Unit 3
1.6 Dynamic HTML (DHTML)

1.6 Dynamic HTML (DHTML)

DHTML is NOT a language. DHTML is the art of combining HTML, JavaScript, DOM, and CSS. According to the World Wide Web Consortium (W3C) - "Dynamic HTML is a term used by some vendors to describe the combination of HTML, style sheets and scripts that allows documents to be animated."

1.6.1 How to make a DHTML document

You can use JavaScript to embed DOM to make your script DHTML. In JavaScript, the statement: document.write(), is used to write output to a web page.

The example 1.63 uses JavaScript to display the current date and time on a page.

```
<html>
<body>

<script type="text/javascript">
document.write(Date());
</script>

</body>
</html>
```

Sat Jul 20 11:05:51 CDT 2014

Figure 1.74

Example 1.63

The example 1.64 changes the content of an h1 element.

```
<html>
<body>

<h1 id="header">Old Header</h1>

<script type="text/javascript">
document.getElementById("header").innerHTML="New Header";
</script>

<p>"Old Header" was changed to "New Header"</p>

</body>
</html>
```

New Header

"Old Header" was changed to "New Header"

Figure 1.75

Example 1.64

In example 1.64, the HTML document contains an **h1** element with **id="header"** We use the HTML DOM to get the element with id="header" . A JavaScript changes the content (**innerHTML**) of that element.

In the example below (Example1.65), initially, the HTML document contains an **img** element with id="image" . Then HTML DOM is used to get the element with id="image", followed by A JavaScript to change the **src** attribute of that element from **"smiley.gif"** to **"landscape.jpg".**

```
<html>
<body>

<img id="image" src="smiley.gif" width="160" height="120">

<script type="text/javascript">
document.getElementById("image").src="landscape.jpg";
</script>

<p>The original image was smiley.gif, but the script changed it to landscape.jpg</p>

</body>
</html>
```

Example 1.65

The original image was smiley.gif, but the script changed it to landscape.jpg

Figure 1.76

1.6.2 HTML Events

Every element on an HTML page has events which can trigger a JavaScript.

For example, we can use the **onClick** event of a button element to indicate that a function will run when a user clicks on the button. We define the events in the HTML tags. Some **Examples of events** are given below:

• A mouse click
• A web page or an image loading
• Hovering the mouse or "Mousing" over a hot spot on the web page
• Selecting an input field in an HTML form
• Submitting an HTML form
• A keystroke

In the Example 1.66, the content of the h1 element will change when a user clicks on it.

```
<html>
<body>

<h1 onclick="this.innerHTML='Ooops!'">Click on this text</h1>

</body>         <!--The 'this' pointer is by default there. The -->
</html>         <!-- statement  will work even without the 'this -->
```

Example 1.66

Click on this text

Figure 1.77

Ooops!

Figure 1.78

You can also add the script in the head section, and then call a function from the event handler.

```
<html>
<head>
<script type="text/javascript">
function changetext(id)
{
id.innerHTML="Ooops!";
}
</script>
</head>
<body>

<h1 onclick="changetext(this)">Click on this text</h1>

</body>
</html>
```

Example 1.67

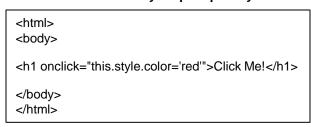**Click on this text**

Figure 1.79

Ooops!

Figure 1.80

1.6.3 DHTML CSS

JavaScript and the HTML DOM can be used to change the style of any HTML element.

1.6.3.1 Change Style of the Current HTML Element

To change the style of the current HTML element, use the following statement:

this.style.property= new style

```
<html>
<body>

<h1 onclick="this.style.color='red'">Click Me!</h1>

</body>
</html>
```

Example 1.68

Click Me!   **Click Me!**

Figure 1.81   Figure 1.82

1.6.3.2 Change Style of a Specific HTML Element

To change the style of a specific HTML element, use the following statement:

document.getElementById(id).style.property=new style

```
<html>
<body>

<h1 id="h1"
onclick="document.getElementById('h1').style.color='red'">Click Me!</h1>

</body>
</html>
```

Example 1.69

Click Me!

Figure 1.83

Click Me!

Figure 1.84