

Unit 8

DHTML

Structure:

8.1 Introduction

Objectives

Introduction to DHTML

Document Object Model

8.2 Event Handling & Data Binding

Event

Event Handlers

Delegation event model

Data Binding

8.3 Browser Object Models

8.4 Summary

8.5 Terminal Questions

8.6 Answers

8.7 References

8.1 Introduction

In the previous unit, you have studied about the Cascading Style Sheets. Now you are comfortable with using CSS for styling and layout. You have also studied different tags of HTML and proper usage of different HTML tags. In this unit we are going to discuss about dynamic HTML. When we will think about dynamic HTML, we need to remember the qualities of standard HTML. Especially that once a page is loaded from the server, it will not change until another request comes to the server.

Dynamic HTML gives us more control over the HTML elements and allows them to change at any time, without returning to the Web server. DHTML, is an umbrella term for a collection of technologies used together to create interactive and animated web sites by using a combination of a static markup language which is a client-side scripting language (such as JavaScript), a presentation definition language (such as CSS), and the Document Object Model (DOM).

In this unit you are going to learn advantages of DHTML, document object model, Events, types of events, Event handlers, delegation event model, data binding and finally browser object models.

Objectives:

After studying this unit, you will be able to:

- explain dynamic HTML
- describe the concept of DOM
- explain the types of events.
- describe the delegation event model.
- define data binding
- explain browser object models

8.1.1 Introduction to DHTML

DHTML stands for dynamic hypertext markup language. As we have already discussed about HTML in previous units. Dynamic HTML is not really a new specification of HTML, but rather a new way of looking at and controlling the standard HTML codes and commands.

DHTML is not a language but a term used for a combination of technologies – HTML, CSS, JavaScript, and DOM (document object model), which are used together to create dynamic and interactive web pages.

DHTML requires four independent components to work: HTML, Cascading Style Sheets, Scripting and the Document Object Model. Following figure 8.1 shows DHTML contain HTML, CSS, Dom and scripting.

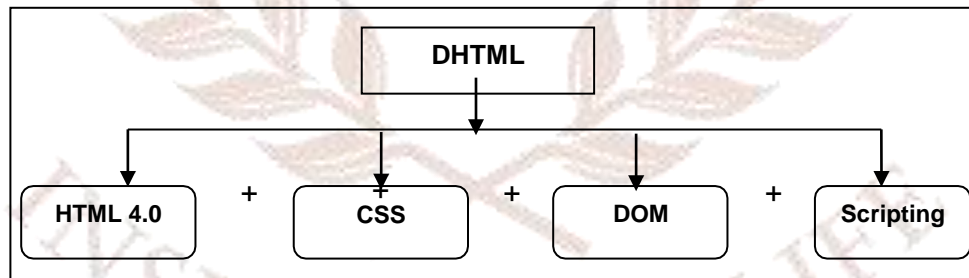


Figure 8.1: Components of DHTML

1. **HTML:** HTML defines the structure of a Web page, using such basic elements as headings, forms, tables, paragraphs and links. (We have already discussed about HTML in previous units)
2. **Cascading Style Sheets (CSS):** Cascading Style Sheets allows authors and users to attach style (like; fonts, spacing, colors, back grounds) to structured documents by separating content of documents and the

presentation style of documents. CSS allows web developers to control the style and layout of web pages. HTML 4 allows a dynamic change to CSS (we have already discussed about CSS in previous unit).

3. **Scripting:** Scripting allows us to access and dynamically controls the individual properties of both HTML and CSS. JavaScript is the most compatible scripting language for browser, so we use JavaScript (we will discuss about JavaScript in next units).
4. **Document Object Model (DOM):** DOM is an Application Programming Interface (API). It provides a platform where programming languages can "talk" to HTML and XML documents. Usually the programming language that does the talking by using JavaScript. The work of DOM is to represent all the attributes of HTML and Style sheets to JavaScript (JavaScript is the most compatible scripting language for browser, so we use JavaScript.) control. The DOM allows you to treat individual elements on the page as objects, which can then be manipulated using JavaScript.

HTML vs. DHTML

Dynamic HTML is an extension of HTML that enables, among other things, the inclusion of small animations and dynamic menus in Web pages. DHTML code makes use of style sheets and JavaScript.

When you see an object, or word(s), on a webpage that becomes highlighted, larger, a different color, or a streak runs through it by moving your mouse cursor over it is the result of adding a DHTML effect. This is done in the language coding and when the file of the webpage was saved it was saved as the .dhtml format instead of .htm or .html.

DHTML sites are dynamic in nature. DHTML uses client side scripting to change variables in the presentation which affects the look and function of static page. DHTML characteristics are the functions while a page is viewed, rather than generating a unique page with each page load (a dynamic website).

On the other hand, HTML is static. HTML sites rely solely upon client-side technologies. This means the pages of the site do not require any special processing from the server side before they go to the browser. In other words, the pages are always the same for all visitors - static. HTML pages have no dynamic content.

Advantages of DHTML

1. **Supported by all browsers:** DHTML is supported in every major browser, including Opera, Safari, Internet Explorer, and Firefox.
2. **Open standards:** You can create your pages according to the standardized technologies of DHTML and you can expect that, they will display much the same on any major browser.
3. **Small file sizes:** Like HTML, we create DHTML with files, which are smaller than graphic files and generally render faster than alternatives such as flash and java.
4. **Change contents on the fly:** When we have loaded our DHTML page then also we can make the necessary changes to the web page without having to reload it. This is where the dynamic in DHTML comes from.
5. **No plug-ins required:** If a browser support HTML, CSS, JavaScript and the DOM it supports DHTML without the need for any additional plug-ins.
6. **Easy to learn:** It is easy to learn DHTML. For DHTML no need to learn an entire Java programming language. Although DHTML is based on JavaScript, which is now a full featured programming language, you do not have to have a degree in computer science to use the basics.
7. **Data binding:** Microsoft developed this to allow easier access to databases from Web sites. It is very similar to using a CGI to access a database, but uses an ActiveX control to function. This feature is very advanced and difficult to use for the beginning DHTML writer.

Real-time positioning: When most people think of DHTML this is what they expect. Objects, images, and text moving around the Web page. This can allow you to play interactive games with your readers or animate portions of your screen.

8.1.2 Document Object Model

The name "Document Object Model" was chosen because it is an "Object Model" in the traditional object oriented design sense: documents are modeled using objects, and the model encompasses not only the structure of a document, but also the behavior of a document and the objects of which it is composed. The object model is the mechanism that makes DHTML programmable.

Basically “DOM is a platform and language neutral interface that allows programs and scripts to dynamically access and updates the content, structure, and style of a document.” The DOM is separated into different levels:

DOM Levels

DOM Level 1

The DOM core, which provides a way to map the structure of an XML based document to allow for easy access and manipulation of any part of a document, and the DOM HTML, which extended the DOM core by adding HTML specific objects and methods.

DOM Level 2

DOM Level 2 introduced the following new modules of the DOM to deal with new types of interfaces:

DOM Views: Describes interfaces to keep track of the various views of a document.

DOM events: Describes interfaces for events and event handling

DOM style: describes interfaces to deal with CSS-based styling of elements.

DOM traversal and range: Describes interfaces to traverse and manipulate a document tree.

DOM Level 3

DOM level3 further extends the DOM with the introduction of methods to load and save documents in a uniform way as well as methods to validate a document (DOM validation). In level3, the DOM Core is extended to support all of XML1.0, including XML Infoset, XPath and XML Base.

XML (extensible markup language) is a markup language much like HTML. XML is designed to transport and store data.

DOM Support in Web Browsers

The DOM had been a standard for some time before web browsers started implementing it. Internet Explorer made its first attempt with version 5, but it didn't have any realistic DOM support until version 5.5, when it implemented most of DOM Level 1.

For Netscape, no DOM support existed until Netscape 6 (Mozilla 0.6.0) was introduced. After Netscape 7, Mozilla switched its development efforts to the Firefox browser. Firefox 3 supports all of Level 1, nearly all of Level 2, and some parts of Level 3.

The following table shows DOM support for popular browsers:

Table 8.1: DOM supported browsers

Browser	DOM Compliances
Netscape 6+(Mozilla 0.6.0+)	Level 1, Level 2 (almost all), Level 3 (partial)
Internet Explorer 5	Level 1 (minimal)
Internet Explorer 5.5 - 7	Level 1 (almost all)
Opera 9+	Level 1, Level 2 (almost all), Level 3 (partial)
Chrome 0.2+	Level 1, Level 2 (partial)

Document Object Methods

The document object contains several methods used for dynamically generating web pages and manipulating elements. Below table list the methods of the document object that are specified in the W3C DOM.

Table 8.2: DOM supported methods

Method	Description
getElementsByName(name)	Returns a collection of elements represented by name.
getElementByTagName(tag name)	Returns a collection of elements represented by tag name.
getElementById(ID)	Returns the element represented by ID.
open()	Opens a new window document in a window.
write(text)	writes a new text to a document
close()	closes a new document that was created with the open() method.

Self Assessment Questions

1. DHTML sites are _____ in nature.
2. _____ defines a standard set of objects for any structured document.

8.2 Event handling & Data Binding

8.2.1 Event

Events are things that occur in the system that you are programming. The system generates a signal of some sort when an event takes place and provides a method by which an action can be done automatically when the event occurs. Events are fired inside the browser window and frequently relate to a particular item that is present there. The entire browser window, the HTML document open in the current tab, a single element, a group of elements, or the full document may be included. There are numerous types of occurrences that might happen.

For instance:

- The user chooses a key on the keyboard.
- The user selects, clicks, or moves the pointer over a certain element.

Types of Events

The following two categories are used to group the events:

1. The Foreground Events: The user must directly interact with the foreground events in order to take place. User interaction with the GUI component results in these kinds of events being produced. For instance, pushing a keyboard key, clicking a button, moving the mouse, choosing an item from a list, etc.
2. The Background Events: The Background events are those that happen as a result of end-user activity. For instance, an operating system can stop a system from failing (Hardware or Software). We require an event handling mechanism with control over the events and reactions to manage these events.

8.2.2 Event handlers

Most changes made in object's styles with the DOM should be triggered by an event handler. Events are an important part of DHTML. Suppose, DHTML code reacts to something that the user did: The content's color changed when user moves mouse, a hidden elements show itself when user

click a specific link, or a list of words popped up that match the keystrokes the user made.

In this section, you will learn about events and how to handle them. The most common way to add the event is to add the event handler's name and the code you want to execute the HTML tag's attributes.

Ex:

```
<Html>
<Body>
<a href = "somepage.html" name= "link page" onmouseover = "alert ('you moved') " Onclick= "alert ('you clicked?') " >
</a>
</body>
</html>
```

The following table shows the lists of different events supported by DOM.

Table 8.3: List of events

Event	Description
FocusEvent	Provides event properties that specific to focus event
KeyboardEvent	Provides event properties that specific to keyboard event
MessageEvent	Provides event properties that specific to onmessage event
Mouse Event	Provides event properties that specific to mouse events
PopStateEvent	Fires when a history entry changes
Storage Event	Provides event properties that specific to the onstorage events
TextEvent	Provides event properties that specific to text events

8.2.3 Delegation Event Model

The Delegation Event model was created to manage events in GUI programming languages. The GUI, or graphical user interface, is where a user interacts with the system graphically or visually. Every user initiated action, such as a mouse movement, a click, a scroll, etc., is referred to as an event and is mapped to a code to provide functionality to the user. This is the nature of GUI programming. It is referred to as event handling.

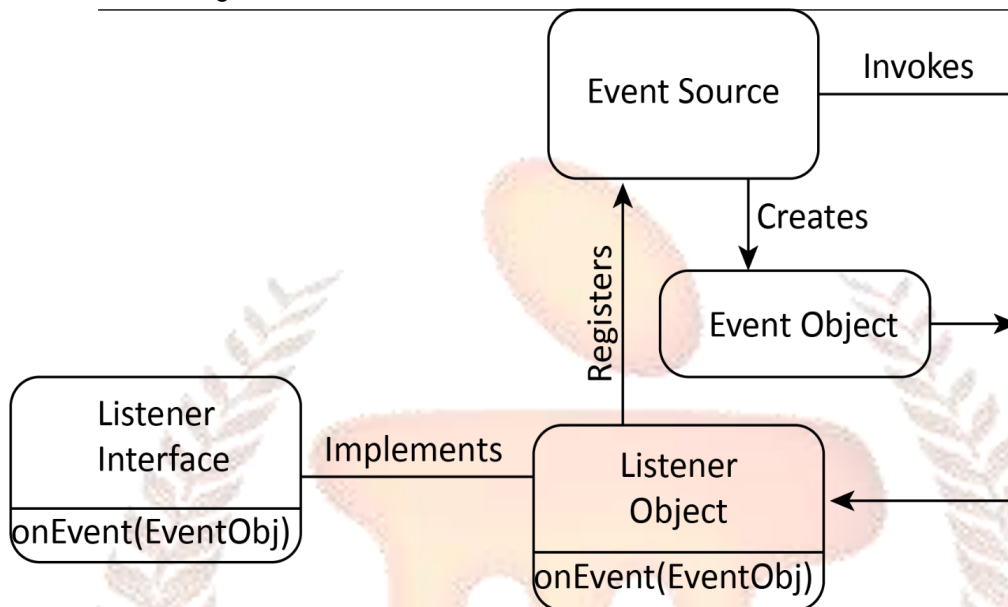


Figure 8.2: Delegation Event Model

The Delegation Model serves as the foundation for contemporary event processing techniques. It specifies a uniform and compatible system for creating and handling events. A source creates an event in this model and sends it to one or more listeners. The listener waits for an event to arrive before acting. The listener processes the event after it is received and then returns it. The processing of an event might be assigned by the UI components to a different function.

The following three elements form the basis of an event model:

- **Events :** The objects that define state change in a source are called Events. A user's response to interacting with GUI elements can result in the generation of an event. Moving the mouse pointer, clicking on a button, pushing a keyboard key, choosing an item from a list, and other actions are examples of event generation activities.
- **Events Sources:** A source is something that produces and causes an event. As the object's internal state changes, it produces an event. The sources are permitted to provide a variety of events. To get alerts for a given event, a source must register a listener. Each event specifies how attendees can register.
- **Events Listeners:** An object that is called when an event occurs is known as an event listener. The listeners need two things in order to function. First, they need to be registered with a source, however they can register with multiple sources if they want to. The second step is for it to put the ways for obtaining and handling notifications into practise.

8.2.4 Data Binding

Data binding is a process that allows an Internet user to manipulate Web page elements using a Web browser. It employs dynamic HTML (hypertext markup language) and does not require complex scripting or programming. Data binding first became available with Microsoft Internet Explorer (MSIE) version 4. It can be used in conjunction with that and all subsequent versions of MSIE to create and view interactive Web sites with a minimum demand on authoring time, subscriber effort, server drive space, and server processing resources.

Data binding is based on a component architecture that consists of four major pieces: the data source object (DSO), data consumers, the binding agent, and the table repetition agent. Data source objects provide the data to a page, data-consuming HTML elements display the data, and the agents ensure that both the provider and the consumer are synchronized. Following diagram 8.2 shows the four component architecture of data binding.

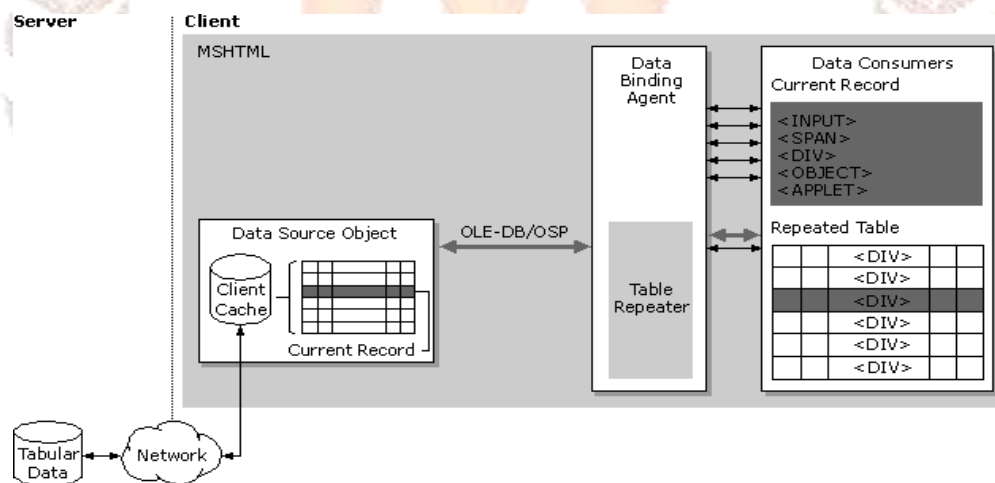


Figure 8.3: Data Binding Architecture

How to create a DSO (Data Source Object)

A data source object (DSO) is a Microsoft ActiveX object embedded within a Web page. It employs data binding, in which an ActiveX control communicates directly with another Web page, or with an external XML data source. A Database object in the Stone field Query data dictionary contains information about a logical database. A Data Source object defines the connection information for a physical instance of the logical database. There are basically two ways to create DSO.

1. Tabular Data Control (TDC)

Tabular Data Control is a Microsoft ActiveX control that will sort through a data file, extrapolate the necessary information, and produce that information in specified areas on the Web page. For example, if we have a text file that contains three fields and these fields are delimited by a character, then it is possible to extract the values of the fields into an HTML page.

The ActiveX control is initialized using the <OBJECT> tag. The CLASSID (unique identifier) for the tabular data control is: CLSID: 333C7BC4-460F-11D0-BC04-0080C7055A83.

Example: <INPUT TYPE=TEXTBOX DATASRC="#dsoComposers" DATAFLD="compsr_last">

Example 2:

```
<OBJECT CLASSID="clsid: 333C7BC4-460F-11D0-BC04-0080C7055A83"
ID=dsoComposer WIDTH=0 HEIGHT=0>
<PARAM NAME="DataURL" VALUE="C:\Users\we\Desktop\data.csv">
</OBJECT>
<HTML>
<Span><INPUT TYPE=TEXTBOX Value="Manipal" id= text1
Datascr="#dsoComposer" datfld="City"></Span>
<input type=button value="click me" id=button1
onclick=dsoComposer.recordset.MoveNext ()>
</HTML>
```

When we will run this example it will fetch the data from data.csv, when a button click it will pick the next value from table (record list).

2. Remote Data Service (RDS)

Remote Data Services (RDS) is a Microsoft technology used in combination with ActiveX Data Objects (ADO) that allowed the retrieval of a set of data from a database server, which the client then altered in some way and then sent back to the server for further processing. RDS supports with Internet Explorer 4.0 and later. RDS obtains its data from a database using OLE DB or Open Database Connectivity (ODBC). We can use RDS if:

- You have any existing database like SQL Server, Microsoft Access, or Oracle.

- You want to specify the data using a Structured Query Language (SQL) statement.
- You want to run update, insert, and delete commands from your database.
- You need to direct, real-time access to the data.

The RDS is initialized using the <OBJECT> tag. The CLASSID (unique identifier) for the Remote Data Service is: CLSID: BD96C556-65A3-11D0-983A-00C04FC29E33.

Example:

```
<OBJECT classid="clsid:BD96C556-65A3-11D0-983A-00C04FC29E33"
ID=dsoForRDS HEIGHT=0 WIDTH=0>
<PARAM NAME="Server" VALUE="http://manipal.com">
<PARAM NAME="Connect" VALUE="dsn=manipal;uid=guest;pwd=">
<PARAM NAME="SQL" VALUE="select std_name from students">
</OBJECT>
```

In the above example first, note the class identifier (CLSID) specific to RDS. Every ActiveX component requires a CLSID to differentiate it from other objects registered on the system. As we know RDS is capable of retrieving and updating data from any OLE DB or ODBC-compliant database. In the above given example of RDS we have used 3 parameters name like server, connect, SQL.

In the server parameter, string that we have given identifies the protocol and the server that supplies the data.

In the connect parameter, ODBC connection string identifying the datasource name configured on the server.

SQL query identifying the table and columns to be selected from the database. According to the above example it will select the entire students name from a student table.

How to bind the data?

Basically there are many ways to bind a data but in this unit we are going to discuss only 2 effective and frequently used techniques to bind the data.

1. Binding a Single-Valued Element to Data
2. Binding a TABLE to Data

1. **Binding a Single-Valued Element to Data:** The purpose of single value

binding is to connect one property to another by tying them together. The binding will be achieved by an event handler which assigns the data given by the event to the target property. Therefore it is necessary for the source event to fire a change event or some other kind of data event. The single value binding is mostly a basis for the higher concepts of the data binding. The procedure for binding a single-valued element to data is the same regardless of the element. Elements can be bound to data at design time using the DATASRC and DATAFLD attributes, or DHTML Object Model Support for Data Binding using the dataSrc and dataFld properties exposed by the corresponding objects in the Document Object Model (DOM).

Example: `<INPUT TYPE=TEXTBOX DATASRC="#dsostudents" DATAFLD="std_name">`

In this example the dataSrc (#dsostudents) attribute specifies the ID, prefixed by a hash mark (#), of a DSO embedded on the page. The hash mark is required. The dataFld (std_name) attribute identifies the field in the data provided by the DSO to which the text box should be bound.

2. Binding a Table to data

We can bind a table also to data. Here we can display full table contents or whatever contents we want on HTML page not only a single value. Because the table element is a tabular data consumer, it relies on the elements that it contains to bind to the individual fields in the data set provided by the DSO. The contained elements serve as a template, and they are repeated once for each record in the data set. The table specifies the dataSrc attribute. The contained elements specify the dataFld attribute and inherit the dataSrc from the table.

Example:

```
<TABLE DATASRC="#dsodata">
<TR>
<TD><SPAN DATAFLD="Std_id"></SPAN></TD>
<TD><SPAN DATAFLD="Std_name"></SPAN></TD>
<TD><SPAN DATAFLD="Std_age"></SPAN></TD>
</TR>
</TABLE>
```

In this example, the dataSrc(#dsodata) attribute specifies the ID, prefixed by a hash mark (#), of a DSO embedded on the page. The hash mark is

required. The dataFld (std_id, std_name, std_age) attribute identifies the field in the data provided by the DSO to which the table row/column should be bound. When we will execute this code it will fetch the student id, name and age in the table on our HTML page.

Self Assessment Questions

3. _____ is a Microsoft technology used in combination with ActiveX Data Objects (ADO).
4. _____ is a Microsoft ActiveX control that will sort through a data file, extrapolate the necessary information, and produce that information.
5. DSO stands for _____.

8.3 Browser Object Model

In section 8.3 we have discussed about Document Object Model. The DOM represents the document displayed in a browser. But there are also circumstances in which we need to access information about the browser itself. DOM is the part of BOM. The Browser Object Model is the one part of JavaScript for which no formal standards exist. It is also the part that is generally the least useful and the part which newbies most want to work with. DHTML refers to the idea of generating Web content dynamically. It relies very heavily on user input.

BOM, the Browser Object Model, describes the way in which we program the different objects made available to use in a browser. In the following figure 8.3 we are going to show standard browser object hierarchy.

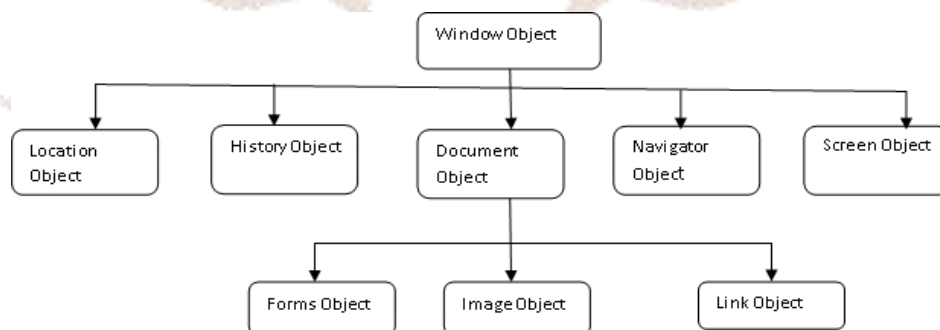


Figure 8.4: Standard Browser Objects Hierarchy

1. **Window Object:** The window object is the root of the object hierarchy. Window object as an object contains other child objects, as well as methods. Many of the properties of the window object can be used to carry out simple tasks, such as displaying messages, prompting for input, and loading new documents into the window. For example, you can display a simple modal message box from the current window using the alert method, then and changing the text in the Internet Explorer status bar using the status property, there are basically three window methods extensively: window.alert(), window.prompt() & window.confirm(). In the following example we have used all the three methods together.

Example:-

```
<input type=button value="click me" id=button1 onclick=button1()>
<SCRIPT language="VBSCRIPT">
sub button1
msgbox "Manipal Education"
window.alert("alert me")
var1=window.prompt("prompt")
window.confirm("confirm")
msgbox(var1)
end sub
</SCRIPT>
```

Status and default Status

As with the DOM, we can read/write various parts of the BOM. The status line at the base of a browser screen can be read or written to using the window.Status property. If the status line is left unattended, the default Status property is displayed.

Example:

```
<HTML>
<HEAD>
<TITLE> BOM Status </TITLE>
</HEAD>
<BODY onload="defaultStatus='Date appears here';">
```

```
<P onmouseover="status=new Date ();">Distance Education</P>  
</BODY>  
</HTML>
```

In this example, when the document is first loaded, we change the default Status property to 'Date appears here'. This is the text that appears in the status line when we are not engaging the status line elsewhere.

When the mouse is over the <P> element, we change the status line to display the current date. The status line is released from the code when the onmouseout event is fired, and this reverts the status line back to the default Status message.

- a) **Location object:** The location object is used to allow the programmers to write code to change the URL of the current page, effectively navigating to a different web page. We can do these one of two ways:
 - (i) First one can be done by invoking the window.location. replace() method. this method replaces the window currentdocument with a new one.
 - (ii) The second one will also replace the previous page in the browser's history.
- b) **The history object:** History object is the part of window object. It allows programmers to write code that navigates a web browser's history. There are basically three methods for history object like
 - (i) **history.go() method** this method takes an integer value to indicate whether to go back (a negative value) or forward (a positive value).
 - (ii) **history.back() and history.forward() methods** allow us to program code to navigate incrementally.
- c) **Document object:** Each HTML document loaded into a browser window becomes a document object. The Document object provides access to all HTML elements in a page, from within a script.
 - (i) **Form object.** In the form object the forms collection returns an array of all the forms in the current document or retrieves a collection, in source order, of all form objects in the document.
Syntax: document.forms[].property

- (ii) **Image object.** In the image object the images collection returns an array of all the images in the current document.
Syntax: document.images[].property
- (iii) **Link object.** In the link object the links collection returns an array of all the links in the current document.
Syntax: document.links[].property
- d) **Navigator object:** The navigator object contains information about the browser and it allows programmers to detect important information about the application the user runs to view a web page. There are some properties of navigator object. Following table 3.1 shows navigator object property.

Table 8.4: Navigator Object Properties

S. No.	Property	Description
1	appName	Returns the code name of the browser
2	appName	Returns the name of the browser
3	appVersion	Returns the version information of the browser
4	cookieEnabled	Determines whether cookies are enabled in the browser
5	platform	Returns for which platform browser is compiled
6	userAgent	Returns the user-agent header sent by the browser to the server

- e) **Screen object:** The screen object contains information about the visitor's screen. The screen object includes a group of useful attributes:
 - (i) **window.screen.width:** it returns the width of the screen.
 - (ii) **window.screen.height:** it retrieves the height of the workingarea of the system's screen, excluding the Windows taskbar.
 - (iii) **window.screen.colorDepth:** it retrieves the number of bits per pixel used for colors on the destination device or buffer.

Self Assessment Questions

6. There are basically three window methods extensively: window.alert(), window.prompt() and_____.

7. _____ allows programmers to write code that navigates a web browser's history.
8. _____ contains information about the browser and it allows programmers to detect important information about the application the user runs to view a web page.

8.4 Summary

- DHTML stands for dynamic hypertext markup language. Dynamic HTML is not really a new specification of HTML, but rather a new way of looking at and controlling the standard HTML codes and commands.
- DHTML sites are dynamic in nature. DHTML uses client side scripting to change variables in the presentation which affects the look and function of an otherwise static page.
- DOM is a platform and language neutral interface that allows programs and scripts to dynamically access and updates the content, structure, and style of a document.
- Most changes made in object's styles with the DOM should be triggered by an event handler.
- An event is created by a source and sent to one or more listeners. The listener waits for an event to arrive before acting. The listener processes the event after it is received and then returns it.
- Data binding is based on a component architecture that consists of four major pieces – the data source object (DSO), data consumers, the binding agent, and the table repetition agent.
- BOM, the Browser Object Model, describes the way in which we program the different objects made available to use in a browser.

8.5 Terminal Questions

1. Differentiate between HTML and DHTML?
2. Define the various advantages of DHTML?
3. List out the various levels of DOM?
4. Explain data binding and its architecture through a proper diagram?
5. Describe the browser object model using suitable diagram?
6. Briefly explain the Delegation event model

8.6 Answers

Self Assessment Questions

1. Dynamic
2. Core DOM
3. RDS
4. TDC
5. Data Source Object
6. Window. Confirm ().
7. History object
8. Navigator object

Terminal Questions

1. Dynamic HTML is an extension of HTML that enables, among other things, the inclusion of small animations and dynamic menus in Web pages. DHTML code makes use of style sheets and JavaScript. For more details refer sub section 8.1.1.
2. Supported by all browsers, open standards, small file sizes. For more details refer sub section 8.1.1.
3. Basically DOM is a platform and language neutral interface that allows programs and scripts to dynamically access and updates the content, structure, and style of a document. The DOM is separated into different levels: For more details Refer section 8.1.2.
4. Data binding is a process that allows an Internet user to manipulate Web page elements using a Web browser. It employs dynamic HTML (hypertext markup language) and does not require complex scripting or programming. For more details refer section 8.2.4.
5. The Browser Object Model is the one part of JavaScript for which no formal standards exist. It is also the part that is generally the least useful and the part which newbies most want to work with. For more details Refer section 8.3.
6. Delegation event model is an event is created by a source and sent to one or more listeners. For more details Refer section 8.2.3.

8.7 References Books

- Teague, J.C. (2001). *DHTML and CSS: for the World Wide Web*. Peachpit press.

E-References:

- <http://msdn.microsoft.com/en-us/library/ms531386%28v=VS.>

- 85%29.aspx
- <http://www.sitepoint.com/control-internet-explorer/>
- http://javascript.about.com/od/browserobjectmodel/Browser_Object_Model.htm

