

## Tutorial 10

### Designing Pop-Up Windows and Frames



## Objectives

- Explore the window object
- Work with the status bar
- Work with the history and location objects
- Create new browser windows
- Work with pop-up blockers
- Apply methods to the window object



## Objectives (cont'd)

- Create dialog boxes
- Exchange data between windows
- Work with frames
- Control frame behavior
- Work with inline frames

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*



## Exploring the Window Object

- JavaScript considers the browser window an object, which it calls the **window object**
- To set a property of the window object:  
`windowObject.property = value`

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*



## Properties of the Window Object

Property	Description
closed	A Boolean value indicating whether the window has been closed
defaultStatus	The default message displayed in the status bar
document	The document object displayed in the window
frames	The collection of frames within the window
history	The history object, containing a list of Web sites visited within the window
innerHeight	The inner height of the window excluding all toolbars, scroll bars, and other features (not supported under Internet Explorer)
innerWidth	The inner width of the window excluding all toolbars, scroll bars, and other features (not supported under Internet Explorer)
location	The location object containing the URL of the current Web document
name	The name of the window
opener	The source browser window, which opened the current window
outerHeight	The outer height of the window including all toolbars, scroll bars, and other features (not supported under Internet Explorer)
outerWidth	The outer width of the window including all toolbars, scroll bars, and other features (not supported under Internet Explorer)
scrollbars	The scroll bar object contained in the browser window
status	The temporary or transient message displayed in the status bar
statusbar	The status bar object used for displaying messages in the browser window
toolbar	A Boolean value indicating whether the window's toolbar is visible

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*



## Working with the Status Bar

- **Chrome**
  - Collective term that refers to borders of a browser window (e.g., toolbars and scroll bars)
- **Status bar**
  - Part of the chrome that is common to all browsers
  - Displays messages (either permanent or transient) about actions occurring within the window

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*



## Working with the Status Bar

- To change default message displayed in browser window's status bar, use JavaScript command:  
`windowObject.defaultStatus = value`  
where *windowObject* is an object reference to the browser window and *value* is text of status bar message
- To change transient or temporary message appearing in status bar, add these commands to the event handler function:

```
    windowObject.status = value;  
    return true;
```

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*



## Working with the History and Location Objects

- **Location object** contains information about the page currently displayed in the window
- **History object** holds a list of sites the Web browser displayed before reaching current page in the window

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*



## Working with the History and Location Objects

- Moving forward and backward in the history list:
  - To navigate history list within current browser window:  
`history.back();`  
`history.forward();`
  - To use history objects in other browser windows:  
`windowObject.history.back();`
  - To navigate to a particular page in the history object:  
`history.go(integer)`

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*



## Working with the History and Location Objects

- Automatic page navigation
  - To redirect users automatically from one page to another, add the meta element:  
`<meta http-equiv="Refresh" content="sec;URL=url" />`
  - To redirect using JavaScript, run the code:  
`setTimeout( function() {  
 location.href = url;  
}, milliseconds);`
  - To avoid client-side redirection, run a server-side script that modifies HTTP 301 status code of HTTP header

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*



## Creating New Browser Windows

- **Secondary window or pop-up window**
  - Window that opens in addition to main browser window
- Opening new windows with HTML
  - For a link to open the target document in a new window, specify window name using target property:  
`<a href="url" target="name">link text</a>`

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*



## Creating New Browser Windows

- Opening new windows with JavaScript
  - Offers more control and more options for creating new windows than target property does
  - To create a pop-up browser window:  
`window.open(url,name,features)`
  - To store a pop-up window as a JavaScript object:  
`var windowObject = window.open(url, name, features)`

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*





## Creating New Browser Windows

- Setting the features of a pop-up window
  - To define width and height of the pop-up window, add:  
`width=x,height=y`
  - *Internet Explorer*: To set screen position of pop-up, add:  
`left=x,top=y`
  - *Other browsers*: To set screen position of a pop-up, add:  
`screenX=x,screenY=y`
  - To display directory buttons, location box, menu bar, scroll bar, status bar, or toolbar, add:  
`directories=value,location=value,menubar=value,scrollbars=value,status=value,toolbar=value`

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*



## Creating New Browser Windows

- Working with **pop-up blockers**
  - Prevent pop-up windows from opening
  - Most block only automated pop-ups, not those activated by a user clicking a link to a pop-up window
  - Use JavaScript to determine whether pop-up window has been opened; if not, JavaScript can open the linked file in the current browser window
- Add a pop-up window to a web site with the `popWin()` function

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*



## Working with Window Methods

Method	Description
<code>blur()</code>	Removes the focus from the window
<code>close()</code>	Closes the window
<code>focus()</code>	Gives the window the focus
<code>moveBy(dx, dy)</code>	Moves the window <i>dx</i> pixels to the right and <i>dy</i> pixels down
<code>moveTo(x, y)</code>	Moves the top left corner of the window to the screen coordinates ( <i>x</i> , <i>y</i> )
<code>print()</code>	Prints the contents of the window
<code>resizeBy(dx, dy)</code>	Resizes the window by <i>dx</i> pixels to the right and <i>dy</i> pixels down
<code>resizeTo(x, y)</code>	Resizes the window to <i>x</i> pixels wide and <i>y</i> pixels high
<code>scrollBy(dx, dy)</code>	Scrolls the document content in the window by <i>dx</i> pixels to the right and <i>dy</i> pixels down
<code>scrollTo(x, y)</code>	Scrolls the document in the window to the page coordinates ( <i>x</i> , <i>y</i> )

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*



## Working with Window Methods

- The **self** keyword
  - Refers to current window; synonymous with window keyword
  - To close currently active window, use either:  
`self.close();`      or      `window.close();`
- The **opener** keyword
  - Refers to window or frame that used the `window.open()` method to open current window
  - To close initial or first window:  
`opener.close();`

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*





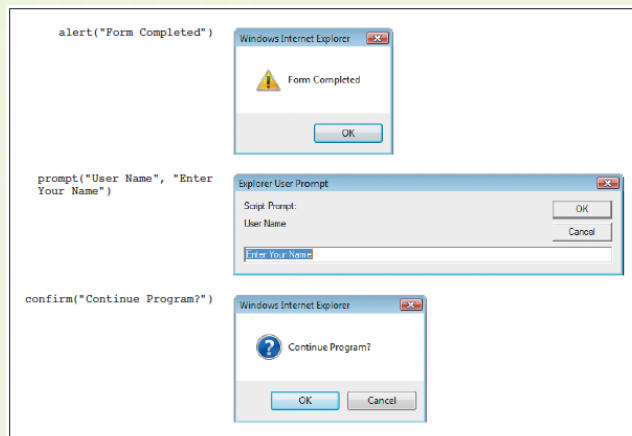
## Creating Dialog Boxes

- JavaScript supports three types of customized dialog boxes
  - To display an **alert dialog box** using windows method:  
`alert(message)`
  - To display a **prompt dialog box**:  
`prompt(message, default)`
  - To display a **confirm dialog box**:  
`confirm(message)`

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*



## Creating Dialog Boxes



*New Perspectives on  
JavaScript and AJAX, 2nd Edition*



## Exchanging Data Between Windows

- To write text to a browser window:  
`windowObject.document.write(content);`  
`windowObject.document.close();`
- To access an object within a browser window:  
`windowObject.document.getElementById(id)`
- To access a variable defined within a browser window:  
`windowObject.variable`
- To access a function defined within a browser:  
`windowObject.function()`

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*



## Exchanging Data Between Windows

- Creating a customized pop-up window
  - Parameters of the writeContent() function
    - The windowObj parameter
    - The choice parameter
    - The guess parameter

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*



## Modal and Modeless Windows

- Dialog boxes and windows can be either modal or modeless
  - A **modal window** prevents users from doing work in any other window or dialog box until the window is closed (e.g., alert dialog box in a browser window)
  - A **modeless window** allows users to work in other windows and dialog boxes even as that window stays open (e.g., pop-up windows)

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*

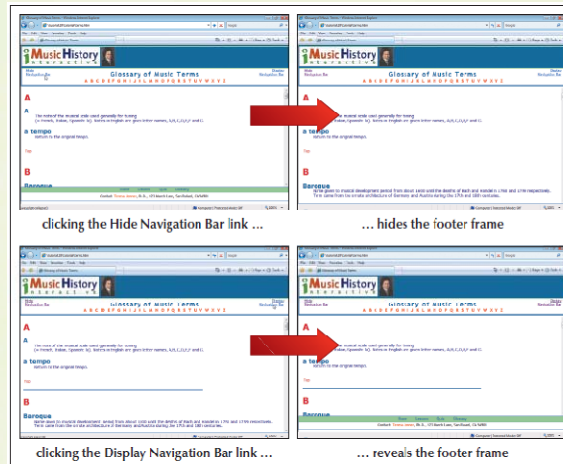


## Working with Frames

- Frames enable division of browser window into multiple panes; each frame displays a separate document
- Each panel is known as a frameset and is created using HTML code where each frame displays a separate URL
- A link that alternately hides and displays a frame provides more screen space to other frames

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*

# Working with Frames



*New Perspectives on  
JavaScript and AJAX, 2nd Edition*

# Working with Frames

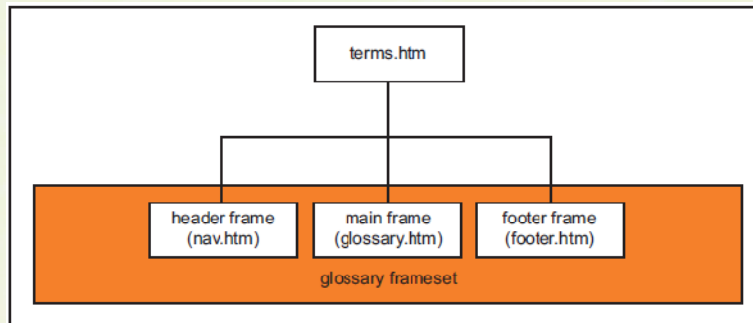
- Working with the frame and frameset objects
  - To reference a particular frame in a frameset:  
`windowObject.frames[idref]`  
where *windowObject* is the browser window containing the frameset and *idref* is the id or index number of a frame in the frames collection
  - To reference a frameset element:  
`document.getElementById(id)`  
where *id* is the value of the id attribute in the `<frameset>` tag

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*



## Working with Frames

- Navigating between frames
  - JavaScript treats frames in a frameset as elements in a hierarchical tree



*New Perspectives on  
JavaScript and AJAX, 2nd Edition*



## Working with Frames

- Navigating between frames
  - To access contents of one frame from another, you must navigate through the hierarchy of objects using two keywords: **parent keyword** and **top keyword**
    - When frames are direct children of the browser window, parent and top keywords are synonymous

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*



## Working with Frames

- Treating frames as windows
  - To change a frame's source document:  
`frameObject.location.href = url;`
  - To write content into a frame:  
`frameObject.document.write(content);`  
`frameObject.document.close();`
  - To reference the document displayed within a frame:  
`frameObject.document.getElementById(id)`

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*



## Working with Frames

- Setting the frameset layout
  - Use rows and cols properties of frameset object:  
`frameset.rows = text`  
`frameset.cols = text`
  - To remove or hide a frame from a frameset, set its height (or width) value to 0 pixels

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*





## Working with Frames

- Collapsing and expanding a frame
  - The `collapse()` function reduces the height of the footer frame
    - Setting the height of the footer frame to 1 pixel (rather than 0 pixels) preserves the footer frame in the layout, but essentially hides it from the user
  - The `expand()` function resets the footer frame's height to its original 64 pixels

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*



## Controlling Frame Behavior

- To block unwanted frame:

```
if (top.location.href != self.location.href) {  
    top.location.href = self.location.href;  
}
```
- To force a page into a frameset:

```
if (top.location.href == self.location.href) {  
    top.location.href = url;  
}
```

*New Perspectives on  
JavaScript and AJAX, 2nd Edition*



## Working with Inline Frames

- Inline frames appear as a separate window within the document
- To access the window object containing an inline frame document:

*frameObject*.documentWindow

where *frameObject* is the object reference to the inline frame

- To access the document within an inline frame:

*frameObject*.documentWindow.document

New Perspectives on  
JavaScript and AJAX, 2nd Edition



## Working with Inline Frames

- To access a variable defined within a document in an inline frame:

*frameObject*.documentWindow.*variable*

where *variable* is the name of a global variable defined within the inline frame document

- To access a function defined within a document in an inline frame:

*frameObject*.documentWindow.*function*()

where *function* is the name of a function defined within the inline frame document

New Perspectives on  
JavaScript and AJAX, 2nd Edition