`close()` method**
    - Closes a Web browser window
  - `window.close()` or `self.close()`
    - Closes the current window
  - Example: Pine Knoll Properties Web page
    - Add links to each of the property Web pages that call the `close()` method

Working with Timeouts and Intervals

- `Window` object’s timeout and interval methods
  - – Creates code that executes automatically
- **`setTimeout()` method**
  - – Executes code after a specific amount of time
  - – Executes only once
  - – Syntax

```
var variable = setTimeout("code",
  milliseconds);
```

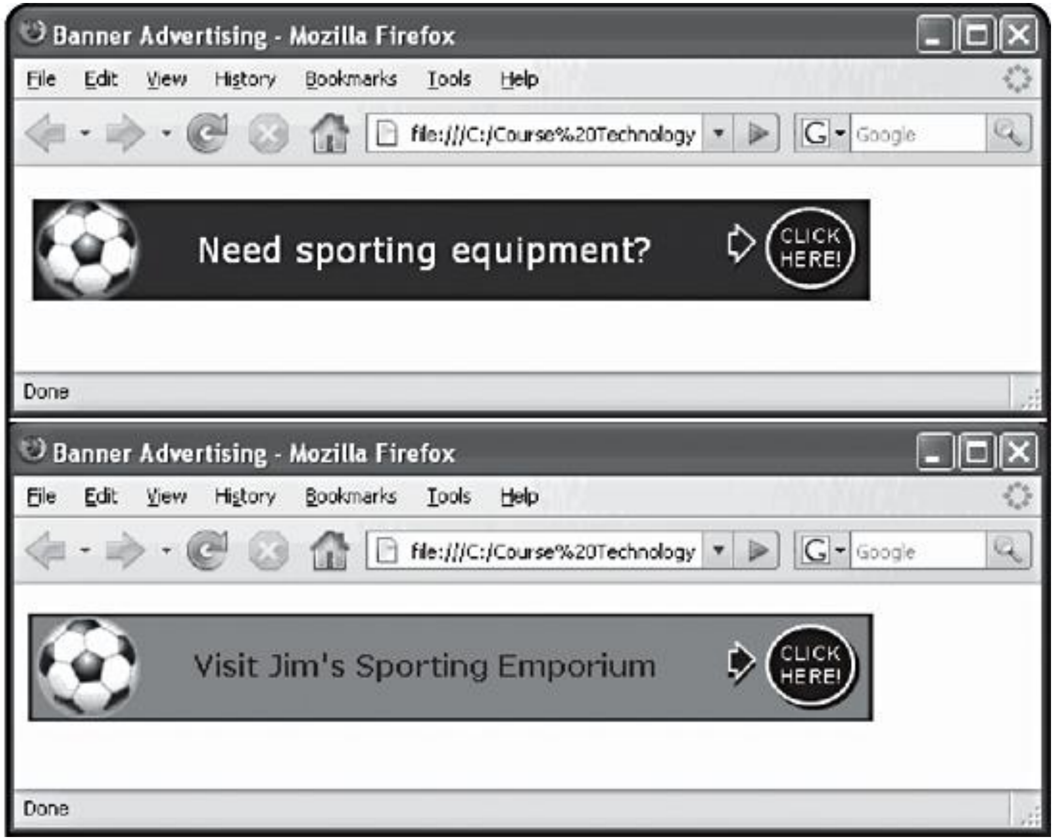
Working with Timeouts and Intervals (continued)

- **`clearTimeout()` method**
  - Cancel a `setTimeout()` before its code executes
- **`setInterval()` method**
  - Repeatedly executes the same code after being called only once
- **`clearInterval()` method**
  - Used to clear a `setInterval()` method call
- Interval methods are most often used for starting animation code

Working with Timeouts and Intervals (continued)

Figure 4-17

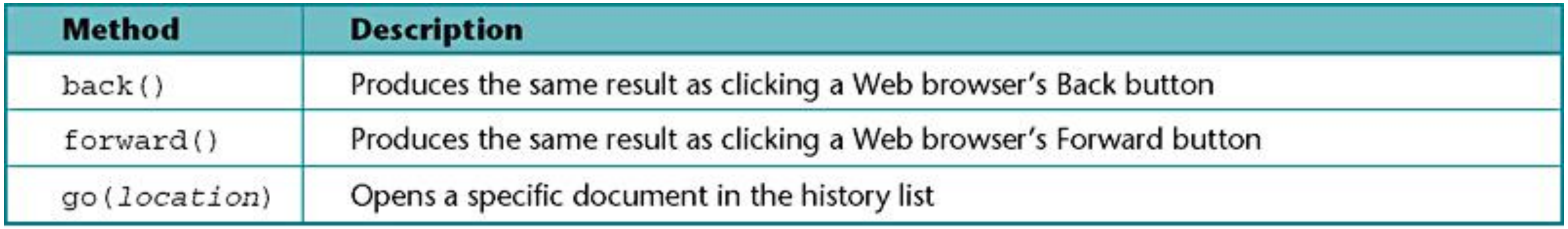
Advertising Images


Banner Advertising - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///C:/Course%20Technology
Google
Need sporting equipment?
CLICK HERE!
Done
Banner Advertising - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///C:/Course%20Technology
Google
Visit Jim's Sporting Emporium
CLICK HERE!
Done

The `History` Object

- **History object**
  - Maintains an internal list (history list) of all documents that were opened during current Web browser session
- Security features
  - Object will not actually display the URLs contained in the history list
  - In Internet Explorer this is only possible if the currently displayed Web page exists
    - In same domain as Web page containing JavaScript code

The History Object (continued)



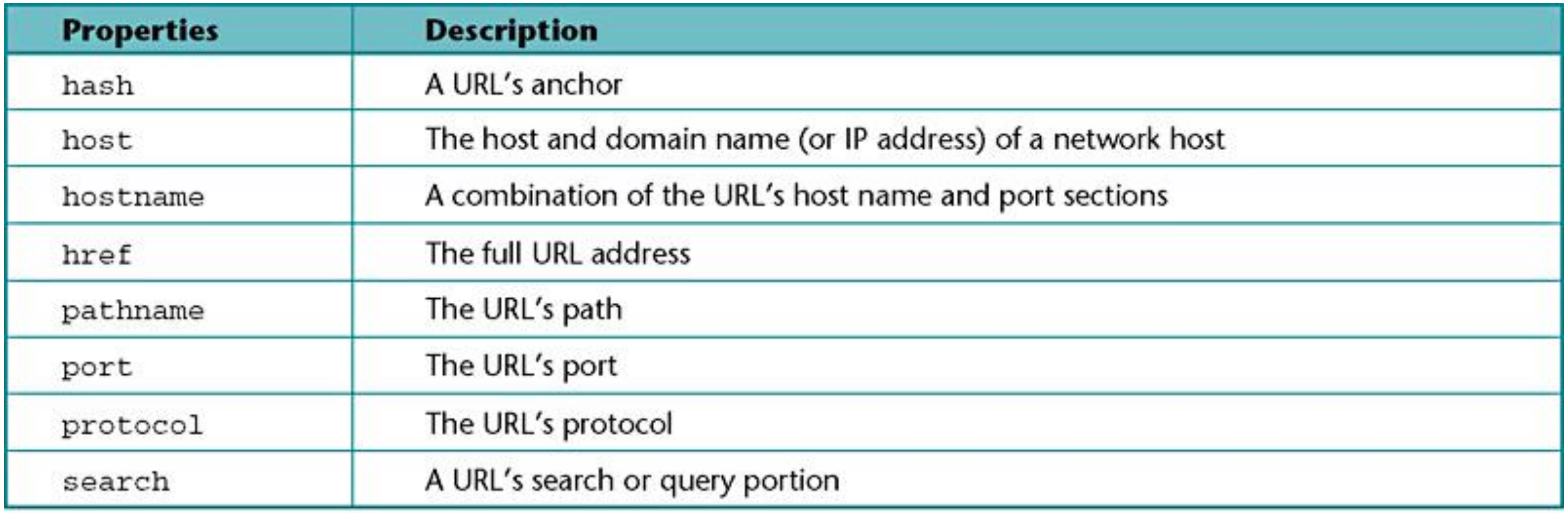
Method	Description
back()	Produces the same result as clicking a Web browser's Back button
forward()	Produces the same result as clicking a Web browser's Forward button
go(*location*)	Opens a specific document in the history list

Table 4-5 Methods of the History object

The `Location` Object

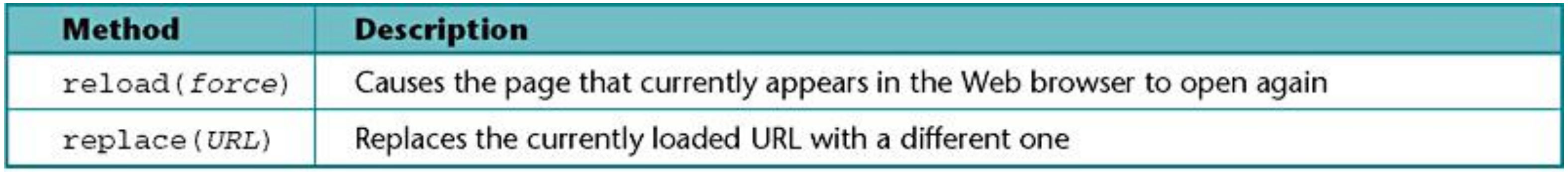
- **`Location` object**
  - Allows you to change to a new Web page from within JavaScript code
- Properties of the `Location` object allow you to modify individual portions of a URL
  - Web browser automatically attempts to open that new URL

The `Location` Object (continued)



Properties	Description
`hash`	A URL's anchor
`host`	The host and domain name (or IP address) of a network host
`hostname`	A combination of the URL's host name and port sections
`href`	The full URL address
`pathname`	The URL's path
`port`	The URL's port
`protocol`	The URL's protocol
`search`	A URL's search or query portion

Table 4-6 Properties of the `Location` object



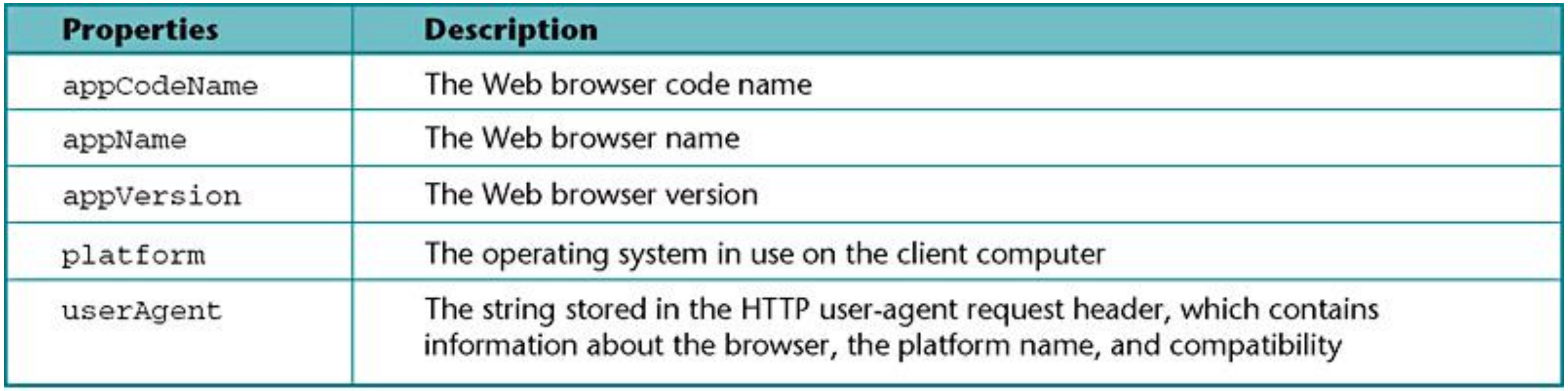
Method	Description
`reload(`*`force`*`)`	Causes the page that currently appears in the Web browser to open again
`replace(`*`URL`*`)`	Replaces the currently loaded URL with a different one

Table 4-7 Methods of the `Location` object

The `Navigator` Object

- **`Navigator` object**
  - To obtain information about the current Web browser
  - Can determine which type of Web browser is running
- **`with` statement**
  - Eliminates need to retype the name of an object
    - When properties of the same object are being referenced in a series

The `Navigator` Object (continued)



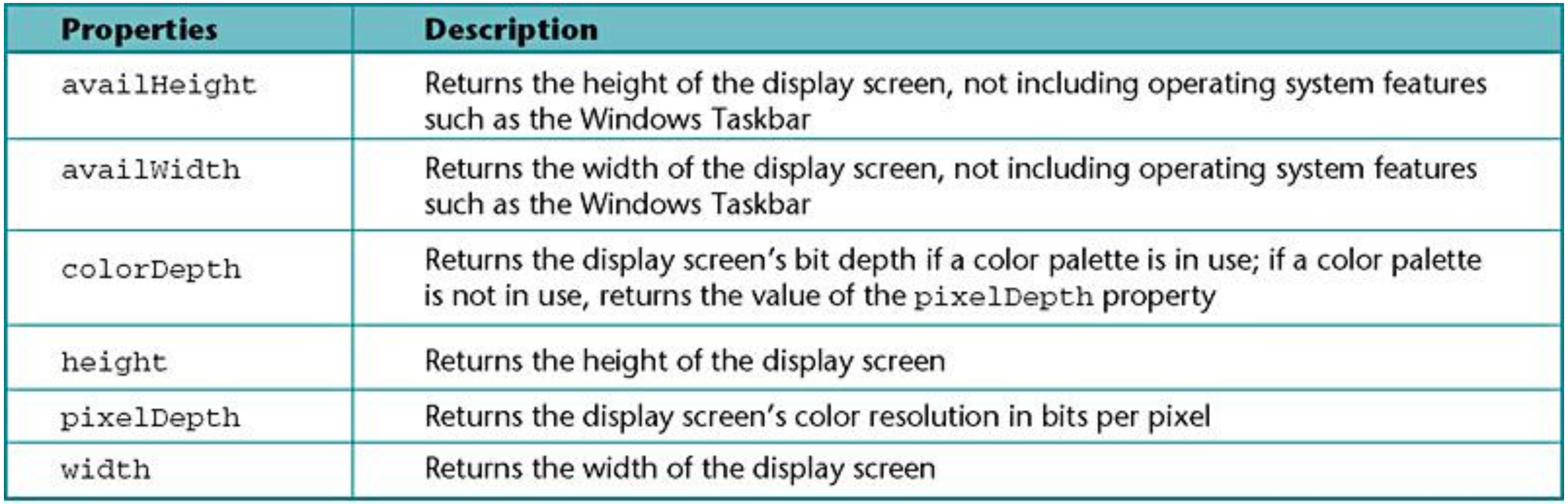
Properties	Description
`appCodeName`	The Web browser code name
`appName`	The Web browser name
`appVersion`	The Web browser version
`platform`	The operating system in use on the client computer
`userAgent`	The string stored in the HTTP user-agent request header, which contains information about the browser,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

Table 4-8 Properties of the `Navigator` object

The `Screen` Object

- **`Screen` object**
  - Obtains information about the display screen’s size, resolution, and color depth
- Common use of the `Screen` object properties
  - To center a Web browser window in the middle of the display area
- Example: Pine Knoll Properties Web page
  - Property window is centered in display area

The `Screen` Object (continued)



Properties	Description
`availHeight`	Returns the height of the display screen, not including operating system features such as the Windows Taskbar
`availWidth`	Returns the width of the display screen, not including operating system features such as the Windows Taskbar
`colorDepth`	Returns the display screen's bit depth if a color palette is in use; if a color palette is not in use, returns the value of the `pixelDepth` property
`height`	Returns the height of the display screen
`pixelDepth`	Returns the display screen's color resolution in bits per pixel
`width`	Returns the width of the display screen

Table 4-9 Properties of the `Screen` object

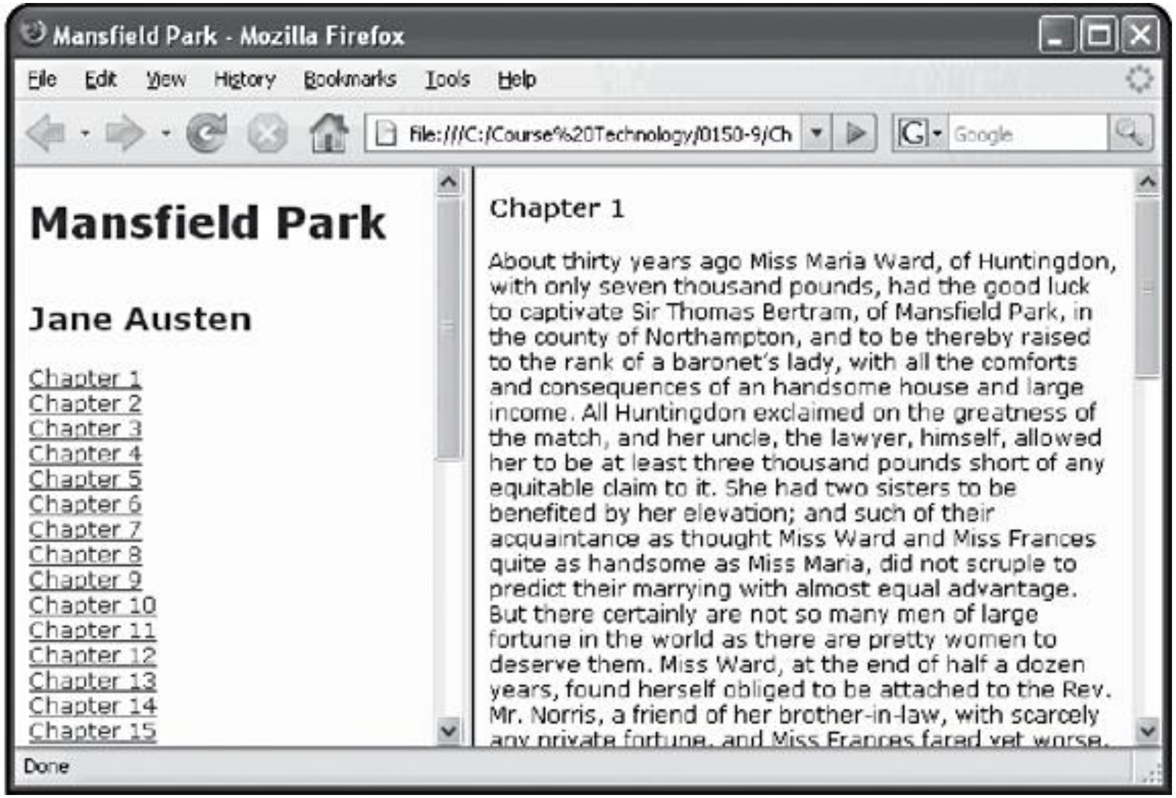
Referring to Frames and Windows

- Learn how to refer to frames and windows from within Web pages

Using the `target` and `base` Attributes

Figure 4-20

Mansfield Park Web page


Mansfield Park - Mozilla Firefox
File
Edit
View
History
Bookmarks
Tools
Help
file:///C:/Course%20Technology/0150-9/Ch
Google
Done

Mansfield Park

Jane Austen

Chapter 1
Chapter 2
Chapter 3
Chapter 4
Chapter 5
Chapter 6
Chapter 7
Chapter 8
Chapter 9
Chapter 10
Chapter 11
Chapter 12
Chapter 13
Chapter 14
Chapter 15

Chapter 1

About thirty years ago Miss Maria Ward, of Huntingdon, with only seven thousand pounds, had the good luck to captivate Sir Thomas Bertram, of Mansfield Park, in the county of Northampton, and to be thereby raised to the rank of a baronet's lady, with all the comforts and consequences of an handsome house and large income. All Huntingdon exclaimed on the greatness of the match, and her uncle, the lawyer, himself, allowed her to be at least three thousand pounds short of any equitable claim to it. She had two sisters to be benefited by her elevation; and such of their acquaintance as thought Miss Ward and Miss Frances quite as handsome as Miss Maria, did not scruple to predict their marrying with almost equal advantage. But there certainly are not so many men of large fortune in the world as there are pretty women to deserve them. Miss Ward, at the end of half a dozen years, found herself obliged to be attached to the Rev. Mr. Norris, a friend of her brother-in-law, with scarcely any private fortune, and Miss Frances fared yet worse.

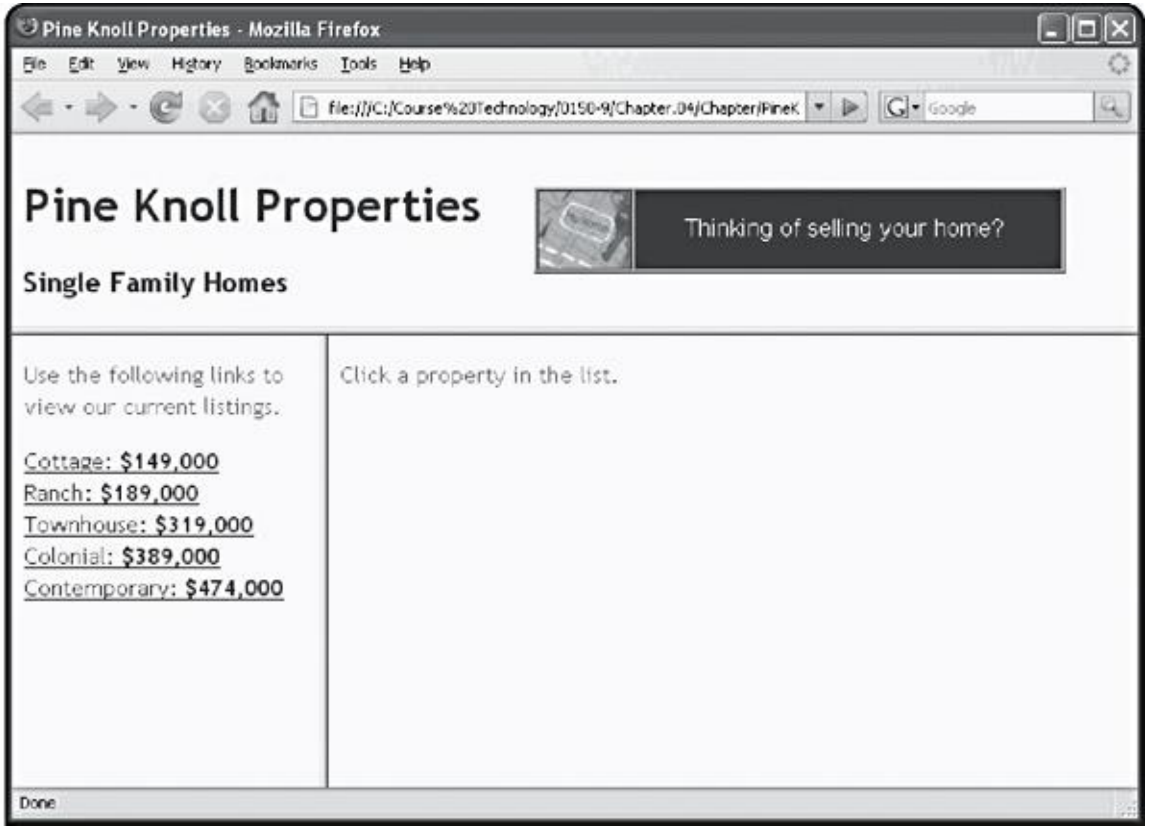
Using the `target` and `base` Attributes (continued)

- **`target` attribute**
  - Determines in which frame or Web browser window a document opens
    - Based on the value assigned to an `<a>` element's target attribute
    - Or the value assigned to a `<frame>` element's name attribute
- Example: Pine Knoll Property Web page
  - Work on a prewritten, frame-based version

Using the `target` and `base` Attributes (continued)

Figure 4-22

Frame-based version of the Pine Knoll Properties Web page



Pine Knoll Properties - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///C:/Course%20Technology/0150-9/Chapter.04/Chapter/PineK
Google
Thinking of selling your home?
Done

Pine Knoll Properties

Single Family Homes

Use the following links to view our current listings.

Cottage: $149,000
Ranch: $189,000
Townhouse: $319,000
Colonial: $389,000
Contemporary: $474,000

Click a property in the list.

Using the `target` and `base` Attributes (continued)

- **`<base>` element**
  - Used with the `target` attribute
  - Specifies a default target for all links in a document, using the assigned name of a window or frame
- You must use the transitional DTD
- Example: Pine Knoll Properties Web page
  - Modify the PropertiesList.html document so it includes a `<base>` element instead of multiple target attributes

The `parent` Property

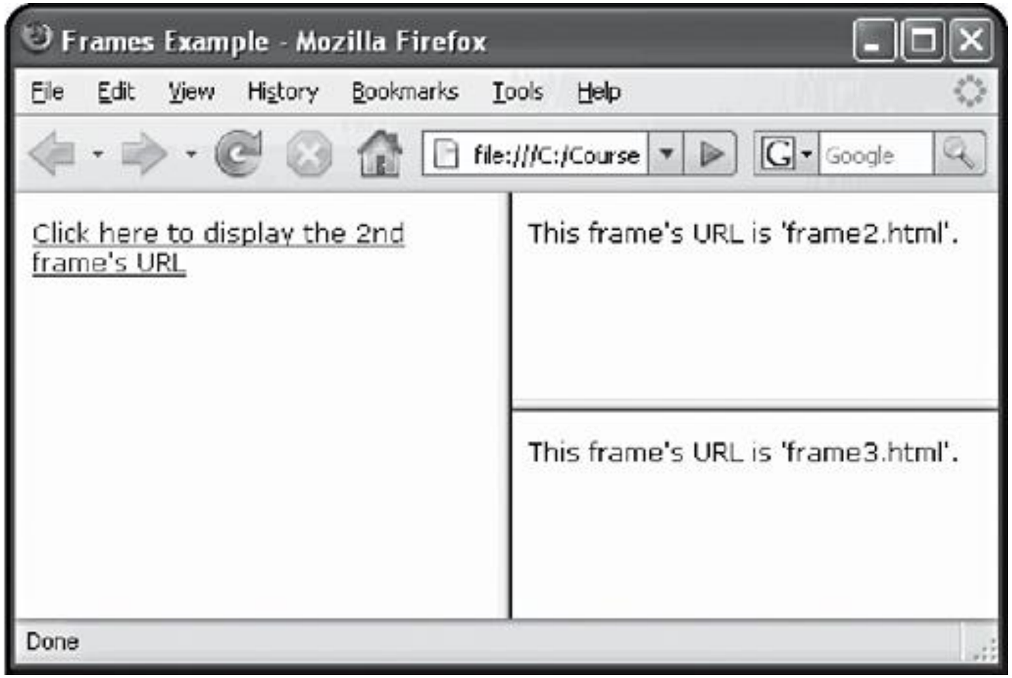
- `frames[]` array contains all frames in a window
- **`parent` property** of the `Window` object
  - – Refers to a frame within the same frameset
    - Combined with the frame’s index number
- With nested frames
  - – Use the `parent` property and the name you assigned to a frame with the `<frame>` element
  - – Nested frames are assigned to the `frames[]` array in the order in which they are encountered

37

The `parent` Property (continued)

Figure 4-25

Referencing a nested frame



Frames Example - Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///C:/Course
Google
Done

Click here to display the 2nd frame's URL

This frame's URL is 'frame2.html'.

This frame's URL is 'frame3.html'.

The `parent` Property (continued)

- Example: Pine Knoll Properties Web page
  - – Modify the PropertiesList.html document so the links open in the right frame

The `top` Property

- **`top` property**
  - Refers to the topmost window on a Web page
- When working with frames, the `top` property refers to the window that constructed the frames
- Example: Pine Knoll Properties Web page
  - Modify the PropertiesList.html document so the links are opened using the `top` property instead of the `parent` property

Summary

- Browser object model (BOM) or client-side object model is a hierarchy of objects
- Top-level object in the browser object model is the `Window` object
- To refer to a JavaScript object in code, you must refer to all of the objects that contain it
- `Document` object is arguably most important object
- `History` object maintains a history list of all the documents that have been opened

Summary (continued)

- `Location` object allows you to change to a new Web page from within JavaScript code
- `Navigator` object obtains information about the current Web browser
- `with` statement eliminates the need to retype the name of an object
- `Screen` object obtains information about the display screen’s size, resolution, and color depth

Summary (continued)

- `target` attribute determines in which frame or Web browser window a document opens
- Use the `target` attribute with the `<base>` element to specify a default target for all links
- To refer to a frame within the same frameset, use the `parent` property
- `top` property refers to the topmost window on a Web page