



# DHTML

## 9.1 DHTML – AN INTRODUCTION

DHTML stands for *Dynamic Hyper Text Markup Language*. DHTML is not a language or a web standard, rather is a term describing the art of making dynamic and interactive web pages. In DHTML, the term “Dynamic” means the ability of the browser to alter a web document's look and style after the document has loaded. DHTML give you more control over the HTML elements and allows them to change at any time, without returning to the web server. DHTML is basically a combination of HTML, JavaScript, and CSS that allows web documents to be more dynamic and interactive. DHTML is a new tool for web designers to create web pages with special effects and animations. *For example*, a piece of text can change from one size or color to another, or a graphic can move from one location to another, in response to some kind of user action, such as clicking a button.

DHTML makes the web experience faster and more interactive for end users. Live updates can be activated by user and browser-driven events. For example, text and images can be hidden in a document until a user moves the mouse over an element or clicks a button. A browser-driven event can be triggered by internal actions, such as when the browser finishes downloading the web page.

### 9.1.1 Components of DHTML

DHTML requires following independent components to work:



- (i) HTML (Hyper Text Markup Language)
- (ii) Cascading Style Sheets (CSS)
- (iii) Scripting

This section provides a brief description of each component.

- (i) **HTML:** HTML stands for Hypertext Markup Language, and it is the most widely used language to design Web Pages. HTML defines the structure of a web page, using some basic elements or tags.
- (ii) **Cascading Style Sheets (CSS):** CSS controls the formatting of HTML elements. It defines **how to display** HTML elements over a webpage. CSS is easy to learn and understand but it provides powerful control over the presentation of an HTML document.
- (iii) **Scripting:** Scripting provides the mechanisms to interpret user actions and produce client-side changes to a page. For example, scripts can interpret mouse actions (such as the mouse passing over a specified area of a page through the event model) and respond to the action by using a set of predefined instructions (such as highlighting the text activated by the mouse action). Although DHTML can communicate with several scripting languages, JavaScript is the de facto standard for creating cross-browser DHTML pages.

## 9.2 STATIC VS. DYNAMIC WEB PAGE

STATIC WEB PAGE	DYNAMIC WEB PAGE
Static Web Page is one that does not change its appearance, layout or content once it has been loaded.	Dynamic Web Page is one that can change its appearance, layout or content once it has been loaded.
Browsing and loading of static web pages are faster than dynamic web pages.	Browsing and loading of dynamic web pages are slower than static web pages due to the intervention of web server.
Static web pages are created through HTML language.	Dynamic web pages are created by the usage of <i>PHP, JavaScript</i> and <i>Actionscript</i> languages.
No Event Handling.	Events can be handled.



No Database connectivity.	Database can be connected at backend with frontend web pages.
---------------------------	---

### 9.3 WEB DESIGNING VS. WEB DEVELOPMENT

WEB DESIGNING	WEB DEVELOPMENT
The process of designing static web pages is called Web Designing.	The process of developing dynamic web pages is called Web Development.
A markup language like HTML can be used to design a static webpage.	DHTML can be used to develop a dynamic webpage.
No Programming.	Programming statements can be embedded to build a dynamic webpage.

### 9.4 HTML VS. DHTML

HTML	DHTML
HTML stands for Hyper Text Markup Language mainly used to design static or flat web pages.	DHTML stands for Dynamic Hyper Text Markup Language mainly used to develop dynamic or interactive web pages.
HTML creates a plain web page without any Styles and Scripts.	DHTML creates a dynamic web page with HTML, CSS, DOM and Scripts.
HTML cannot have any server side code.	DHTML can contain server side code.
In HTML, there is no need for database connectivity.	DHTML may require connecting to a database as it interacts with user.
HTML sites work slowly upon client-side technologies.	DHTML sites work faster upon client-side technologies.
HTML does not make use of any methods for making it dynamic.	DHTML uses events, methods and much more for providing dynamism for HTML pages.

### 9.5 DOCUMENT OBJECT MODEL (DOM)

Every web page resides inside a browser window which can be considered as an *object*. A *Document object* represents the HTML document that is displayed in that window. The



Document object has various properties that refer to other objects which allow access to and modification of document content.

The way, a document content is accessed and modified is called the *Document Object Model*, or *DOM*.

DOM is a W3C (*World Wide Web Consortium*) standard used for accessing web documents. The objects are organized in a hierarchy. This hierarchical structure applies to the organization of objects in a Web document.

- **Window Object:** Top of the hierarchy is the WINDOW object. It is the outmost element of the object hierarchy.
- **Document Object:** Each HTML document that gets loaded into a window becomes a DOCUMENT object. The document contains the contents of the page.
- **Form Object:** Every-thing enclosed in the <FORM>...</FORM> tags set the FORM object.
- **Form Control Elements:** The form object contains all the elements defined for that object such as *text fields, buttons, radio buttons, and checkboxes*.

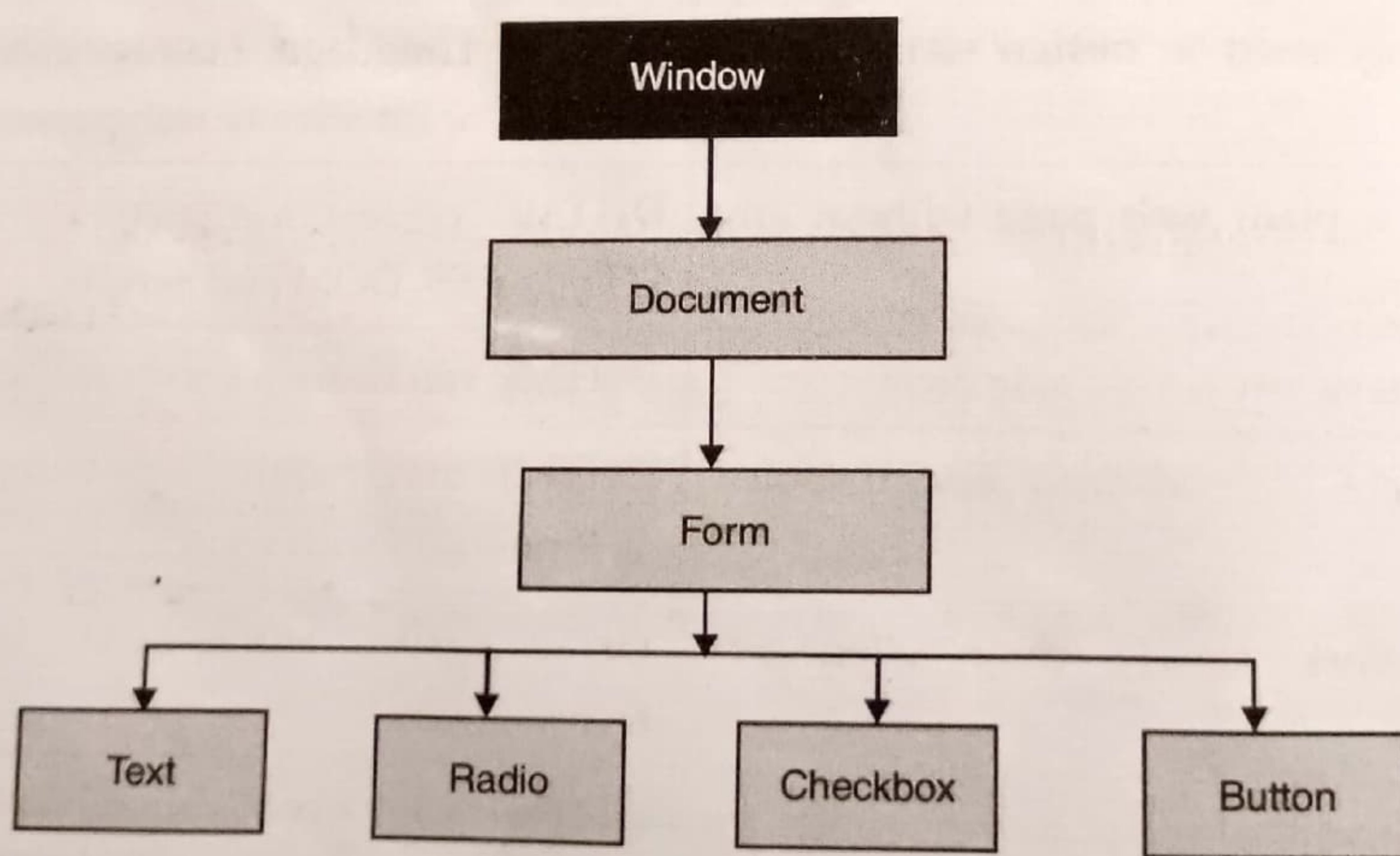


Fig. 9.1 (Hierarchy of Objects in a Web document)

The HTML DOM is a standard object model and programming interface for HTML. It defines:

- The HTML elements as *objects*.
- The *properties* of all HTML elements.



- The *methods* to access all HTML elements.
- The *events* for all HTML elements.

In other words, the HTML DOM is a standard for how to get, change, add, or delete HTML elements.

## 9.6 HTML AND SCRIPTING ACCESS

Even though the DOM specifies a model for all the objects and HTML elements that make a Web document, they need to be named properly to allow scripting languages to easily read and manipulate them. The basic way to attach a unique identifier to an HTML element is by using the *ID attribute*. The ID attribute is associated with nearly every element. The point of the ID attribute is to bind a unique identifier to the element.

*Example:*

```
<P ID="PR">Paragraph Starts</P>
```

*Naming* is very important. Authors are encouraged to adopt a consistent naming style and to avoid using potentially confusing names that include the names of HTML elements themselves. For example, `button` does not make a very good name because it is the name of the HTML element itself.

The NAME attribute is often used with few HTML elements, like- the NAME attribute is commonly defined for `<A>`, `<APPLET>`, `<BUTTON>`, `<EMBED>`, `<FORM>`, `<FRAME>`, `<IFRAME>`, `<IMG>`, `<INPUT>`, `<OBJECT>`, `<MAP>`, `<SELECT>`, and `<TEXTAREA>`.

Earlier browsers will not recognize the ID attribute, so use NAME as well. However, as NAME and ID share the same namespace, you must not use the same value for both.

*Example:*

```
<IMG NAME="Pic1.jpg">
```

```
<FORM NAME="F1">
```

When HTML documents are well formed, scripting languages such as JavaScript and VBScript can be used to read and manipulate the various objects in a page. DOM defines a special set of reserved names that use this notation to allow scripting languages like JavaScript to refer to entities in the browser and the document, including form elements.



The basic notation uses a series of object and property names separated by dots. To access the form defined by

```
<FORM NAME="myform">  
  <INPUT TYPE="TEXT" NAME="username">  
</FORM>
```

with a scripting language (like- JavaScript), use either *window.document.myform* or simply *document.myform*. The field and its value can be accessed in a similar fashion. To access the text field, use *document.myform.username*. To access the actual value of the username field, access the *value* property using *document.myform.username.value*.

## 9.7 ROLLOVER BUTTONS

A rollover button is a dynamic web button that changes appearance depending on the location of the user's mouse pointer. It contains three states: *normal*, *over* and *down*.

- The *normal* state appears when your mouse is off the button. (*Unavailable state*)
- The *over* state applies when your mouse rolls over the button. (*Active state*)
- The *down* state applies to when you click on the button. (*Inactive state*)

To create a rollover button, you will first need at least two, perhaps even three images, to represent each of the button's states- *inactive*, *active*, and *unavailable*. A sample set of rollover images is shown in Fig. 9.2.



Fig. 9.2 (Rollover Images)

*Rollover buttons* come in different shapes, sizes, colors and styles. Since images can be converted to rollover buttons, a rollover button can also contain images.

Rollover buttons are used primarily as *navigational buttons* on a web page to direct people to other locations. They are also used in drop-down and pop-up menus. Other rollover buttons are used for animated effects and sounds, so an image, color, shape, text or sound can change as the user rolls over the button on the Web.

Rollover buttons extend the normal functionality of a button. Web designers can add



sound effects that will play if a user moves his mouse over these buttons. When used in drop-down or pop-up menus, rollover buttons help save navigational space.

A rollover button can be created with image-editing software, such as *Adobe Photoshop*, *Image Ready*, *Flash* and *Fireworks*. Web-editing software programs like *Dreamweaver* and *Microsoft Visual Studio Developer* can also be used to create rollover button effects.

## 9.8 MOVING OBJECTS WITH DHTML

Moving an object from one fixed point to another is great, but you have to know exactly where you want to move the object.

Moving the object is not a difficult process. The basic idea is to change the *left* and *top* properties of the <DIV> element by a set number of pixels.

- **Move Left** - Use a negative value for *left*.
- **Move Right** - Use a positive value for *left*.
- **Move Up** - Use a negative value for *top*.
- **Move Down** - Use a positive value for *top*.

*For Example:*

```
<DIV ID="dhtml" STYLE="position: absolute; width: 200px;  
height:115px; z-index:1; left: 100px; top: 50px">  
<IMG SRC="pk.gif" WIDTH="275" HEIGHT="175">  
</DIV>
```

In the above example,

- To move the image ("pk.gif") in the right direction, set a positive (increased) value for *left*.
- To move the image ("pk.gif") in the left direction, set a negative (decreased) value for *left*.
- To move the image ("pk.gif") in the downward direction, set a positive (increased) value for *top*.
- To move the image ("pk.gif") in the upward direction, set a negative (decreased) value for *top*.



## 9.9 RAMIFICATIONS OF DHTML

DHTML is not a language. DHTML is a combination of Web development technologies used to create dynamically changing websites. Web pages may include animation, dynamic menus and text effects. The technologies used include a combination of HTML, JavaScript or VBScript, CSS and the Document Object Model (DOM).

To use DHTML, you need to make sure your pages are coded properly in HTML.

Second, you need to understand Style Sheets and JavaScript very well.

Finally, you need to understand the Document Object Model.

If you consider all of the things you need to know to create a DHTML page, you realize that creating a DHTML page is much more complicated than creating a static HTML page.

The bottom line is that DHTML has changed the process of web development drastically.