

2

HTML and JavaScript Programming

<i>If you need information on:</i>	<i>See page:</i>
HTML	42
Creating Headings on a Web Page	48
Creating a Paragraph	54
Working with Tables	61
Introduction to Forms and HTML Controls	87
Introducing DHTML	106

HTML, also known as Hypertext Markup Language, is the most commonly used markup language to create Web pages. A markup language provides a way to describe the structure of text-based information on a Web page. Hypertext though similar to regular text, has one additional advantage that is when you click the hypertext present on a Web page, you are directed to another Web page on the Internet. The hypertext is called hyper because the navigation through the pages using the hypertext is not linear. It means that if you click the hypertext present on a Web page, you are directed to the relevant page on the website or Internet, which is not necessarily the next page on the website. World Wide Web Consortium (W3C) is an organization that defines new specifications for HTML and is responsible to update the language.

HTML allows you to format, arrange, and group text, display text as links, and add images and multimedia to a Web page. It also allows you to create and work with style sheets, controls, and embed scripting language code (such as JavaScript code) in a Web page.

JavaScript is an interpreted, client-side (the scripts run in the Web browser), and object-based scripting language that offers various functionalities to enliven the static HTML Web pages. The scripts written in JavaScript are processed line by line, which is the reason to refer JavaScript as an interpreted language. The scripts are interpreted by the JavaScript interpreter that is an in-built component of the Web browser. While working with the JavaScript scripts, you can use the in-built objects and other programmable features, such as control flow statements and events to make your HTML Web pages dynamic.

We begin this chapter by getting familiar with basic concepts of HTML. Then, we discuss how to use the HTML form and form controls in a Web page, and how to use cascading style sheets in an HTML document. In the later sections, we discuss the process to create variables and objects, handle events, and use operators, functions and control and looping statements in JavaScript.

Let's first know about some basic concepts of HTML.

HTML

HTML is written in the form of tags, which are by pairs of angle brackets (< and >) and some text placed between these brackets. The text present between a pair of angle brackets defines an HTML element. Most of the HTML elements have two basic properties—attributes and contents; therefore, they have an opening tag (<element-name>) and a closing tag (</element-name>). However, some HTML elements, such as
 and <hr> do not have any content; therefore, they do not need a closing tag. HTML tags, elements, and attributes are collectively known as HTML markup.

The attributes of an HTML element are placed within the opening tag (after the element's name) and its content is placed between the opening and closing tags. Most of the HTML attributes are name-value pairs, separated by the = sign. However, some attributes, such as ismap attribute of the element is written without specifying a value for it. Attribute values should be enclosed within either single or double quotes. The syntax of an HTML element is given as follows:

```
<element-name attribute-name="attribute-value">content </element-name>
```

In this section, we learned about the HTML markup. Now, let's know how an HTML document is structured.

Introducing HTML Document Structure

The basic structure of an HTML document is given in Listing 2.1:

Listing 2.1: Basic Structure of an HTML Document

```
<!DOCTYPE>
<html>
<head>
<title>
Title of the web Page
</title>
</head>
<body>
Contents of the web Page
</body>
</html>
```

You can notice in Listing 2.1 that an HTML document formally begins with the `<!DOCTYPE>` element. However, all your HTML code resides between the `<html>` and `</html>` tags. The `<html>` element has two subelements inside it: `<head>` and `<body>`. The `<head>` element is used to specify information about the HTML Web page, such as title of the Web page, whereas actual contents of the Web page are specified inside the `<body>` element.

Now, we describe all the elements present in the preceding syntax, one by one.

The `<!DOCTYPE>` Element

The `<!DOCTYPE>` element is the first element in the HTML document, which specifies the Document Type Definition (DTD) used by the document. A DTD is a separate file containing formal definition of grammar, such as supported elements and attributes used in markup language. The browser checks the code of the document against the rules in the `<!DOCTYPE>` declaration. The `<!DOCTYPE>` element does not have a closing tag.

HTML 4.01 specifies following three types of DTDs:

- **HTML 4.01 Strict DTD**—Includes all elements and attributes that have not been deprecated (those are elements and attributes that W3C can remove in the future HTML version) or do not appear in the frameset documents. You should use this DTD when you do not want the presentation elements and attributes to appear with markup. In addition, it should be used together with Cascading Style Sheets (CSS). The strict DTD uses the following `<!DOCTYPE>` declaration:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

- **HTML 4.01 Transitional DTD**—Includes deprecated elements and attributes in addition to the elements and attributes that are included in the strict DTD. It also includes the presentation elements and attributes that are included in a style sheet. You should use this DTD when you want to use the presentation features of HTML. DTD is useful because your readers may have browsers that do not support CSS. The transitional DTD use the following `<!DOCTYPE>` declaration:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

- **HTML 4.01 Frameset DTD**—Includes frames in addition to the elements and attributes that are present in the transitional DTD. The only difference between the transitional DTD and the frameset DTD is that the `<body>` element in the transitional DTD is replaced by the `<frameset>` element in the frameset DTD. This DTD should be used for documents with frames. The frameset DTD use the following `<!DOCTYPE>` declaration:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
```

The `<html>` Element

The `<html>` element starts an HTML document and it comes after the `<!DOCTYPE>` element in an HTML document. It contains everything an HTML document contains except the `<!DOCTYPE>` element. Table 2.1 lists all the attributes of the `<html>` element:

<code>class</code>	Represents the class of the element and is used to render the content.
<code>dir</code>	Gives the direction to directionally neutral text; you can set this attribute to <code>ltr</code> for left to right text direction or <code>rtl</code> for right to left text direction.
<code>id</code>	Represents unique alphanumeric identifier for the element.
<code>lang</code>	Represents base language used for the element.
<code>version</code>	Represents the version of the language used. This attribute has been marked as deprecated.
<code>xmlns</code>	Declares a namespace for custom tags in an HTML document.

The <head> Element

The <head> element contains general information about the HTML document, such as its title, keywords for search engines, and a base address for URLs. Table 2.2 lists all the attributes of the <head> element:

Table 2.2: Attributes of the <head> Element	
class	Represents the class of the element and is used to render the content.
dir	Gives the direction to directionally neutral text; you can set this attribute to ltr for left to right text direction or rtl for right to left text direction.
id	Represents unique alphanumeric identifier for the element.
lang	Represents base language used for the element.
profile	Gives the location of one or more white-space separated metadata profile URLs for the current document.
style	Represents inline style indicating how to render the element.
title	Holds additional information for the element.

Table 2.3 lists all the elements that can be added inside the <head> element:

Table 2.3: Elements to be Added to the <head> Element	
<base>	Represents base URL for the document.
<basefont>	Represents base font for the document.
<bgsound>	Represents background sound.
<isindex>	Represents rudimentary input control.
<link>	Represents a relationship between the document and another object.
<meta>	Represents header information.
<nextid>	Represents hint for the name value to use when creating a new hyperlink element.
<noscript>	Holds text that only appears if the browser does not support the <script> element.
<script>	Holds programming script statements, such as JavaScript.
<style>	Includes style information to render the content.
<title>	Represents title of the Web page that appears in the Web browser.

The <title> Element

The <title> element contains the title of the HTML document, which appears in the title bar of the Web browser and is used by search engines to refer the document. Each <head> element should contain a <title> element. You should try to keep the title text relatively short and to the point because some browsers face difficulties in handling titles longer than 256 characters. Table 2.4 lists all the attributes of the <title> element:

Table 2.4: Attributes of the <title> Element	
class	Represents the class of the element and is used to render the content.
id	Represents unique alphanumeric identifier for the element.
lang	Represents base language used for the element.

Table 2.4: Attributes of the <title> Element

Table 2.4: Attributes of the <title> Element	
style	Represents inline style indicating how to render the element.

The <body> Element

The <body> element contains the body of the HTML document, which includes the entire content that will appear in the Web browser. It can also include text, images, and multimedia elements. Table 2.5 lists all the attributes of the <body> element:

Table 2.5: Attributes of the <body> Element

Table 2.5: Attributes of the <body> Element	
alink	Specifies the color of the hyperlinks when they are clicked; you can set this attribute to a predefined color name or value. This attribute has been marked as deprecated.
background	Represents the URL of a graphic file, which is used as the background of the browser. This attribute has been marked as deprecated.
bgcolor	Specifies the color of the background of the browser. You can set this attribute to a predefined color name or a color value. This attribute has been marked as deprecated.
bgproperties	Indicates if the background should scroll when the text scrolls. If you set this attribute to fixed, which is the only allowed value, the background will not scroll when the text scrolls.
bottommargin	Specifies the bottom margin, which is the empty space at the bottom of the document, in pixels.
class	Represents the class of the element and is used to render the content.
dir	Gives the direction to directionally neutral text. You can set this attribute to ltr for left to right text direction or rtl for right to left text direction.
id	Represents unique alphanumeric identifier for the element.
lang	Represents base language used for the element.
language	Represents scripting language used for the element.
leftmargin	Specifies the left margin, which is the empty space to the left of the document, in pixels.
link	Specifies the color of the hyperlinks that have not yet been visited. You can set this attribute to a predefined color name or value. This attribute has been marked as deprecated.
marginheight	Gives the height of the top and bottom margins, in pixels.
marginwidth	Gives the width of the left and right margins, in pixels.
rightmargin	Specifies the right margin, which is the empty space to the right of the document, in pixels.
scroll	Specifies whether a vertical scrollbar appears to the right of the document.
style	Specifies whether a vertical scrollbar appears to the right of the document; you can set this attribute to yes (which is the default value for the attribute) or no.
text	Specifies the color of the text in the document. You can set this attribute to a predefined color name or value. This attribute has been marked as deprecated.
title	Holds additional information for the element, such as tooltips.
topmargin	Specifies the top margin, which is the empty space at the top of the document, in pixels.

Table 2.5: Attributes of the <body> Element	
vlink	Specifies the color of the hyperlinks that have been visited. You can set this attribute to a predefined color name or value. This attribute has been marked as deprecated.

After getting familiar with the basic structure of an HTML document, let's now create and save an HTML document.

You can create an HTML document by forming a new, blank document in a text editor, such as Notepad, adding your HTML code to the document, and saving it with .html extension. We are using Notepad as our text editor to create HTML documents throughout the chapter.

Let's perform the following steps to create and save an HTML document:

1. Click Start→All Programs→Accessories→Notepad, as shown in Figure 2.1:

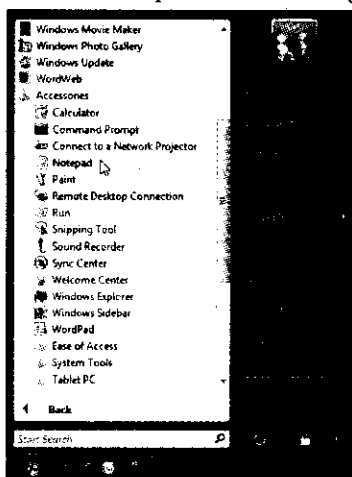


Figure 2.1: Opening Notepad

The Notepad window opens with a new, blank document, as shown in Figure 2.2:

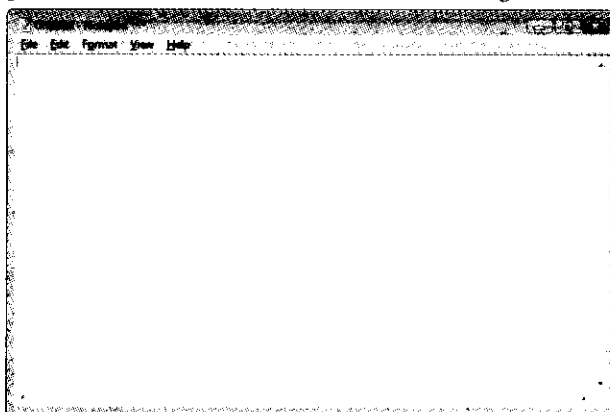


Figure 2.2: The Notepad Window

2. Add code to the document to display some information on the Web page. In this case, we add the code given in Listing 2.2 to the document:

Listing 2.2: Creating a Simple HTML Document

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>
Title of the Web Page
</title>
</head>
<body>
Contents of the Web Page
</body>
</html>

```

3. Click File→Save As, as shown in Figure 2.3:

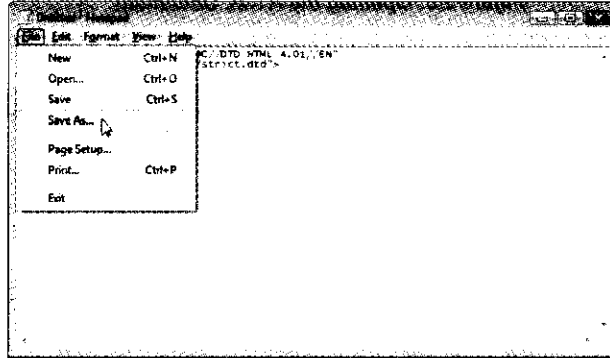


Figure 2.3: Saving the Code

The Save As dialog box appears, as shown in Figure 2.4:

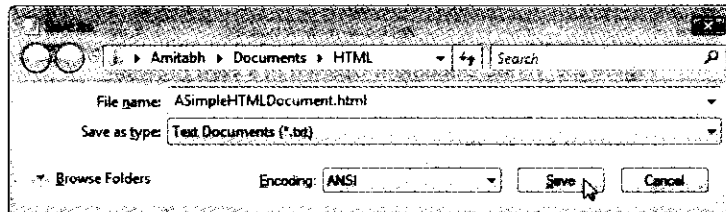


Figure 2.4: The Save As Dialog Box

In the Save As dialog box, you need to specify a location, name, and file type for your document.

4. Select the location where you want to save your document, from the address bar. In this case, we have selected HTML, as shown in Figure 2.4.
5. Type a name for the document with the .html extension, in the File name combo box. In this case, we have typed ASimpleHTMLDocument.html, as shown in Figure 2.4.
6. Click the Save button, as shown in Figure 2.4. The Save As dialog box is closed and your document is saved as an HTML document at the specified location.

You can find the ASimpleHTMLDocument.html file in the Code\HTML\Chapter 2 folder on the CD. After creating an HTML document, you can open it in a Web browser to view the generated HTML Web page.

Let's perform the following steps to open an HTML document in a Web browser:

1. Open the folder containing your HTML document, as shown in Figure 2.5:

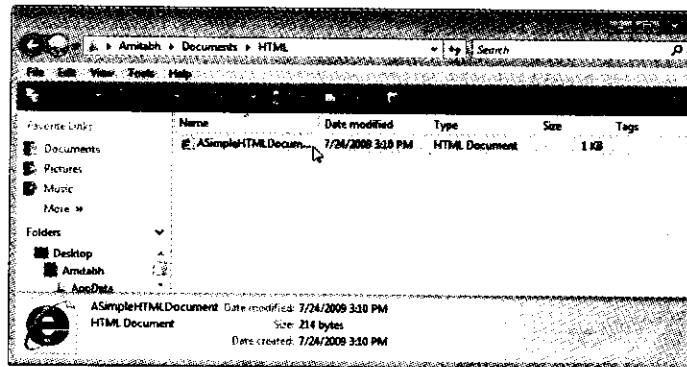


Figure 2.5: Opening the HTML Document

2. Double-click the HTML document, as shown in Figure 2.5. The HTML document opens in the default Web browser set in your computer, as shown in Figure 2.6:

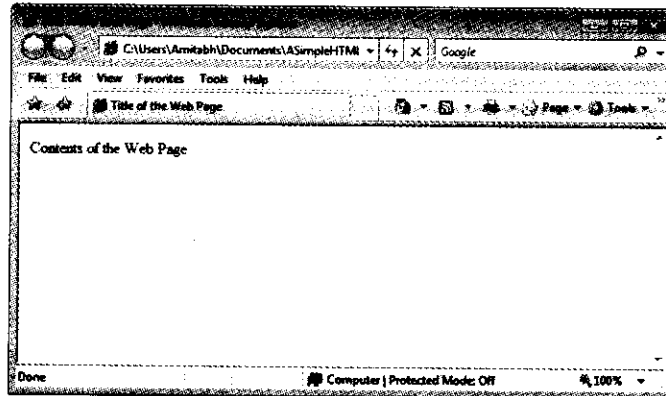


Figure 2.6: HTML Document Opened in a Web Browser

Let's learn about the creation of headings on a Web page in the next section.

Creating Headings on a Web Page

Headings help to define the format and structure of a document and highlight important topics. HTML has special, pre-formatted tags to create headings on a Web page. The headings in HTML appears bold by default and their size depends on their level. HTML has the following six heading tags: `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, and `<h6>`. The heading tags are in pairs, with an opening tag and a closing tag. Any text inside these tags is displayed differently depending on the heading number. They get their own line on the Web page by starting in a new line. Let's create a Web page, named `Headings.html` to understand the working of the heading tags. You can find the `Headings.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.3 shows the code of the `Headings.html` page:

Listing 2.3: Creating Headings on a Web Page

```

<html>
<head>
<title>
Web Page Heading
</title>
</head>
<body>
<h1>This is an example of H1 heading.</h1>
<br>

```



```

<h2>This is an example of H2 heading.</h2>
<br>
<h3>This is an example of H3 heading.</h3>
<br>
<h4>This is an example of H4 heading.</h4>
<br>
<h5>This is an example of H5 heading.</h5>
<br>
<h6>This is an example of H6 heading.</h6>
<br>
</body>
</html>

```

The output of Listing 2.3 appears, as shown in Figure 2.7:

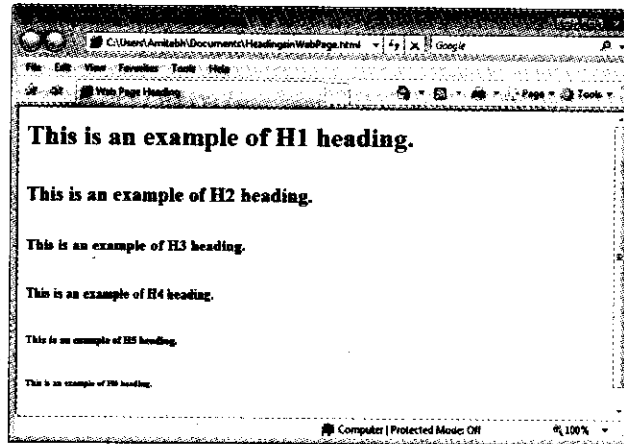


Figure 2.7: The Heading Tag

In Figure 2.7, you can observe how the different levels of headings appear on a Web page with their different relative sizes.

NOTE

The heading tags `<h1>` to `<h6>` have a common `<align>` attribute, which you can use to align your text on a Web page.

Let's learn the alignment of headings in the following section.

Aligning the Headings

In HTML you can align your text and display it according to your requirement on a Web page. By default, the text you enter in your Notepad appears left-aligned on the Web page. You can use the `align` attribute to change the alignment of your text. The values of the `align` attribute are `center`, `left`, `right`, and `justify`. Table 2.6 briefly describes the values of the `align` attribute:

center	Aligns the whole text to the center of the Web page.
left	Aligns the whole text to the left of the Web page. This is the default alignment.
right	Aligns the whole text to the right of the Web page.
justify	Justifies the whole text and also indents the first line.

Listing 2.4 shows an example of how the `align` attribute works:

Listing 2.4: Aligning Text

```
<html>
<head>
<title>
Aligning Text
</title>
</head>
<body>
<h2 align="center">This text is aligned in the center.
<br>
<h2 align="left">This text is aligned in the left.
<br>
<h2 align="right">This text is aligned in the right.
</body>
</html>
```

You can find this example as `Headings.html` file in the `Code\HTML\Chapter 2` folder on the CD. The output of Listing 2.4 appears, as shown in Figure 2.8:

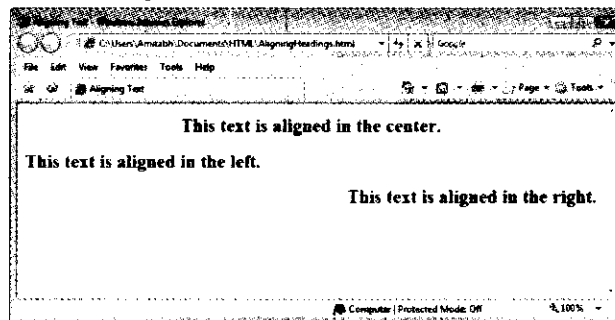


Figure 2.8: Aligning the Heading

In Figure 2.8, you can observe that the first line is aligned to the center, the second line is aligned to the left, and the third line is aligned to the right.

Working with Links

Links (or hyperlinks) are used to connect one Web page to another. When you click a link, you are directed (sent) to the Web page specified as the destination Web page for the link. You can use HTML to create links and therefore provide a means of connection between Web pages or for different sections of the same Web page.

In this section, you learn how to create a hyperlink, set the hyperlink color, and linking different sections of the Web page. Let's first learn about creating a hyperlink.

Creating a Hyperlink

A hyperlink is a link between Web pages and when you click it, you move to another Web page or some other section of the same Web page. You can create a hyperlink by using the anchor tag (`<a>`). Inside the `<a>` tag, give the reference of a Web page that you want to open. The `href` attribute of the `<a>` tag takes the reference of the Web page. The term `href` stands for Hypertext Reference. Let's create two Web pages, named `Hyperlink.html` and `Page1.html` to understand the working of the anchor tag. You can find the `Hyperlink.html` and `Page1.html` files in the `Code\HTML\Chapter 2` folder on the CD. In the first Web page, we create a hyperlink by using the code given in Listing 2.5:

Listing 2.5: Creating a Hyperlink

```
<html>
<head>
<title>Link</title>
```

```

</head>
<body>
<h2>To View Page 1 click the Hyperlink</h2>
<a href="Page1.html" target="_blank" >
<h1>Page 1</h1>
</a>
</body>
</html>

```

Now, create another Web page used as a reference page for the hyperlink, that is the Web page to which this hyperlink will take you after clicking. Listing 2.6 shows the code of the reference Web page:

Listing 2.6: Creating a Reference Page for the Hyperlink

```

<html>
<head>
<title>Page
</title>
</head>
<body bgcolor='aliceblue'>
<h1 align=center>Page 1</h1>
<h4 align=center>This is Page 1</h4>
</body>
</html>

```

When you open the Web page in which the reference link is added, the page appears, as shown in Figure 2.9:

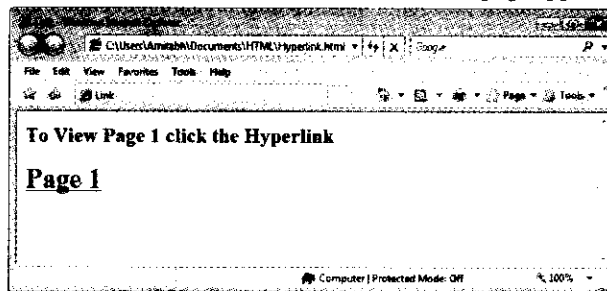


Figure 2.9: Hyperlink in a Web Page

Click the Page 1 link on the Web page. When you click the Page 1 link, you are directed to the Page 1 Web page, as shown in Figure 2.10:

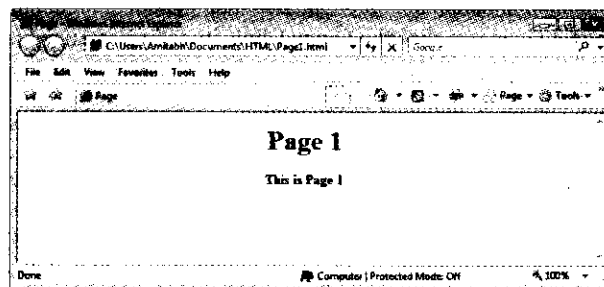


Figure 2.10: Referenced Page of the Hyperlink

Setting the Hyperlink Colors

Whenever you open a Web page for the first time, the hyperlinks appears in blue color. However, when you click a hyperlink, its color changes from blue to purple. If you want, you can also set the hyperlink color according to your requirements. To do this, HTML provides three attributes (`link`, `vlink`, and `alink`), which you can use inside the `<body>` tag. These links are briefly explained as follows:

- ❑ link attribute specifies the color of the hyperlink that has not been visited (clicked) before on a Web page.
- ❑ vlink attribute specifies the color of the hyperlink that has been visited before on a Web page.
- ❑ alink specifies the color of currently active hyperlink.

Let's create a Web page named `LinkColors.html` to learn how to set the hyperlink colors. You can find the `LinkColors.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.7 shows the code of the `LinkColors.html` page:

Listing 2.7: Setting the Hyperlink Colors

```
<html>
<head>
<title>Links</title>
</head>
<body link="green" vlink="brown" alink="blue">
<h1>Setting Hyperlink Colors</h1>
<a href="Page1.html" target="_blank">
<h2>Page 1</h2>
<a href="Page2.html" target="_blank">
<h2>Page 2</h2>
</a>
</body>
</html>
```

The output of Listing 2.7 after visiting the Page 1 link appears, as shown in Figure 2.11:



Figure 2.11: Setting Hyperlink Colors

As you can observe in Figure 2.11, as you click the Page 1 link, its color is changed from green to blue.

Linking Different Sections of a Web Page

If you want, you can move to different locations of a Web page either in the same document or in different documents through links. Links to a specific location within the same document are created for Web pages containing a large amount of text. To do this, you need to first create a named anchor tag (`<a>`). Inside the anchor tag, specify the location where you want to move on the Web page. Then, create another anchor tag (`<a>`) and use the `href` attribute of the `<a>` tag, give the reference of the named anchor tag.

Let's create a Web page, named `LinkingSections.html` to learn how to link different sections of a Web page. You can find the `LinkingSections.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.8 shows the code of the `LinkingSections.html` page:

Listing 2.8: Linking Different Sections of a Web Page

```
<html>
<head>
<title>Links</title>
</head>
<body>
```

```

<center>
<h1> Linking to a Section in a Page </h1>
<h4>Click here to go at the
<a href="#BOTTOM"> bottom </a>
of the page.</h4>
<br/><br/><br/><br/><br/><br/><br/><br/>
<br/><br/><br/><br/><br/><br/><br/><br/>
<br/><br/><br/><br/><br/><br/><br/><br/>
<br/><br/><br/><br/><br/><br/><br/><br/>
<br/><br/><br/><br/><br/><br/><br/><br/>
<hr>
<a name="BOTTOM"><h4>This is the bottom of the page.</h4></a>
</center>
</body>
</html>

```

The output of Listing 2.8 appears, as shown in Figure 2.12:

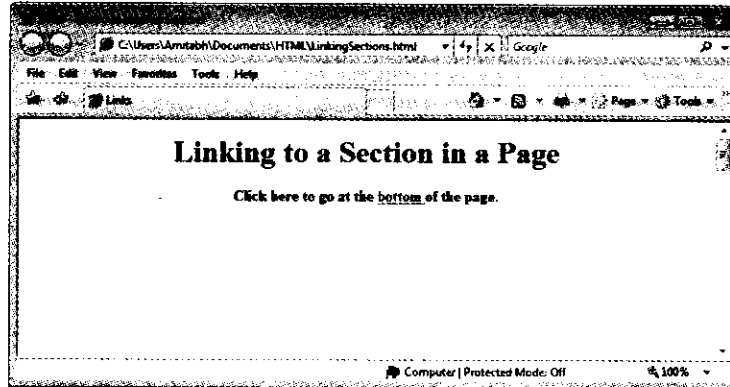


Figure 2.12: Hyperlink to a Section in a Page

Now, click the bottom link on the Web page. As you click the link, the page section moves to the bottom of the page, as shown in Figure 2.13:

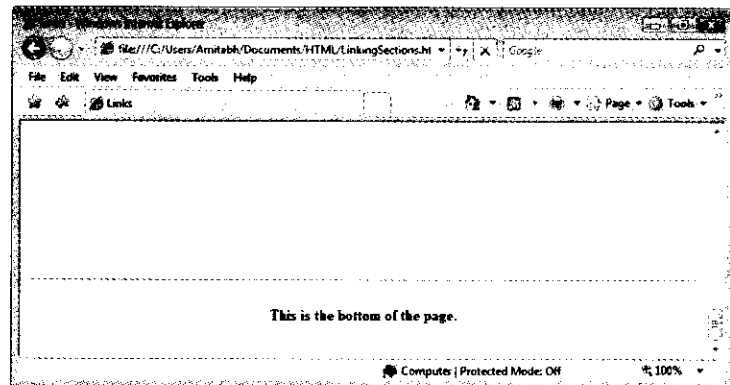


Figure 2.13: Referenced Section of the Hyperlink

NOTE

The reference of the named anchor tag inside the href attribute starts with the # symbol. It is also important to note that in place of the name attribute of the <a> tag, you can also use the id attribute.

Let's learn about the process to create a paragraph in next section.

Creating a Paragraph

When you write texts within the heading tags or within the `<body>` tag, HTML only recognizes single spaces between characters. The `<p>` tag marks a block of text as a paragraph. The tag `<p>` marks the beginning of a paragraph, and the `</p>` tag marks its end. The Web browser formats the text into a paragraph to fit the current page width. Breaking your document into paragraphs provides an easy way of formatting your text.

Let's create a Web page, named `CreatingParagraphs.html` to learn how the `<p>` tag affects its text. You can find the `CreatingParagraphs.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.9 shows the code of the `CreatingParagraphs.html` page:

Listing 2.9: Creating a Paragraph

```
<html>
<head>
<title>
Creating Paragraphs
</title>
</head>
<body>
<p>
This is an example of creating paragraphs on the web.
</p>
<p>
The number of lines in a paragraph depends on the size of your browser window.
If you resize the browser window, the number of lines in this paragraph will
change.
</p>
</body>
</html>
```

The output of Listing 2.9 appears, as shown in Figure 2.14:

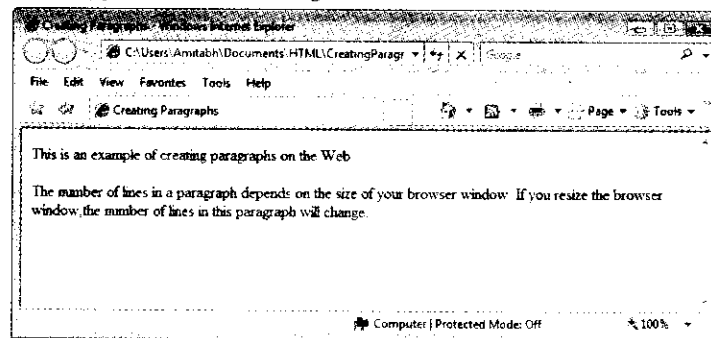


Figure 2.14: The Paragraph Tag

In Figure 2.14, you can observe that the browser skips some additional vertical spaces when a new paragraph begins.

Working with Images

An image plays a very important role on a Web page by making it more attractive and enhancing its overall look. They help to break the monotony of text in a Web page besides adding a visual element that might attract the user to a particular part of the page.

In this section, you learn to insert an image on a Web page, display alternate text for an image, add a border to an image, align an image, use images as links, and create image maps.

Let's first learn how to insert an image on a Web page.

Inserting an Image on a Web Page

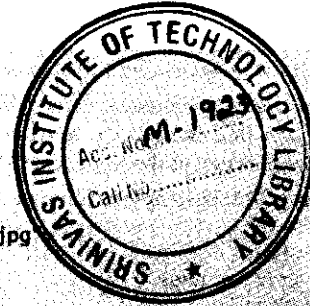
You can draw the attention of the user more easily by inserting an image in addition to text. HTML provides a facility to insert images in Web pages by using the `` tag. The `` tag uses the `src` attribute, which takes an image file as a value. If your image is in the same folder where your HTML file is stored, then there is no need to specify the full path of the image file in the `src` attribute. However, if your image is in some other folder, then you must specify the full path of the image file in the `src` attribute of the `` tag.

Let's create a Web page, named `InsertingImage.html` to learn how to insert an image on a Web page. You can find the `InsertingImage.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.10 shows the code of the `InsertingImage.html` page:

Listing 2.10: Inserting an Image on a Web Page

```
<html>
<head>
<title>Image
</title>
</head>
<body>
<h1>Inserting an Image in a Web Page</h1>

</body>
</html>
```



NOTE


In the `src` attribute of the `` tag, we have specified the path of the image (in our computer) we are adding to our Web page. Similarly, you need to specify the path of the image (in your computer) you want to add to your Web page. The image that you want to add to your Web page must exist in your computer.

The output of Listing 2.10 appears, as shown in Figure 2.15:



Figure 2.15: An Image in a Web Page

NOTE

If the image does not exist in your computer, a cross mark inside a box appears () in place of image.

In Figure 2.15, you can observe the inserted image on a Web page. Now, let's learn to display the alternate text for an image.

Displaying Alternate Text for an Image

Sometimes, an inserted image does not appear in the Internet Explorer (or the Web browser you are using). This may happen because of the incorrect file path or invalid file name. In that case, you can specify alternate text for an image that is displayed even if the image is not visible on the Web page. You can provide a short description about the image in the alternate text. To apply an alternate text to an image, use the `alt` attribute of the `` tag. If the image is visible, you can see the alternate text by placing the mouse pointer over the image.

Let's create a Web page, named `AlternateText.html` to learn how to display an alternate text for an image. You can find the `AlternateText.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.11 shows the code of the `AlternateText.html` page:

Listing 2.11: Displaying Alternate Text for an Image

```
<html>
<head>
<title>Image
</title>
</head>
<body>
<h1>Displaying Alternate Text for an Image</h1>

</img>
</body>
</html>
```

When you open this page, the output appears, as shown in Figure 2.16:



Figure 2.16: Displaying Alternate Text for an Image

In Figure 2.16, you can observe that an alternate text is displayed when the mouse pointer moves over the image.

NOTE

If the image does not appear in the Internet Explorer, the alternate text still appears beside the cross mark (Happy New Year 2008).

Next, let's learn to add a border to an image.

Adding a Border to an Image

If you want, you can also add a border to an Image according to your requirements. To add a border to an image, use the border attribute of the tag.

The ImageBorder.html Web page displays how to add borders to an image. You can find the ImageBorder.html file in the Code\HTML\Chapter 2 folder on the CD. Listing 2.12 shows the code of the ImageBorder.html page:

Listing 2.12: Adding a Border to an Image

```
<html>
<head>
<title>Image
</title>
</head>
<body>
<h1>Adding a Border to an Image</h1>

</img>
</body>
</html>
```

When you open this page the output appears, as shown in Figure 2.17:



Figure 2.17: Adding a Border to an Image

In Figure 2.17, you can observe that a border is added surrounding the image.

NOTE

By default, an image displays without a border.

Now, let's learn to align an image in an HTML document.

Aligning an Image

A Web page can have a combination of both text and images. If you are creating a Web page with both text and images, you need to align the images as per your requirements. To align an image on a Web page, use the align attribute of the tag. The align attribute of the tag takes one of the following three values: left, right, or center.

Let's create a Web page, named AligningImage.html to learn how to align an image on the Web page. You can find the AligningImage.html file in the Code\HTML\Chapter 2 folder on the CD. Listing 2.13 shows the code of the AligningImage.html page:

Listing 2.13: Aligning an Image

```
<html>
<head>
<title>Image
</title>
</head>
<body>
<h1>Aligning An Image</h1>

</img>
</body>
</html>
```

When you open this page, the output appears, as shown in Figure 2.18:



Figure 2.18: Aligning an Image

NOTE

By default, an image aligns to the left of the Web page.

In Figure 2.18, you can observe that the image is aligned to the right of the Internet Explorer.

Using Images as Links

You can also create hyperlink images similar to hyperlink text. To create a hyperlink image, first create an anchor tag (<a>) and give the reference of an HTML page through the href attribute of the <a> tag. Inside the <a> tag, create a tag and give the reference of an image file through the src attribute of the tag. When you click the image, the referenced page opens in the Internet Explorer.

Let's create a Web page named ImageLink.html to learn how to use images as links. You can find the ImageLink.html file in the Code\HTML\Chapter 2 folder on the CD. Listing 2.14 shows the code of ImageLink.html page:

Listing 2.14: Using Images as Links

```
<html>
<head>
<title>Image
</title>
</head>
<body>
<center>
<h1>Both the Image and Text can be a Link</h1>
<a href="Page1.html">

</img>
</a>
</center>
</body>
</html>
```

The output of this page appears, as shown in Figure 2.19:

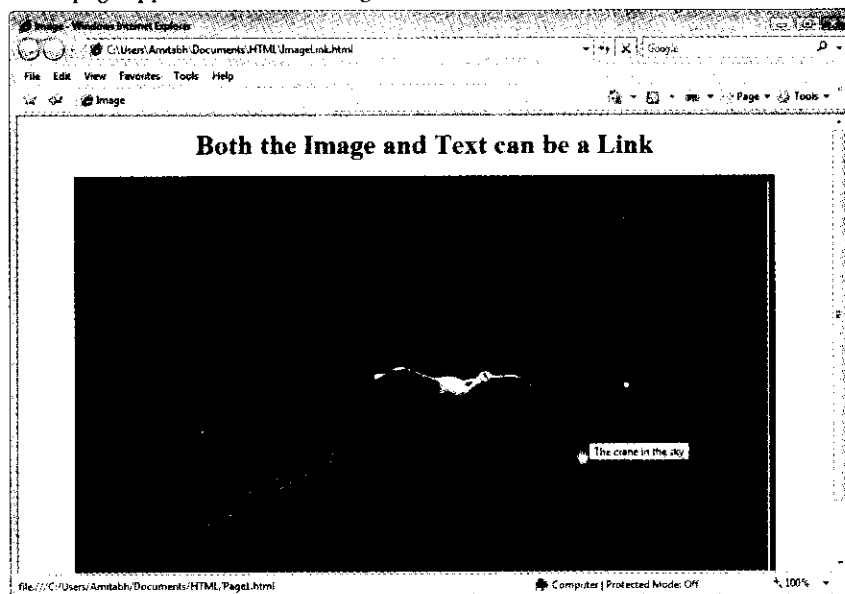


Figure 2.19: Linking an Image

In Figure 2.19, you can observe that the mouse pointer changes its appearance to hand shape when we place the mouse pointer over the image. This means that the image has become a hyperlink.

NOTE

If the image does not appear in the Internet Explorer, you can click the alternate text to link other Web pages.

When you click the image, the referenced page appears, as shown in Figure 2.20:

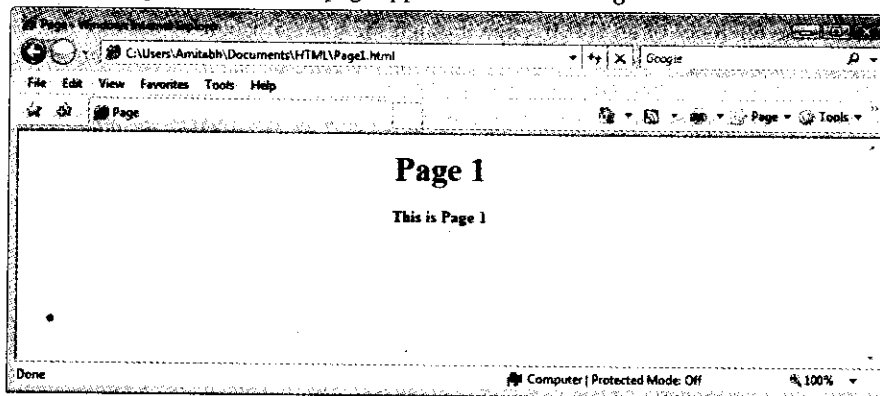


Figure 2.20: Referenced Page of the Image

Let's learn to create an image map in the next section.

Creating Image Maps

When a hyperlink is created on an image, you can open the Web page specified by the hyperlink. However, HTML also provides the facility to link multiple Web pages to a single image through an Image Map. Image Map is a technique that divides the image into multiple sections and allows linking of each section to different Web pages. Linked regions of an Image Map are called hot regions and each hot region is associated with an HTML file that is loaded when the hot region is clicked. hot regions are specified by the shape attribute of the area tag (<area>).

Let's create a Web page, named ImageMap.html to learn how to create an image map. You can find the ImageMap.html file in the Code\HTML\Chapter 2 folder on the CD. Listing 2.15 shows the code of the ImageMap.html page:

Listing 2.15: Creating Image Maps

```
<html>
<head>
<title>Image
</title>
</head>
<body>
</img>
<map name="MyMap">
<area shape="polygon"
coords="19,44,45,11,87,37,82,76,49,98" href="Area1.html">
<area shape="rect" coords="28,12,24,19" href="Area2.html">
<area shape="circle" coords="23,21,23" href="Area3.html">
</map>
</body>
</html>
```

The output of Listing 2.15 appears, as shown in Figure 2.21:



Figure 2.21: Image Map in a Web Page

Move the mouse pointer over the different sections of the image. When you move the mouse pointer over a hot region present on the image, the mouse pointer changes its appearance to hand shape, as you can see in Figure 2.21. When you click a hot region present on the image, the referenced page of the concerned hot region opens, as shown in Figure 2.22:

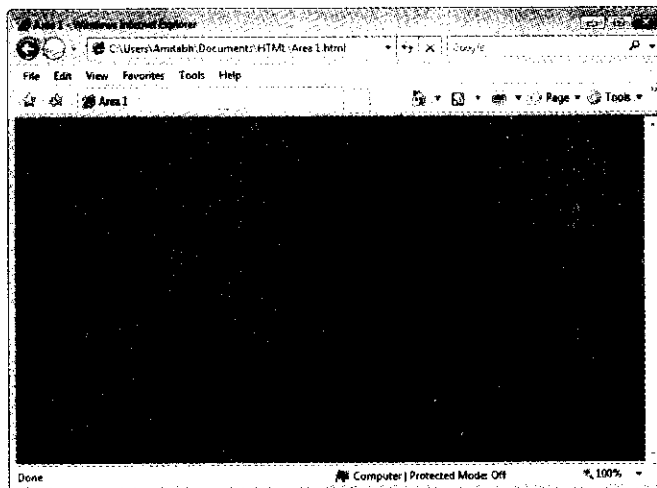


Figure 2.22: Reference Page of the Image Map

NOTE

To navigate to the other referenced pages, you need to create the referenced pages that you have specified in the code.

In the next section, you learn the process of working with tables.

Working with Tables

An HTML table allows you to arrange data, such as texts, images, links, forms, and other tables. In a table, data is arranged in the form of rows and columns of cells. A table can have an associated caption to provides a short

description of the purpose of the table. You can specify headings of a table and set the table border and color. You can also set the width of the table and its columns, as well as alignment and cell content as per the requirement. In addition, you can set the distance between cell borders and the content within the cell through the cellpadding attribute of the <table> tag. You can set the distance between the cell borders through Cell Spacing. If you want, you can also create multiple tables inside a table and combine rows and columns in a table using the rowspan and colspan attributes of the <table> tag respectively.

Let's learn about these in detail, next.

Creating a Table

A pair of <table> and </table> tags represents an HTML table and a pair of <tr> and </tr> tags represents a row in a table. A pair of <th> and </th> tags is used to add a column heading in a column and pairs of <td> and </td> tags are used to add data values in the column. While creating a table, the number of pairs of <tr> and </tr> tags added between the <table> and </table> tags determines the number of rows in the table. In the same way, each pair of <tr> and </tr> tags contains several pairs of <td> and </td> tags (or pairs of <th> and </th> tags for the row containing column headings) to determine the number of columns in the table.

Let's create a Web page, named Table.html that contains a table. You can find the Table.html file in the Code\HTML\Chapter 2 folder on the CD. Listing 2.16 shows the code of the Table.html page:

Listing 2.16: Creating a Table

```
<html>
<head>
<title>Table</title>
</head>
<body bgcolor="aliceblue">
<table>
<tr>
<td>Sumit Saxena</td>
<td>15-03-1983</td>
<td>Flat No. 303, Shipra Suncity, Ghaziabad</td>
</tr>
<tr>
<td>Amitabh Kumar</td>
<td>22-02-1984</td>
<td>H.No- 125, Kalkaji, Delhi</td>
</tr>
<tr>
<td>Rohit Jandial</td>
<td>05-07-1983</td>
<td>Flat No- 324, South Ext, Delhi</td>
</tr>
<tr>
<td>Avantika Srivastava</td>
<td>10-12-1984</td>
<td>H.No-541, Vikas Puri, Delhi</td>
</tr>
</table>
</body>
</html>
```

The output of Listing 2.16 appears, as shown in Figure 2.23:

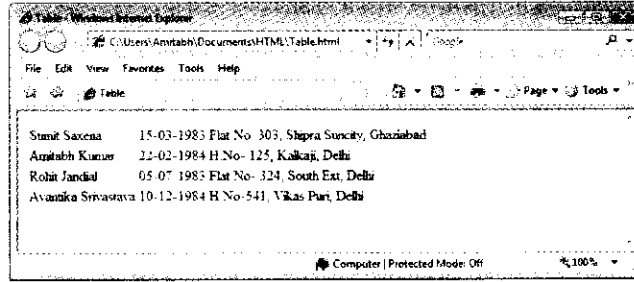


Figure 2.23: Creating a Table

As you can observe in Figure 2.23, a table is created on the Web page.

Specifying a Caption to a Table

A caption is a short description of a table that provides a brief explanation about it and helps you to understand its purpose. For example, when you create table containing the details about students, you can specify a relevant caption to the it, such as Student Details. A caption is specified to a table with the <caption> tag.

Let's create a Web page, named TableCaption.html to learn how to set a table caption. You can find the TableCaption.html file in the Code\HTML\Chapter 2 folder on the CD. Listing 2.17 shows the code of the TableCaption.html page:

Listing 2.17: Specifying a Caption to a Table

```

<html>
<head>
<title>Table</title>
</head>
<body bgcolor="aliceblue">
<table>
<caption>
<h2>Student Details</h2>
</caption>
<tr>
<td>Sumit Saxena</td>
<td>15-03-1983</td>
<td>Flat No. 303, Shipra Suncity, Ghaziabad</td>
</tr>
<tr>
<td>Amitabh Kumar</td>
<td>22-02-1984</td>
<td>H.No- 125, Kalkaji, Delhi</td>
</tr>
<tr>
<td>Rohit Jandial</td>
<td>05-07-1983</td>
<td>Flat No- 324, South Ext, Delhi</td>
</tr>
<tr>
<td>Avantika Srivastava</td>
<td>10-12-1984</td>
<td>H.No-541, Vikas Puri, Delhi</td>
</tr>
</table>
</body>
</html>

```

The output of Listing 2.17 appears, as shown in Figure 2.24:

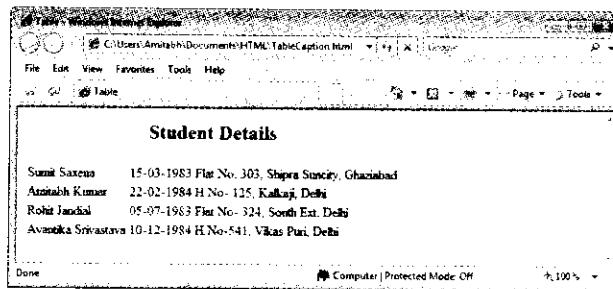


Figure 2.24: Setting Table Caption

In Figure 2.24, you can observe that a caption is added to the table, named as Student Details.

NOTE

By default, the table caption is aligned to center of the table.

Now, let's learn to add a table heading in detail.

Adding a Table Heading

A table heading is used to specify the title of a particular cell. For example, when you create a student database containing student name, date of birth, and address fields, you can specify these fields as table headers to facilitate better understanding.

Let's create a Web page, named TableHeading.html to learn how to add a table heading. You can find the TableHeading.html file in the Code\HTML\Chapter 2 folder on the CD. Listing 2.18 shows the code of the TableHeading.html page:

Listing 2.18: Adding a Table Heading

```
<html>
<head>
<title>Table Heading</title>
</head>
<body bgcolor="aliceblue">
<table>
<caption>
<h2>Student Details</h2>
</caption>
<th>Name</th>
<th>Date of Birth</th>
<th>Address </th>
<tr>
<td>Sumit Saxena</td>
<td>15-03-1983</td>
<td>Flat No. 303, Shipra Suncity, Ghaziabad</td>
</tr>
<tr>
<td>Amitabh Kumar</td>
<td>22-02-1984</td>
<td>H.No- 125, Kalkaji, Delhi</td>
</tr>
<tr>
<td>Rohit Jandial</td>
<td>05-07-1983</td>
<td>Flat No- 324, South Ext, Delhi</td>
</tr>
<tr>
```



```

<td>Avantika Srivastava</td>
<td>10-12-1984</td>
<td>H.No-541, Vikas Puri, Delhi</td>
</tr>
</table>
</body>
</html>

```

When you open this page, the output appears, as shown in Figure 2.25:

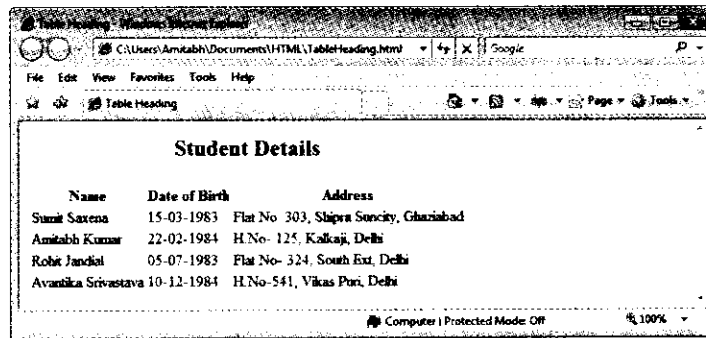


Figure 2.25: Adding a Table Heading

In Figure 2.25, you can observe that the following headings are added to the table: Name, Date of Birth, and Address.

NOTE

By default, the table headings are aligned center in bold font.

In the next section, you learn the process of setting a table border in an HTML document.

Setting the Table Border

When you create a table, it appears without border and looks incomplete. You can add a border to a table with the border attribute of the <table> tag. In addition to add a border to a table, you can also set the thickness of the border as per the need.

Let's create a Web page, named TableBorder.html to learn how to set the table border. You can find the TableBorder.html file in the Code\HTML\Chapter 2 folder on the CD. Listing 2.19 shows the code of the TableBorder.html page:

Listing 2.19: Setting the Table Border

```

<html>
<head>
<title>Table</title>
</head>
<body bgcolor="aliceblue">
<table border="1">
<caption>
<h2>Student Details</h2>
</caption>
<th>Name</th>
<th>Date of Birth</th>
<th>Address </th>
<tr>
<td>Sumit Saxena</td>
<td>15-03-1983</td>
<td>Flat No. 303, Shipra Suncity, Ghaziabad</td>
</tr>

```

```

<tr>
<td>Amitabh Kumar</td>
<td>22-02-1984</td>
<td>H.No- 125, Kalkaji, Delhi</td>
</tr>
<tr>
<td>Rohit Jandial</td>
<td>05-07-1983</td>
<td>Flat No- 324, South Ext, Delhi</td>
</tr>
<tr>
<td>Avantika Srivastava</td>
<td>10-12-1984</td>
<td>H.No-541, Vikas Puri, Delhi</td>
</tr>
</table>
</body>
</html>

```

When you open this page, the output appears, as shown in Figure 2.26:

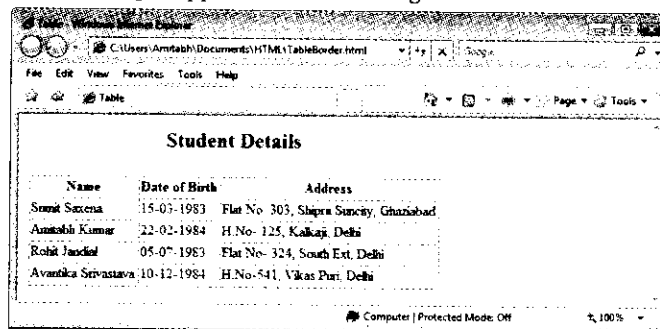


Figure 2.26: Setting the Table Border

In Figure 2.26, you can observe that a border is added to the table.

NOTE

By default, the table border is set to no or 0.

Now, let's learn to align a table and cell content.

Aligning a Table and Cell Content

You can align your table with the `align` attribute of the `<table>` tag. You can also align the cell content inside a table by specifying the `align` attribute and its value inside the `<th>` tag or `<td>` tag.

Aligning a Table in a Web Page

While creating a Web page, when you have both text and table, you can align your table as per the requirement. Table alignment can be done in one of the three different ways, such as, left, right, and center.

Let's create a Web page, named `AlignTable.html` to learn how to align a table in a Web page. You can find the `AlignTable.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.20 shows the code of the `AlignTable.html` page:

Listing 2.20: Aligning a Table in a Web Page

```

<html>
<head>
<title>Table</title>
</head>

```

```

<body bgcolor="aliceblue">
<table border="1" align="right">
<caption>
<h2>Student Details</h2>
</caption>
<th>Name</th>
<th>Date of Birth</th>
<th>Address </th>
<tr>
<td>Sumit Saxena</td>
<td>15-03-1983</td>
<td>Flat No. 303, Shipra Suncity, Ghaziabad</td>
</tr>
<tr>
<td>Amitabh Kumar</td>
<td>22-02-1984</td>
<td>H.No- 125, Kalkaji, Delhi</td>
</tr>
<tr>
<td>Rohit Jandial</td>
<td>05-07-1983</td>
<td>Flat No- 324, South Ext, Delhi</td>
</tr>
<tr>
<td>Avantika Srivastava</td>
<td>10-12-1984</td>
<td>H.No-541, Vikas Puri, Delhi</td>
</tr>
</table>
</body>
</html>

```

The output of Listing 2.20 appears, as shown in Figure 2.27:

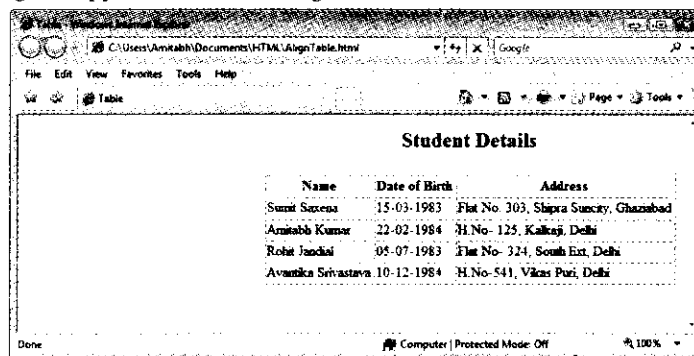


Figure 2.27: Aligning a Table

NOTE

By default, the table in a Web page is aligned to the left.

In Figure 2.27, you can observe that the table is aligned to the right of the Internet Explorer.

Aligning the Cell Content

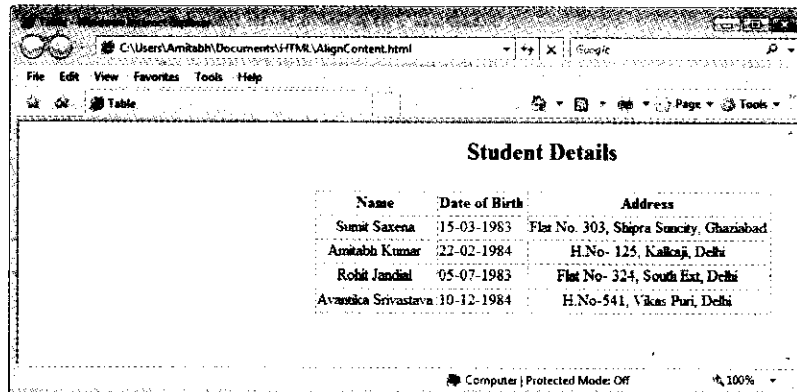
You can also align the cell content in one of three different ways, such as left, right, and center similar to a table. Let's create a Web page, named `AlignContent.html` to align the cell content of a table. You can find the

AlignContent.html file in the Code\HTML\Chapter 2 folder on the CD. Listing 2.21 shows the code of the AlignContent.html page:

Listing 2.21: Aligning the Cell Content

```
<html>
<head>
<title>Table</title>
</head>
<body bgcolor="aliceblue">
<table border="1" align="right">
<caption>
<h2>Student Details</h2>
</caption>
<th>Name</th>
<th>Date of Birth</th>
<th>Address </th>
<tr>
<td align="center">Sumit Saxena</td>
<td>15-03-1983</td>
<td align="center">Flat No. 303, Shipra Suncity, Ghaziabad</td>
</tr>
<tr>
<td align="center">Amitabh Kumar</td>
<td>22-02-1984</td>
<td align="center">H.No- 125, Kalkaji, Delhi</td>
</tr>
<tr>
<td align="center">Rohit Jandial</td>
<td>05-07-1983</td>
<td align="center">Flat No- 324, South Ext, Delhi</td>
</tr>
<tr>
<td align="center">Avantika Srivastava</td>
<td>10-12-1984</td>
<td align="center">H.No-541, Vikas Puri, Delhi</td>
</tr>
</table>
</body>
</html>
```

The output of Listing 2.21 appears, as shown in Figure 2.28:



Name	Date of Birth	Address
Sumit Saxena	15-03-1983	Flat No. 303, Shipra Suncity, Ghaziabad
Amitabh Kumar	22-02-1984	H.No- 125, Kalkaji, Delhi
Rohit Jandial	05-07-1983	Flat No- 324, South Ext, Delhi
Avantika Srivastava	10-12-1984	H.No-541, Vikas Puri, Delhi

Figure 2.28: Aligning the Cell Content in a Table

In Figure 2.28, you can observe that the cell contents of the first and third columns are aligned to center.

NOTE

By default, the cell content in a table is aligned to the left.

Now, let's learn to set the table and its columns.

Setting the Width of a Table and Table Columns

If you are creating a table with large number of columns, which makes it difficult to see the records clearly, you can increase the table width to enhance visibility. You can also decrease the table width as per the requirement.

In addition, if your table contain large values and the small size of columns effects its visibility, you can increase the column width. You can also decrease the column width as per the requirement.

You can specify the width of both table and table columns using the width attribute of <table>, <th>, and <td> tags respectively.

Setting the Width of a Table

A table width can be expressed as an absolute value in pixels or as a percentage of screen width. If you have a table of graphics, such as a Navigation bar, it is better to define the table width in pixels. If you have a table of text, such as student record, it is better to define the table width in percentage.

Let's create a Web page, named TableWidth.html to learn how to set the table width. You can find the TableWidth.html file in the Code\HTML\Chapter 2 folder on the CD. Listing 2.22 shows the code of the TableWidth.html page:

Listing 2.22: Setting the Width of a Table

```
<html>
<head>
<title>Table</title>
</head>
<body bgcolor="aliceblue">
<table border="1" align="right" width="95%">
<caption>
<h2>Student Details</h2>
</caption>
<th>Name</th>
<th>Date of Birth</th>
<th>Address </th>
<tr>
<td align="center">Sumit Saxena</td>
<td>15-03-1983</td>
<td align="center">Flat No. 303, Shipra Suncity, Ghaziabad</td>
</tr>
<tr>
<td align="center">Amitabh Kumar</td>
<td>22-02-1984</td>
<td align="center">H.No- 125, Kalkaji, Delhi</td>
</tr>
<tr>
<td align="center">Rohit Jandial</td>
<td>05-07-1983</td>
<td align="center">Flat No- 324, South Ext, Delhi</td>
</tr>
<tr>
<td align="center">Avantika Srivastava</td>
<td>10-12-1984</td>
<td align="center">H.No-541, Vikas Puri, Delhi</td>
</tr>
</table>
</body>
</html>
```

The output of Listing 2.22 appears, as shown in Figure 2.29:

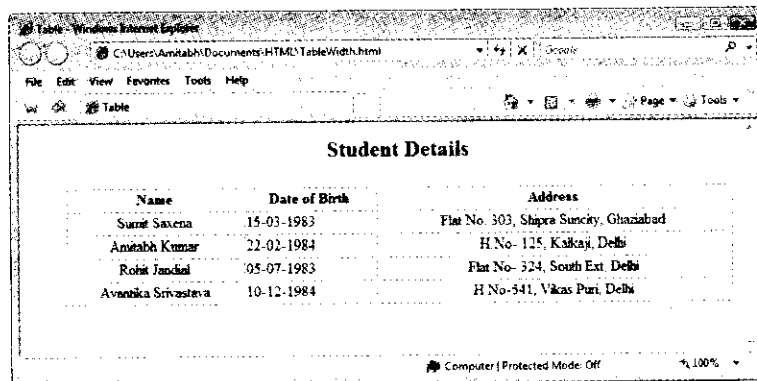


Figure 2.29: Setting the Table Width

In Figure 2.29, you can observe that the table is changed to the size specified in the width attribute of the `<table>` tag.

Setting the Width of Table Columns

A table column width can also be expressed as an absolute value in pixels, or as a percentage of screen width. If the cell content contains graphics, such as a Navigation button, it is better to define the column width in pixels. If the cell content contains text, such as student name, it is better to define the column width in percentage.

Let's create a Web page, named `ColumnWidth.html` to learn how to set the column width of a table. You can find the `ColumnWidth.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.23 shows the code of the `ColumnWidth.html` page:

Listing 2.23: Setting the Width of Table Columns

```
<html>
<head>
<title>Table</title>
</head>
<body bgcolor="aliceblue">
<table border="1" align="right" width="95%">
<caption>
<h2>Student Details</h2>
</caption>
<th width="20%">name</th>
<th width="20%">Date of Birth</th>
<th width="60%">Address</th>
<tr>
<td align="center">Sumit Saxena</td>
<td>15-03-1983</td>
<td align="center">Flat No. 303, Shipra Suncity, Ghaziabad</td>
</tr>
<tr>
<td align="center">Amitabh Kumar</td>
<td>22-02-1984</td>
<td align="center">H.No- 125, Kalkaji, Delhi</td>
</tr>
<tr>
<td align="center">Rohit Jandial</td>
<td>05-07-1983</td>
<td align="center">Flat No- 324, South Ext, Delhi</td>
</tr>
<tr>
<td align="center">Avantika Srivastava</td>
```

```

<td>10-12-1984</td>
<td align="center">H.No-541, Vikas Puri, Delhi</td>
</tr>
</table>
</body>
</html>

```

The output of Listing 2.23 appears, as shown in Figure 2.30:

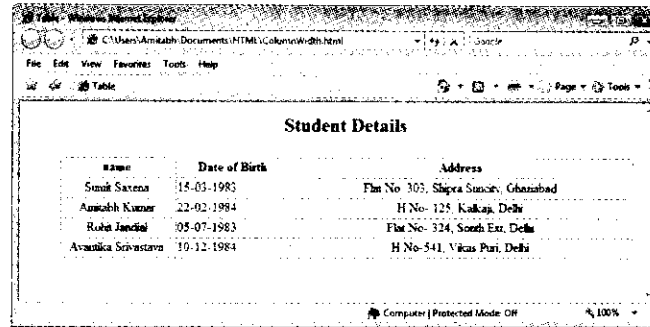


Figure 2.30: Setting the Column Width

In Figure 2.30, you can observe that the width of the table columns is changed to the size specified in the width attribute of the <th> tag in Listing 2.23.

Changing the Background Color of a Table

If you want to change the background color for the entire table, use the bgcolor attribute in the <table> tag. You can also change the color for the entire row by inserting the bgcolor attribute inside the <tr> tag. In addition, you can change the color for the individual cell by inserting the bgcolor attribute inside the <td> tag. To specify a color for the bgcolor attribute, you can use a RGB value, such as #22dd55 and #22ddaa, as well as color name, such as white and yellow.

Let's create a Web page, named BackgroundColor.html to learn how to change the background color of a table. You can find the BackgroundColor.html file in the Code\HTML\Chapter 2 folder on the CD. Listing 2.24 shows the code of the BackgroundColor.html page:

Listing 2.24: Changing the Background Color of a Table

```

<html>
<head>
<title>Table</title>
</head>
<body bgcolor="aliceblue">
<table border="1" align="right" width="95%" bgcolor="pink">
<caption>
<h2>Student Details</h2>
</caption>
<th width="20%">name</th>
<th width="20%">Date of Birth</th>
<th width="60%">Address</th>
<tr>
<td align="center">Sumit Saxena</td>
<td>15-03-1983</td>
<td align="center">Flat No. 303, Shipra suncity, Ghaziabad</td>
</tr>
<tr>
<td align="center">Amitabh Kumar</td>
<td>22-02-1984</td>
<td align="center">H.No- 125, Kalkaji, Delhi</td>

```

```

</tr>
<tr>
<td align="center">Rohit Jandial</td>
<td>05-07-1983</td>
<td align="center">Flat No- 324, South Ext, Delhi</td>
</tr>
<tr>
<td align="center">Avantika Srivastava</td>
<td>10-12-1984</td>
<td align="center">H.No-541, Vikas Puri, Delhi</td>
</tr>
</table>
</body>
</html>

```

The output of Listing 2.24 appears, as shown in Figure 2.31:

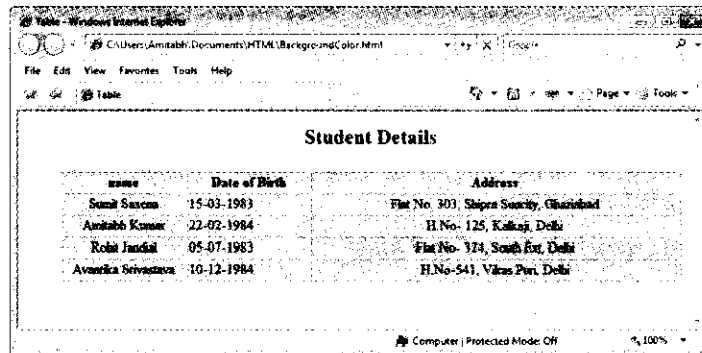


Figure 2.31: Changing the Background Color of a Table

NOTE

By default, the table background color is same as the background color of the page.

In Figure 2.31, you can observe that the background color of the table is changed to pink. Let's learn to set cell padding and cell spacing in the next section.

Setting Cell Padding and Cell Spacing

If you want extra space between your cell content and cell border, you can specify it by using cell padding. In addition, if you want extra space between cell borders in the table, you can specify it by using cell spacing. Let's learn about these in detail in the following sections.

Setting Cell Padding

Cell padding is used to specify the space between the edges of the cell and cell content in terms of pixels. You can set the cell padding with the `cellpadding` attribute of the `<table>` tag.

Let's create a Web page, named `CellPadding.html` to learn how to set cell padding. You can find the `CellPadding.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.25 shows the code of the `CellPadding.html` page:

Listing 2.25: Setting Cell Padding

```

<html>
<head>
<title>Table</title>
</head>
<body bgcolor="aliceblue">
<table border="1" align="right" width="95%" bgcolor="pink" cellpadding="8">

```



```

<caption>
<h2>Student Details</h2>
</caption>
<th width="20%">name</th>
<th width="20%">Date of Birth</th>
<th width="60%">Address</th>
<tr>
<td align="center">Sumit Saxena</td>
<td>15-03-1983</td>
<td align="center">Flat No. 303, Shipra Suncity, Ghaziabad</td>
</tr>
<tr>
<td align="center">Amitabh Kumar</td>
<td>22-02-1984</td>
<td align="center">H.No- 125, Kalkaji, Delhi</td>
</tr>
<tr>
<td align="center">Rohit Jandial</td>
<td>05-07-1983</td>
<td align="center">Flat No- 324, South Ext, Delhi</td>
</tr>
<tr>
<td align="center">Avantika Srivastava</td>
<td>10-12-1984</td>
<td align="center">H.No-541, Vikas Puri, Delhi</td>
</tr>
</table>
</body>
</html>

```

The output of Listing 2.25 appears, as shown in Figure 2.32:

name	Date of Birth	Address
Sumit Saxena	15-03-1983	Flat No. 303, Shipra Suncity, Ghaziabad
Amitabh Kumar	22-02-1984	H.No- 125, Kalkaji, Delhi
Rohit Jandial	05-07-1983	Flat No- 324, South Ext, Delhi
Avantika Srivastava	10-12-1984	H.No-541, Vikas Puri, Delhi

Figure 2.32: Setting Cell Padding

In Figure 2.32, you can observe that the space between the table boundaries and content is changed.

Setting Cell Spacing

Cell spacing is used to specify the space between the borders of cells. You can set the cell spacing with the `cellspacing` attribute of the `<table>` tag.

Let's create a Web page, named `CellSpacing.html` to learn how to set cell spacing. You can find the `CellSpacing.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.26 shows the code of the `CellSpacing.html` page:

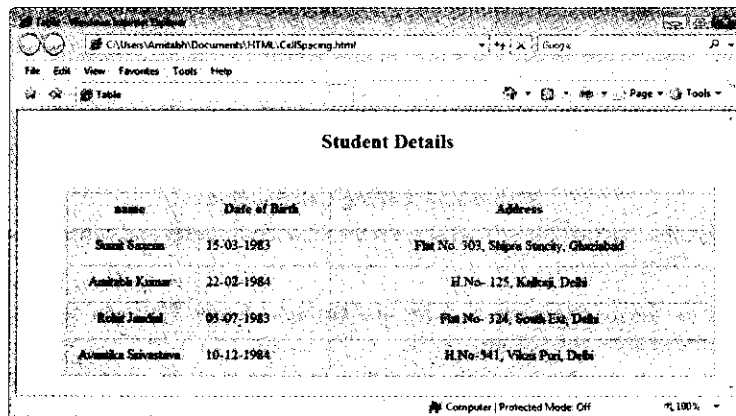
Listing 2.26: Setting Cell Spacing

```

<html>
<head>
<title>Table</title>
</head>
<body bgcolor="aliceblue">
<table border="1" align="right" width="95%" bgcolor="pink" cellpadding="8"
cellspacing="6">
<caption>
<h2>Student Details</h2>
</caption>
<th width="20%">name</th>
<th width="20%">Date of Birth</th>
<th width="60%">Address</th>
<tr>
<td align="center">sumit Saxena</td>
<td>15-03-1983</td>
<td align="center">Flat No. 303, Shipra Suncity, Ghaziabad</td>
</tr>
<tr>
<td align="center">Amitabh Kumar</td>
<td>22-02-1984</td>
<td align="center">H.No- 125, Kalkaji, Delhi</td>
</tr>
<tr>
<td align="center">Rohit Jandial</td>
<td>05-07-1983</td>
<td align="center">Flat No- 324, South Ext, Delhi</td>
</tr>
<tr>
<td align="center">Avantika Srivastava</td>
<td>10-12-1984</td>
<td align="center">H.No-541, Vikas Puri, Delhi</td>
</tr>
</table>
</body>
</html>

```

The output of Listing 2.26 appears, as shown in Figure 2.33:



name	Date of Birth	Address
Sumit Saxena	15-03-1983	Flat No. 303, Shipra Suncity, Ghaziabad
Amitabh Kumar	22-02-1984	H.No- 125, Kalkaji, Delhi
Rohit Jandial	05-07-1983	Flat No- 324, South Ext, Delhi
Avantika Srivastava	10-12-1984	H.No-541, Vikas Puri, Delhi

Figure 2.33: Setting Cell Spacing

In Figure 2.33, you can observe that the space between the borders of cells is changed.

Spanning Rows and Columns

When you work with tables, it is sometimes required to combine one or two adjacent cells into one. The process of combining one or more adjacent cells into one is known as spanning cells. The vertical spanning of cells is known as rowspan and the horizontal spanning of cells is known as colspan.

Let's learn about these in detail.

Spanning Rows

You can span cells vertically by using the rowspan attribute with the <td> or <th> tag. The rowspan attribute requires a number that is equal to the number of cells you want to combine vertically in one column.

Let's create a Web page, named RowSpan.html to learn how to span rows in a table. You can find the RowSpan.html file in the Code\HTML\Chapter 2 folder on the CD. Listing 2.27 shows the code of the RowSpan.html page:

Listing 2.27: Spanning Rows

```
<html>

<head>
<title>Table</title>
</head>
<body bgcolor="aliceblue">
<table border="1" align="right" width="95%" bgcolor="pink" cellpadding="8"
cellspacing="6">
<caption>
<h2>Student Details</h2>
</caption>
<th rowspan="5">student name<br>
with <br>
Date of Birth<br>
and <br>
Address</th>
<tr>
<td align="center">Sumit Saxena</td>
<td>15-03-1983</td>
<td align="center">Flat No. 303, Shipra Suncity, Ghaziabad</td>
</tr>
<tr>
<td align="center">Amitabh Kumar</td>
<td>22-02-1984</td>
<td align="center">H.No- 125, Kalkaji, Delhi</td>
</tr>
<tr>
<td align="center">Rohit Jandial</td>
<td>05-07-1983</td>
<td align="center">Flat No- 324, South Ext, Delhi</td>
</tr>
<tr>
<td align="center">Avantika Srivastava</td>
<td>10-12-1984</td>
<td align="center">H.No-541, Vikas Puri, Delhi</td>
</tr>
</table>
</body>

</html>
```

The output of Listing 2.27 appears, as shown in Figure 2.34:

The screenshot shows a web browser window displaying a table titled "Student Details". The table has four columns: "Student Name", "Date of Birth", and "Address". The first cell in the first row, which contains the text "Student Name", "with", "Date of Birth", and "and", "Address", is a single cell that spans all four columns. This cell is followed by three columns of data for the first student: "Sumit Saxena", "15-03-1983", and "Flat No. 303, Shipra Suncity, Ghaziabad". The table continues with three more rows of student data.

Student Name with Date of Birth and Address			
Sumit Saxena	15-03-1983	Flat No. 303, Shipra Suncity, Ghaziabad	
Amitabh Kumar	22-02-1984	H.No- 125, Kalkaji, Delhi	
Rohit Jindal	05-07-1983	Flat No- 124, South Est, Delhi	
Arunika Srivastava	18-12-1984	H.No-541, Vikas Puri, Delhi	

Figure 2.34: Spanning Rows

NOTE

When you combine the rows the current cell is also included.

In Figure 2.34, you can observe that the first cell combines four columns into a single column.

Spanning Columns

You can span cells vertically by using the `colspan` attribute with the `<td>` or `<th>` tag. The `colspan` attribute requires a number that is equal to the number of cells you want to combine horizontally into one column.

Let's create a Web page, named `ColSpan.html` to learn how to span columns in a table. You can find the `ColSpan.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.28 shows the code of the `ColSpan.html` page:

Listing 2.28: Spanning Columns

```
<html>
<head>
<title>Table</title>
</head>
<body bgcolor="aliceblue">
<table border="1" align="right" width="95%" bgcolor="pink" cellpadding="8"
cellspacing="6">
<caption>
<h2>Student Details</h2>
</caption>
<th colspan="3">Student Name<br>
with <br>
Date of Birth<br>
and <br>
Address</th>
<tr>
<td align="center">Sumit Saxena</td>
<td align="center">15-03-1983</td>
<td align="center">Flat No. 303, Shipra Suncity, Ghaziabad</td>
</tr>
<tr>
<td align="center">Amitabh Kumar</td>
<td align="center">22-02-1984</td>
<td align="center">H.No- 125, Kalkaji, Delhi</td>
</tr>
<tr>
```

```

<td align="center">Rohit Jandial</td>
<td>05-07-1983</td>
<td align="center">Flat No- 324, South Ext, Delhi</td>
</tr>
<tr>
<td align="center">Avantika Srivastava</td>
<td>10-12-1984</td>
<td align="center">H.No-541, Vikas Puri, Delhi</td>
</tr>
</table>
</body>
</html>

```

The output of Listing 2.28 appears, as shown in Figure 2.35:

Student Name with Date of Birth and Address		
Sanku Sharma	15-05-1983	Flat No. 303, Shyva Society, Chaudhary
Avantika Kumar	23-02-1984	H.No- 125, Kirti, Delhi
Rohit Jandial	05-07-1983	Flat No- 324, South Ext, Delhi
Avantika Srivastava	10-12-1984	H.No-541, Vikas Puri, Delhi

Figure 2.35: Spanning Columns

NOTE

When you combine the columns the current cell is also included.

In Figure 2.35, you can observe that the first cell combines three rows into a single row. Let's learn about the process of nesting tables, next.

Nesting Tables

In simple terms, nesting refers to the placement of one thing inside another. For example, if you are creating a Web page with different types of tables placed one inside the others, you can nest all the tables.

Let's create a Web page, named `NestingTable.html` to learn how to nest one or more tables. You can find the `NestingTable.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.29 shows the code of the `NestingTable.html` page:

Listing 2.29: Nesting Tables

```

<html>
<head>
<title>Table</title>
</head>
<body bgcolor="aliceblue">
<table border="1" align="center">
<caption>

```

```

<h2>Student Details</h2>
</caption>
<tr>
<td><table border="1" bgcolor="pink">
<th>Name</th>
<th>Date of Birth</th>
<th>Address</th>
<tr>
<td>Sumit Saxena</td>
<td>15-03-1983</td>
<td>Flat No. 303, Shipra Suncity, Ghaziabad</td>
</tr>
<tr>
<td>Amitabh Kumar</td>
<td>22-02-1984</td>
<td>H.No- 125, Kalkaji, Delhi</td>
</tr>
<tr>
<td>Rohit Jandial</td>
<td>05-07-1983</td>
<td>Flat No- 324, South Ext, Delhi</td>
</tr>
<tr>
<td>Avantika Srivastava</td>
<td>10-12-1984</td>
<td>H.No-541, Vikas Puri, Delhi</td>
</tr>
</table></td>
<td><table border="1" bgcolor="silver">
<tr>
<th>Land Line No.</th>
<th>Mobile No.</th>
</tr>
<tr>
<td>0120-2546787</td>
<td>9910317699</td>
</tr>
<tr>
<td>011-2346787</td>
<td>9810415668</td>
</tr>
<tr>
<td>011-2446764</td>
<td>9914566565</td>
</tr>
<tr>
<td>011-2467890</td>
<td>9815464565</td>
</tr>
</table></td>
</tr>
</table>
</body>
</html>

```

The output of Listing 2.29 appears, as shown in Figure 2.36:

Name	Date of Birth	Address	Landline No.	Mobile No.
Samir Saxena	15-03-1983	Flat No. 303, Shyama Sanchay, Ghazipur	011-2546787	9910517099
Anjali Kumar	21-02-1984	H.No- 123, Kirti, Delhi	011-2345678	9810412345
Rohit Jindal	05-07-1983	Flat No- 324, South Ext, Delhi	011-2444789	991456789
Avaniha Srinivasa	10-12-1984	H.No- 541, Vikas Park, Delhi	011-3447890	981567890

Figure 2.36: Nesting Tables

In Figure 2.36, you can observe that two tables are nested into a single table.

Working with Frames

HTML provides the facility to divide the Internet Explorer into many sections by using frameset. A frameset in HTML is a way to display multiple Web pages in the same Internet Explorer, at the same time. You can create vertical and horizontal frames as well as set the frame border thickness and color as per the requirement. In addition, you can apply hyperlink to a frame.

Creating a Frame

You can create a frame by using `<frameset>` tag. Inside the `<frameset>` tag, you have to use `<frame>` tag. The `<frame>` tag uses an attribute named as `src`. The `src` attribute takes an html file, as a value, which you want to upload to that frame. You can create multiple `<frame>` tag inside a `<frameset>` tag.

Let's create a Web page, named `CreatingFrames.html` to learn how to create a frame. You can find the `CreatingFrames.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.30 shows the code of the `CreatingFrames.html` page:

Listing 2.30: Creating a Frame

```
<html>
<head>
<title>Frames</title>
</head>
<frameset cols="50% ,50%">
<frame src="Frame1.html">
<frame src="Frame2.html">
</frameset>
<noframes></noframes>
</html>
```

Now, create another Web page, named `Frame1.html` for the frame. You can find the `Frame1.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.31 shows the code of the `Frame1.html` page:

Listing 2.31: Creating Web Page for Frame 1

```
<html>
<head>
<title>Frames</title>
</head>
<body>
<h1 align="center">Frame 1</h1>
<h4 align="center">This is Frame 1</h4>
</body>
</html>
```

Now, create one more Web page, named `Frame2.html` for the frame. You can find the `Frame2.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.32 shows the code of the `Frame2.html` page:

Listing 2.32: Creating Web Page for Frame 2

```
<html>
<head>
<title>Frames</title>
</head>
<body>
<h1 align="center">Frame 2</h1>
<h4 align="center">This is Frame 2</h4>
</body>
</html>
```

The output of Listing 2.30 appears, as shown in Figure 2.37:

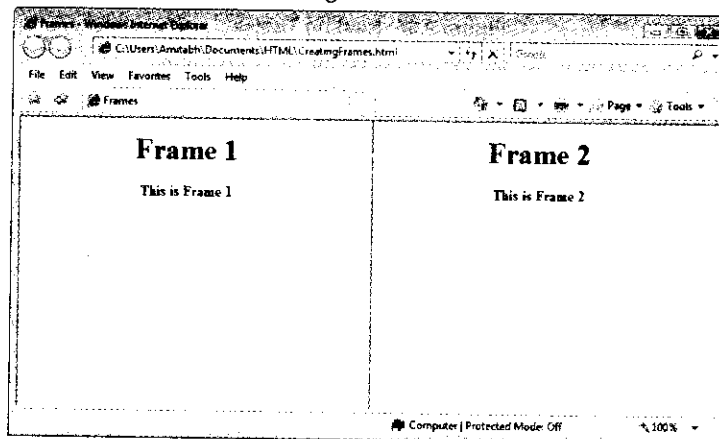


Figure 2.37: Frame in a Web Page

In Figure 2.37, you can observe that the Internet Explorer is divided into two frames: Frame 1 and Frame 2.

Creating Vertical and Horizontal Frames

Frames are the best way to show multiple Web pages into a single window as they divide the Internet Explorer into sections. You can divide the Internet Explorer in many vertical sections, horizontal sections, or a combination of both.

Creating Vertical Frames

You can divide the Internet Explorer into vertical sections by using the `cols` attribute of the `<frameset>` tag. The `cols` attribute takes the values in percentage separated by commas.

Let's create a Web page, named `VerticalFrames.html` to learn how to generate vertical frames. You can find the `VerticalFrames.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.33 shows the code of the `VerticalFrames.html` page:

Listing 2.33: Creating Vertical Frames

```
<html>
<head>
<title>Frames</title>
</head>
<frameset cols="33%,33%,33%">
<frame src="Frame1.html">
<frame src="Frame2.html">
<frame src="Frame3.html">
</frameset>
```



```
</noframes></noframes>
</html>
```

Now, create a Web page, named `Frame3.html` to learn how to generate a frame for the Web page. You can find the `Frame3.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.34 shows the code of the `Frame3.html` page:

Listing 2.34: Creating Web Page for Frame 3

```
<html>
<head>
<title>Frames</title>
</head>
<body>
<h1 align="center">Frame 3</h1>
<h4 align="center">This is Frame 3</h4>
</body>
</html>
```

The output of Listing 2.33 appears, as shown in Figure 2.38:

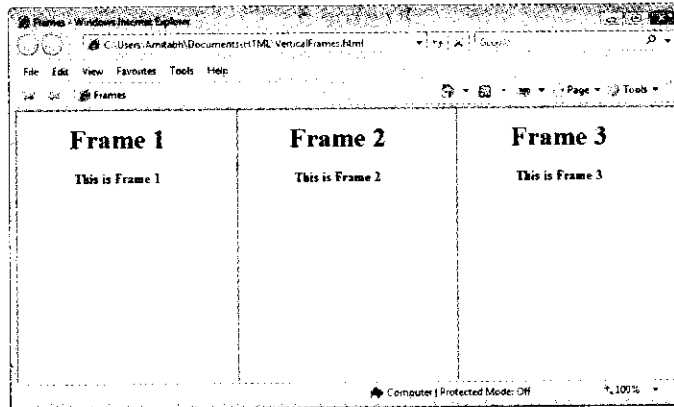


Figure 2.38: Creating Vertical Frames

In Figure 2.38, you can observe that the Internet Explorer is divided into three vertical frames: Frame 1, Frame 2, and Frame 3.

Creating Horizontal Frames

You can divide the Internet Explorer into horizontal sections by using the `rows` attribute of the `<frameset>` tag. The `rows` attribute takes the values in percentage separated by commas.

Let's create a Web page, named `HorizontalFrames.html` to learn how to generate horizontal frames. You can find the `HorizontalFrames.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.35 shows the code of the `HorizontalFrames.html` page:

Listing 2.35: Creating Horizontal Frames

```
<html>
<head>
<title>Frames</title>
</head>
<frameset rows="33%,33%,33%">
<frame src="Frame1.html">
<frame src="Frame2.html">
<frame src="Frame3.html">
</frameset>
</noframes></noframes>
</html>
```

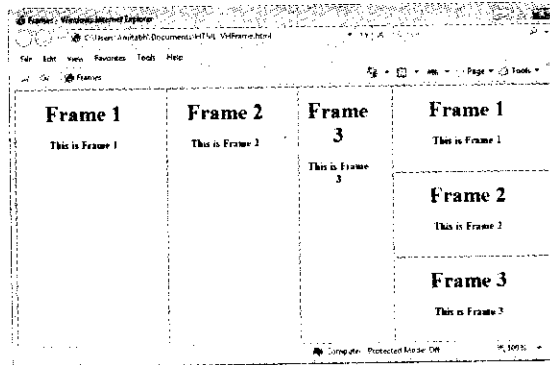



Figure 2.40: Creating Vertical and Horizontal Frames

In Figure 2.40, you can observe that the Internet Explorer is divided into three vertical frames and three horizontal frames.

Setting the Frame Border Thickness

You can also apply border to the frame using the `frameborder` attribute of the `<frameset>` tag. If you want to set a border to your frame, specify either `yes` or `1` as a value of the `frameborder` attribute. If you do not want to set a border to your frame, specify either `no` or `0` as a value of the `frameborder` attribute. By default, a frame border is applied to the frames. You can also set the thickness of the frame border using the `framespacing` attribute of the `<frameset>` tag.

Let's create a Web page, named `FrameBorderThickness.html` to learn how to set the frame border thickness. You can find the `FrameBorderThickness.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.37 shows the code of the `FrameBorderThickness.html` page:

Listing 2.37: Setting the Frame Border Thickness

```
<html>
<head>
<title>Frames</title>
</head>
<frameset cols="33%,33%,33%" frameborder="1" framespacing="10">
<frame src="Frame1.html">
<frame src="Frame2.html">
<frame src="Frame3.html">
</frameset>
<noframes></noframes>
</html>
```

The output of Listing 2.37 appears, as shown in Figure 2.41:

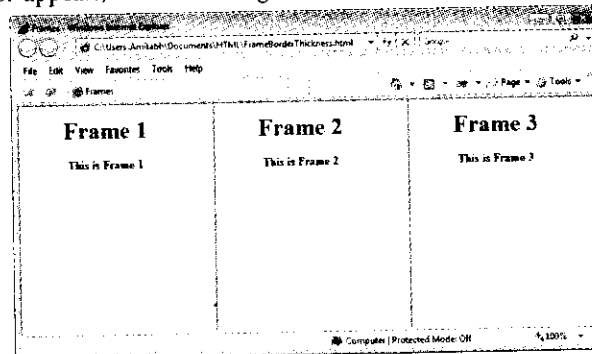


Figure 2.41: Setting the Thickness of the Frame Border

In Figure 2.41, you can observe that the thickness of the frame border is changed.

Applying Hyperlink Targets to a Frame

A hyperlink is used to create a link between Web pages. When you click a hyperlink, it takes you to the referenced section on the same Web page or another by using the `href` attribute of the anchor tag `<a>`.

You can specify the target location of the desired Web page by using the `target` attribute inside the `<a>` tag. You can use a name frame or predefined target name as hyperlink targets, which are explained next.

Using a Named Frame as Hyperlink Targets

Named frames are those frames for which names are assigned and they can be used as hyperlink targets by specifying a frame name to the `target` attribute. Therefore, when you click the hyperlink, the referenced page opens in the target frame specified in the `target` attribute.

Let's create a Web page, named `NamedFrame.html` to learn how to use a named frame as hyperlink targets. You can find the `NamedFrame.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.38 shows the code of the `NamedFrame.html` page:

Listing 2.38: Using a Named Frame as Hyperlink Targets

```
<html>
<head>
<title>Named Frames</title>
</head>
<frameset cols="25% ,75%">
<frame src="Mainpage.html">
<frame src="Defaultpage.html" name="display">
</frameset>
<noframes></noframes>
</html>
```

Now, create another page, named `Mainpage.html` that is referenced in the frame. You can find the `Mainpage.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.39 shows the code of the `Mainpage.html` page:

Listing 2.39: Creating the Main Page

```
<html>
<head>
<title>Page</title>
</head>
<body>
<a href= "Area 1.html" target="display">
<h1>Area 1</h1>
</a> <br>
<a href= "Area 2.html" target="display">
<h1>Area 2</h1>
</a> <br>
<a href= "Area 3.html" target="display">
<h1>Area 3</h1>
</a>
</body>
</html>
```

Now, create the `Defaultpage.html` that is referenced in the frame. You can find the `Defaultpage.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.40 shows the code of the `Defaultpage.html` page:

Listing 2.40: Creating the Default Page

```
<html>
<head>
<title>Page</title>
</head>
<body>
```

```
<h1 align="center">Click A Hyperlink</h1>
</body>
</html>
```

The output of Listing 2.38 appears, as shown in Figure 2.42:

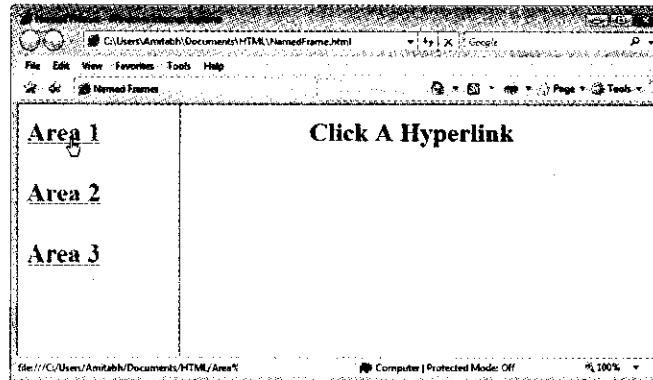


Figure 2.42: Applying Hyperlink Target Using a Named Frame

Click a hyperlink in the left frame, as shown in Figure 2.42. For example, if we click the Area 1 hyperlink, Area 1.html file is loaded in the right frame, as shown in Figure 2.43. The color of selected hyperlink changes from blue to purple.

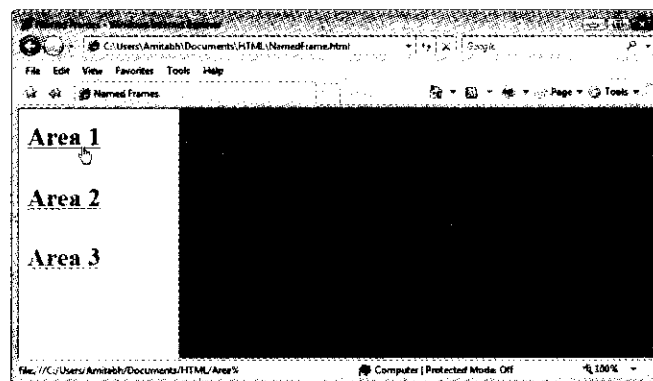


Figure 2.43: Frame with the Opened Referenced Page

In Figure 2.43, you can observe that the referenced Web page opens in the right frame.

Using a Predefined Target Name as Hyperlink Targets

When you do not want to open a Web page in a named frame, HTML provides the facility to use predefined target names to set the target locations of the desired Web page. Predefined target names take one of the four values: `_blank`, `_self`, `_parent`, and `_top` to the target attribute. So when you click the hyperlink, the referenced page opens in the target frame that you assign in the target attribute. For example, if you use `_blank`, the Web page opens in a new blank window.

Let's create a Web page, named `PredefinedTarget.html` to learn how to set a predefined target name as hyperlink targets. You can find the `PredefinedTarget.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.41 shows the code of the `PredefinedTarget.html` page:

Listing 2.41: Using a Predefined Target Name as Hyperlink Targets

```
<html>
<head>
<title>Page</title>
```

```
</head>
<body>
<a href="Area 1.html" target="_blank">
<h1>Area 1</h1>
</a>
<br>
<a href="Area 2.html" target="_blank">
<h1>Area 2</h1>
</a>
<br>
<a href="Area 3.html" target="_blank">
<h1>Area 3</h1>
</a>
</body>
</html>
```

The output of Listing 2.41 appears, as shown in Figure 2.44:

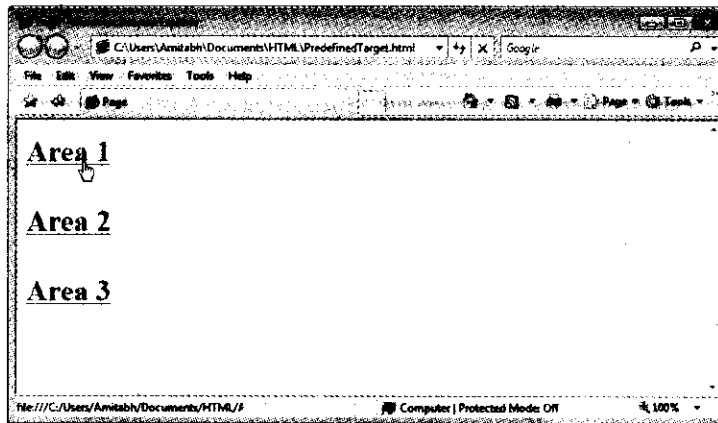


Figure 2.44: Hyperlink Target Using a Predefined Target Name

Figure 2.44 shows three hyperlinks on the Web pages: Area 1, Area 2, and Area 3. If you click the Area 1 hyperlink, the Area 1.html file is loaded in a new window, as shown in Figure 2.45:

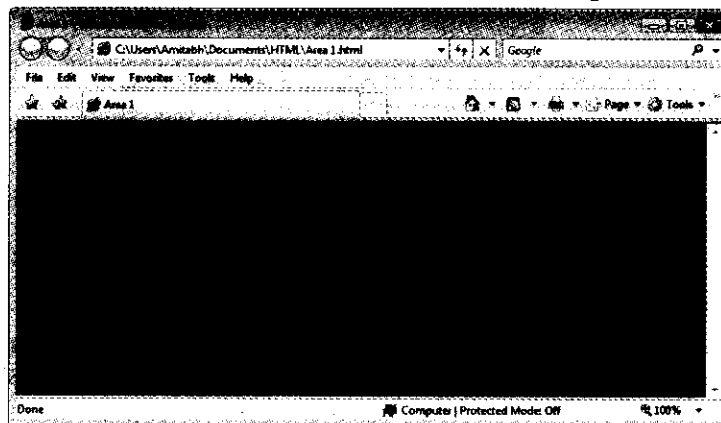


Figure 2.45: Referenced Page of the Frame

Introduction to Forms and HTML Controls

In the previous sections, you learned to create a Web page to display information. HTML is also designed to create forms that are the primary elements of a Web page and gather information from the user. The basic purpose of an HTML form is to allow the user to enter data on one end and then send the data on the other end through the Web server. It helps the user to purchase goods on the Internet and sign up for newsletters and mail accounts. The controls are the basic parts of an HTML form. The user fills forms by entering data in the text boxes, clicking the radio button, and clearing the check boxes. In addition, each form has a submit button that helps to send data through the server to the action URL.

In this section, you learn to work with HTML forms and controls. This includes creating an HTML form, adding controls to an HTML form, grouping the controls of HTML forms, and specifying a label for the control.

Let's first learn the process to create an HTML form.

Creating an HTML Form

An HTML form is a Web page that contains form elements. A basic form has three important parts: the `<form>` tag, which includes the URL of the script needed to process the form; the form elements, which are similar to the text fields; and the submit button, which sends the data on the server. All the input elements should be enclosed within the opening `<form>` and closing `</form>` tags. Table 2.7 lists a brief description of the attributes of the `<form>` tag:

Attribute	Value	Description
action	URL	Defines where to send the data when the submit button is clicked.
method	get post	Represents the HTTP method that sends data to the action URL. The default value for the method attribute is get.

Let's create a Web page, named `CreatingaForm.html` to learn how to create an HTML form. You can find the `CreatingaForm.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.42 shows the code of the `CreatingaForm.html` page:

Listing 2.42: Creating an HTML Form

```
<html>
<head>
<title>
Creating HTML Form
</title>
</head>
<body>
<form>
First Name:
<input type="text" name="firstname">
<br>
Last Name:
<input type="text" name="lastname">
<br>
<input type="submit" value="SUBMIT">
</form>
</body>
</html>
```

The output of Listing 2.42 appears, as shown in Figure 2.46:

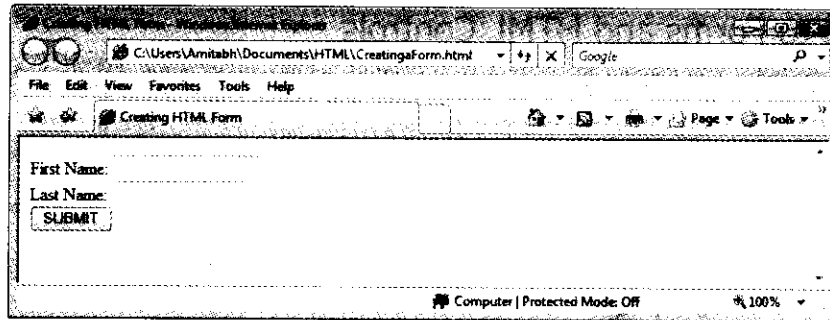


Figure 2.46: Creating an HTML Form

In Figure 2.46, you can observe that a basic HTML form is created with text boxes and the SUBMIT button.

Specifying the Action URL and Method to Send the Form

In HTML, you can direct the data that the user enters in the form to the server. You can do this by specifying the action URL and the method in your HTML code. Specified inside the `<form>` tag, action URL is the physical address of the server to which you want the user data to be redirected at the click of the submit button. You can specify two different submission methods for a form. The method is specified inside the `<form>` tag using the method attribute. Table 2.8 lists a brief description of the values of the method attribute:

Method	Description
method="get"	Indicates that the form data has to be encoded by the Web browser into a URL. This is the default method.
method="post"	Indicates that the form data appears within the message body.

Let's create a Web page, named `FormwithActionURL.html` to learn how to specify the action URL and method to send the form. You can find the `FormwithActionURL.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.43 shows the code of the `FormwithActionURL.html` page:

Listing 2.43: Specifying the Action URL and Method

```
<html>
<head>
<title>
Creating HTML Form
</title>
</head>
<body>
<form action="Example.html" method="post">
First Name:
<input type="text" name="firstname">
<br>
Last Name:
<input type="text" name="lastname">
<br>
<input type="submit" value="SUBMIT">
</form>
<p>
If you click the <b><u>"SUBMIT"</u></b> button you will be redirected to the
<b><u>Example.html</u></b> page.
</p>
</body>
</html>
```


The output of Listing 2.43 appears, as shown in Figure 2.47:

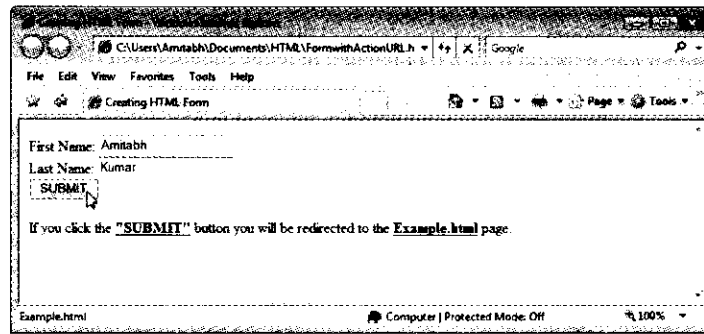


Figure 2.47: Submitting the Form Data

When you enter the data in the text boxes and click the SUBMIT button, you are redirected to another page, named `Example.html`, as shown in Figure 2.48:

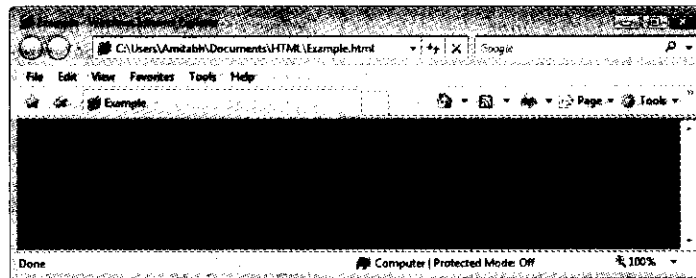


Figure 2.48: Resulting Page After the Submission of the Form Data

NOTE

Before clicking the Submit button, you need to create the `Example.html` page that is specified as the reference page in the code.

In Figure 2.48, you can observe that data has been successfully entered in the form.

Using the HTML Controls

A user interacts with forms through controls that are the basic element of a Web page form. The HTML controls help to make a Web page user friendly. A user can enter data in the Web page form with the help of the HTML controls. There are plenty of controls available in HTML and the controls that provide input to the form are created using the `<input>` tag and a different value for the `type` attribute.

Using the `<input>` Tag to Add Controls

The primary tag used in HTML to add controls is the `<input>` tag that defines the start of an input field where the user can enter data. The `<input>` tag prompts the user to enter data on the Web server. Table 2.9 lists a brief description of the attributes of the `<input>` tag:

Attribute	Value	Description
<code>align</code>	left right top texttop middle absmiddle	Defines the alignment of the text following the image. It is used only with <code>type="image"</code> .

Attribute	Value	Description
	baseline bottom absbottom	
alt	text	Defines an alternate text for the image. It is used only with type="image".
checked	checked	Indicates that the input element should be checked when it loads on the Web server. It is used only with type="checkbox" and type="radio".
disabled	disabled	Disables the input element when it loads on the Web server, so that the user cannot write text in it or select it. It cannot be used with type="hidden".
maxlength	number	Specifies the maximum number of characters allowed in a text field. It can be used only with type="text".
name	field_name	Defines a unique name for the input element. This attribute is required with type="button", type="checkbox", type="field", type="hidden", type="image", type="password", type="text", and type="radio".
readonly	readonly	Indicates that the value of this field cannot be modified. It can be used only with type="text".
src	URL	Defines the URL of the image to display. It can be used only with type="image".
type	button checkbox file hidden image password radio reset submit text	Indicates the type of the input element. The default value is "text".

NOTE

In HTML, the <input> tag has no end tag.

Let's learn to add a text field to a form in detail.

Adding a Text Field to a Form

The <input type="text"> tag is used to add a text field to a form. This tab provides a box to enter text that is transferred to the Web server. Let's create a Web page, named AddingTextField.html to learn how to add a text field to a form. You can find the AddingTextField.html file in the Code\HTML\Chapter 2 folder on the CD. Listing 2.44 shows the code of the AddingTextField.html page:

Listing 2.44: Adding a Text Field to a form

```
<html>
<head>
<title>
Adding Text Field
</title>
```

```

</head>
<body>
<form action="">
First name:
<input type="text" name="firstname" size="30" />
<br />
Last name:
<input type="text" name="lastname" size="30" />
</form>
</body>
</html>

```

When you open this page, the output appears, as shown in Figure 2.49:

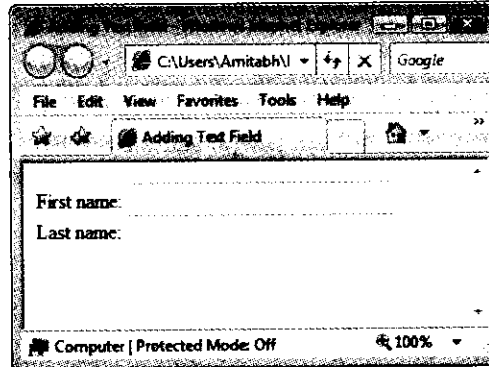


Figure 2.49: Text Field in the HTML Form

In Figure 2.49, you can observe that the text fields added to the form appear on the Web page.

Adding a Button

A button is a type attribute value of the `<input>` tag that is used to create a button on the HTML form.

To add a button on your Web page, add the `<input type="button">` tag in your HTML code. Let's create a Web page, named `AddingButton.html` to learn how to add a button control to a form. You can find the `AddingButton.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.45 shows the code of the `AddingButton.html` page:

Listing 2.45: Adding a Button

```

<html>
<head>
<title>
Adding Button
</title>
</head>
<body>
<form action="">
<input type="button" value="Hello world!">
</form>
</body>
</html>

```

When you open this page, the output appears, as shown in Figure 2.50:

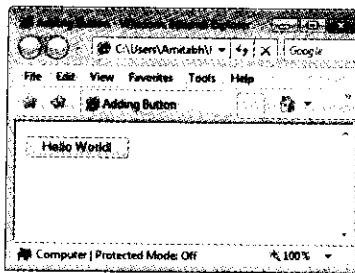


Figure 2.50: Button in the HTML Form

In Figure 2.50, you can observe that a button is added to your Web page.

Adding a Check Box

A check box is a small box with a check mark in it and a user can either select or clear it by a click. To add a check box to your Web page, add the `<input type="checkbox">` tag to your HTML code. Let's create a Web page, named `AddingCheckBox.html` to learn how to add a check box control to a form. You can find the `AddingCheckBox.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.46 shows the code of the `AddingCheckBox.html` page:

Listing 2.46: Adding a Check Box

```

<html>
<head>
<title>
Adding Check Box
</title>
</head>
<body>
<form action="">
<b>I love reading:</b>
<input type="checkbox" name="hobby" value="books">
</br>
<b>I love playing:</b>
<input type="checkbox" name="hobby" value="play" checked="">
</br>
<b>I love driving:</b>
<input type="checkbox" name="hobby" value="drive">
</form>
</body>
</html>

```

When you open this page, the output appears, as shown in Figure 2.51:

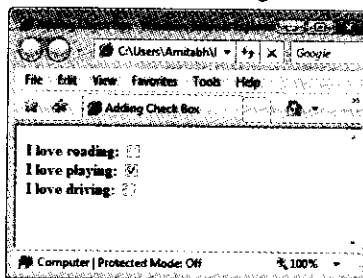


Figure 2.51: Check Box in the HTML Form

In Figure 2.51, you can observe that check box is added to your Web page.

Adding a Radio Button

A radio button is displayed as a circle and displays a dot in the middle when selected. To add radio button to your Web page, add the `<input type="radio">` tag to your HTML code. Let's create a Web page, named `AddingRadioButton.html` to learn how to add a radio button control to a form. You can find the `AddingRadioButton.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.47 shows the code of the `AddingRadioButton.html` page:

Listing 2.47: Adding a Radio Button

```
<html>
<head>
<title>
Adding Radio Button
</title>
</head>
<body>
<form action="">
YES:
<input type="radio" name="poll" value="yes"/>
</br>
NO:
<input type="radio" name="poll" value="no"/>
</form>
</body>
</html>
```

When you open this page, the output appears, as shown in Figure 2.52:

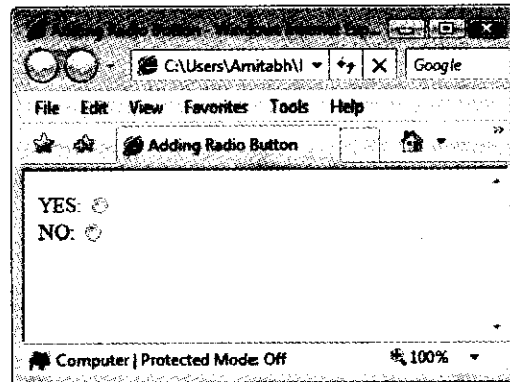


Figure 2.52: Radio Button in the HTML Form

NOTE

The difference between a checkbox and a radio button is that radio buttons work in mutually exclusive groups and only one radio button can be selected at a time.

In Figure 2.52, you can observe that radio button is added to your Web page.

Adding a Submit Button

A submit button is the most important control in a form because when you click this button, all the data in the form is sent to the Web server. When the user clicks the submit button, the data in the form is transferred to the URL specified in the `<form action>` tag. To add a submit button to your Web page, add the `<input type="submit">` tag to your HTML code. Table 2.10 lists a brief description of the attributes of the submit button:

Attribute	Description
name	Adds a name to the button.
value	Defines the text to be written on the button.
align	Defines the alignment of the button.
tabindex	Defines the tab order of the button, if there are multiple buttons in your Web page.

Let's create a Web page, named `AddingSubmitButton.html` to learn how to add a submit button to a form. You can find the `AddingSubmitButton.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.48 shows the code of the `AddingSubmitButton.html` page:

Listing 2.48: Adding a Submit Button

```

<html>
<head>
<title>
Adding a Submit Button
</title>
</head>
<body>
<form action="Example.html" method="post">
First Name:
<input type="text" name="Enter your name">
<br>
Last Name:
<input type="password" name="Password">
<br>
<input type="submit" value="SUBMIT">
</form>
</body>
</html>

```

When you open this page, the output appears, as shown in Figure 2.53:

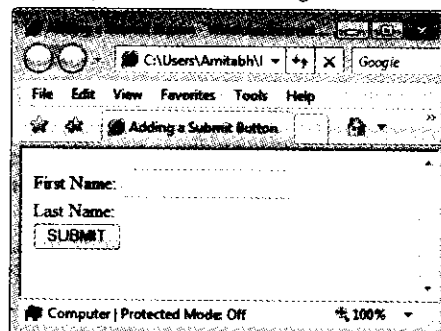


Figure 2.53: Submit Button in the HTML Form

In Figure 2.53, you can observe that a submit button is added to your Web page.

Adding a Reset Button

A reset button helps the user to clear all the data that they have entered in the text fields and start all over again. When the user clicks the reset button, all the controls in the form are returned to their original state and the values in the form fields are cleared. To add a reset button to your Web page, add the `<input type="reset">` tag to your HTML code.

Let's create a Web page, named AddingResetButton.html to learn how to add a reset button to a form. You can find the AddingResetButton.html file in the Code\HTML\Chapter 2 folder on the CD. Listing 2.49 shows the code of the AddingResetButton.html page:

Listing 2.49: Adding a Reset Button

```
<html>
<head>
<title>
Adding Reset Button
</title>
</head>
<body>
<form action="Example.html" method="post">
User Name:
<input type="text" size="30"/>
</br>
Password:
<input type="password" size="30"/>
</br>
<input type="submit" value="SUBMIT">
<input type="reset">
</form>
</body>
</html>
```

When you open this page, the output appears, as shown in Figure 2.54:

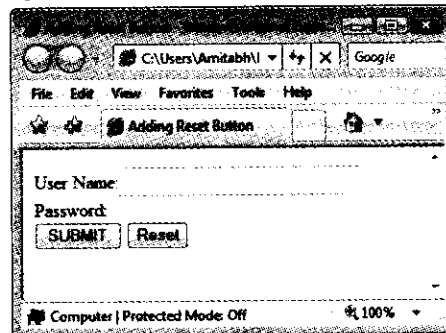


Figure 2.54: Reset Button in the HTML Form

Enter your name and password in the User Name and Password text fields, as shown in Figure 2.55:

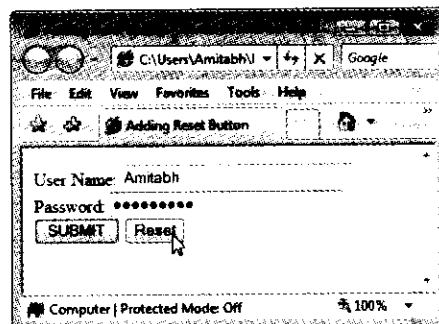


Figure 2.55: Resetting a Form

When you click the Reset button, the data entered in the text fields are cleared, as shown in Figure 2.56:

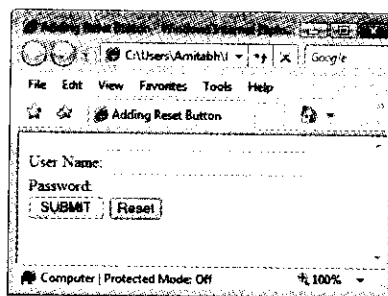


Figure 2.56: Form After Resetting

In Figure 2.56, you can observe the blank text fields after clicking the Reset button.

Adding a Text Area

A text area is a multi-line text input control and displays text entered in it. A user can write unlimited number of characters in the text area. You must set the numbers of rows and columns you want in the text area using the rows and cols attributes.

Let's create a Web page, named `AddingTextArea.html` to learn how to add a text area to a form. You can find the `AddingTextArea.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.50 shows the code of the `AddingTextArea.html` page:

Listing 2.50: Adding a Text Area

```

<html>
<head>
<title>
Adding Textarea
</title>
</head>
<body>
<form>
<textarea rows="10" cols="30">
This is an example of text area on the web page. A user can input text in this
area.
</textarea>
</form>
</body>
</html>
  
```

When you open this page, the output appears, as shown in Figure 2.57:

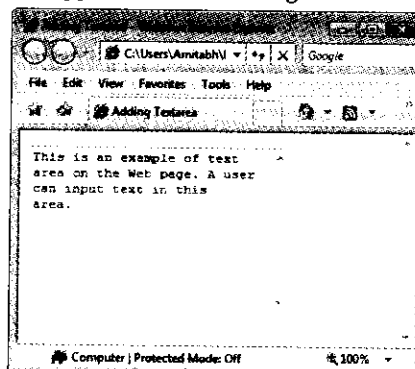


Figure 2.57: Text Area in the HTML Form

In Figure 2.57, you can observe that a text area is added to your Web page.

Adding a Selection Control

In HTML, you can add selection controls to your Web page. The selection control includes the `<select>`, `<option>`, and `<optgroup>` tags. The `<select>` tag defines the control for the selection of options and creates a drop-down list. You can use this tag in a form to accept a user input from a list of items. The `<option>` tag defines an option in a drop-down list. Its most important attribute is the `<value>` attribute that determines the value being sent to the server.

The `<optgroup>` tag allows you to group your choices in the form. This tag helps you to group related choices, when you have a long list of options to select from in the drop-down list created by using the `<select>` tag.

Let's create a Web page, named `SelectionControl.html` to learn how to add a selection control to a form. You can find the `SelectionControl.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.51 shows the code of the `SelectionControl.html` page:

Listing 2.51: Adding a Selection Control

```
<html>
<head>
<title>
Adding a Selection Control
</title>
</head>
<body>
<form>
<select>
<optgroup label="General hobby">
<option value ="playing">Playing</option>
<option value ="music">Music</option>
</optgroup>
<optgroup label="Educational hobby">
<option value ="stamp collection">Stamp Collection</option>
<option value ="reading">Reading</option>
</optgroup>
</select>
</form>
</body>
</html>
```

When you open this page, the output after selecting an option from the drop-down list appears, as shown in Figure 2.58:

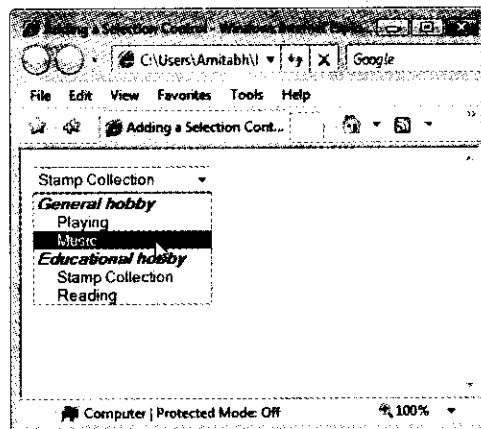


Figure 2.58: Selection Control in the HTML Form

In Figure 2.58, you can observe that a selection control is added to your Web page.

Adding a Multiple Selection Control

In the previous example, you can observe that the selection control allows you to select only a single option from the list. However, you can also select multiple options from the list at a time by using the same `<select>` tag. In this case, the `<select>` tag includes the `multiple` attribute to specify that multiple options can be selected at a time. To select multiple options from the list, you are required to hold down the CTRL key or the SHIFT key. Let's create a Web page, named `MultipleSelection.html` to learn how to add a multiple selection control to a form. You can find the `MultipleSelection.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.52 shows the code of the `MultipleSelection.html` page:

Listing 2.52: Adding a Multiple Selection Control

```
<html>
<head>
<title>
Adding a Selection Control
</title>
</head>
<body>
<form>
<select MULTIPLE>
<optgroup label="General hobby">
<option value="playing">Playing</option>
<option value="music">music</option>
</optgroup>
<optgroup label="Educational hobby">
<option value="stamp collection">Stamp Collection</option>
<option value="reading">reading</option>
</optgroup>
</select>
</form>
</body>
</html>
```

When you open this page, the output appears after selecting the options from the list, as shown in Figure 2.59:

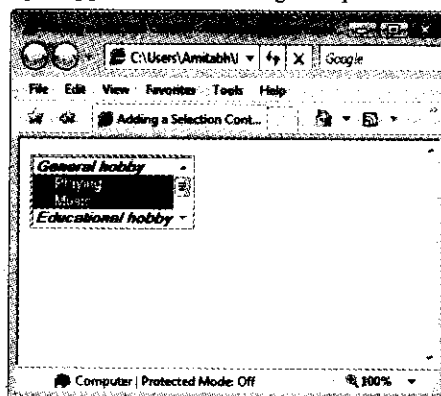


Figure 2.59: Multiple Selection Control in the HTML Form

In Figure 2.59, you can observe that a multiple selection control is added to your Web page. When you hold down the CTRL key and click the options in the list, the clicked options get selected.

Introducing Cascading Style Sheets

Cascading Style Sheets (CSS) or simply style sheets are text files that contain one or more rules in the form of property/value pairs to determine how elements in a Web page should be displayed. They were developed with the aim to create the structure, look and feel of a Web page but the elements present on the Web page are

handled separately. In addition, CSS deals with all the style-related aspects important to create a Web page. W3C has developed some specifications (or rules) to create and use style sheets. These specifications are called as the Cascading Style Sheet (CSS) specifications. The two versions of CSS specifications have been released till now: the first version is known as CSS1 and the second as CSS2. After the introduction of CSS, HTML elements that purely deal with style-related aspects, such as `<u>`, `<centre>`, and `<strike>` have been deprecated and W3C has recommended that in place of these HTML elements, their replacements should be used in CSS.

The syntax of CSS is slightly different from that of an HTML. In contrast to the angle brackets (`<` and `>`), equal signs, and quotation marks found in the HTML syntax, the CSS syntax contains curly braces, colons, and semicolons. The syntax of a CSS rule is as follows:

```
Selector {property1: property1-value; property2: property2-value; property3:
property3-value;}
```

In the preceding syntax, selector is the element that the rule defines, property1, property2, and property3 are the properties (attributes) defined for that element, and property1-value, property2-value, and property3-value are values assigned to these properties. The portion of the syntax enclosed within the curly braces is termed as declaration.

Using the preceding CSS rule syntax, we can create a CSS rule to set three background properties (background-color, background-image, and background-repeat) for the `<body>` element, as follows:

```
body {background-color: #0000ff; background-image: url(C:\Image1.jpg);
background-repeat: repeat-x}
```

Notice that the preceding CSS rule sets three background properties (background-color, background-image, and background-repeat) for the `<body>` element in three declarations. You can, however, set these three properties in one declaration, using the shorthand property named background. This shorthand property is a CSS property which enables you to set two or more properties in one declaration as follows:

```
body {background: #0000ff url(C:\Image1.jpg) repeat-x}
```

In the preceding CSS rule, you can note that a shorthand property specifies two or more properties by separating them with spaces.

You can create cascading styles in a Web page in four ways:

- Using inline styles
- Using external style sheets
- Using internal style sheets
- Using style classes

Let's now discuss these ways to create cascading styles, one by one.

Inline Styles

In the Inline Styles method, style for an HTML element is specified using its style attribute. Inline styles are useful when you want to define specific styles for individual elements present on a Web page.

Let's create a Web page, named `InlineStyles.html` to learn how to create inline styles in an HTML document. You can find the `InlineStyles.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.53 shows the code of the `InlineStyles.html` page:

Listing 2.53: Inline Style Example

```
<html>
<head>
<title>Inline styles</title>
</head>
<body>
<table border="1">
<caption>h2>Student Details</h2></caption>
<th style="background-color: #800000">Name</th>
<th style="background-color: #800000">Date of Birth</th>
```

```

<th style="background-color: #800000">Address</th>
<tr>
<td style="background-color: #808080">Sumit Saxena</td>
<td style="background-color: #808080">15-03- 1983</td>
<td style="background-color: #808080">Flat No.303, Shipra Suncity,
Ghaziabad</td>
</tr>
<tr>
<td style="background-color: #008000">Amitabh Kumar</td>
<td style="background-color: #008000">22-02- 1984</td>
<td style="background-color: #008000">H.No- 125, Kalkaji, Delhi</td>
</tr>
<tr>
<td style="background-color: #00ff00">Rohit Jandial</td>
<td style="background-color: #00ff00">05-07- 1983</td>
<td style="background-color: #00ff00">Flat No- 324, South Ext, Delhi</td>
</tr>
<tr>
<td style="background-color: #808000">Avantika Srivastava</td>
<td style="background-color: #808000">10-12- 1984</td>
<td style="background-color: #808000">H.No-541, Vikas Puri, Delhi</td>
</tr>
</table>
</body>
</html>

```

When you open this page, the output appears, as shown in Figure 2.60:

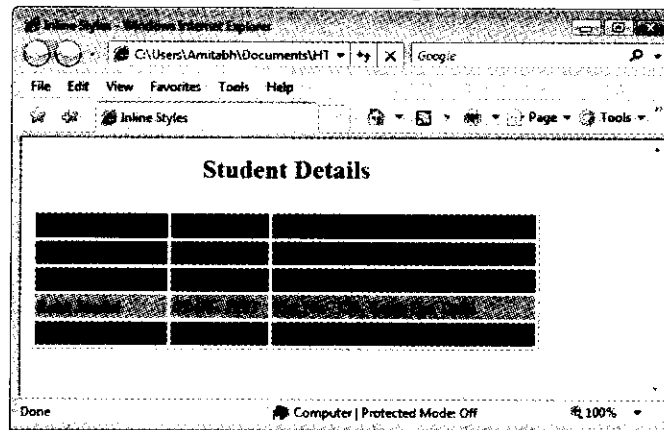


Figure 2.60: Applying CSS Using Inline Styles

You can notice in Figure 2.60, the background color of different rows in the table changes according to the style defined.

External Style Sheets

An external style sheet is a separate document that contains only CSS rules and has .css extension. External style sheets are used to apply uniform styles to all the Web pages. For example, let's assume that you are creating a website that contains more than one Web page and you want same look and feel for same types of HTML elements in all the Web pages. In such a situation, you can first create all the required CSS rules in an external style sheet and then link it to all the Web pages of the website.

You can link your Web page to an external style sheet by setting the href attribute of the <link> element to the name of the style sheet. The <link> element is added inside the <head> element of the HTML document.

You can create an external style sheet by creating a new, blank document in a text editor, such as Notepad, and create your styles in the document. After creating your styles, you need to save the document with .css extension. Let's create an external style sheet, named `Style.css`. You can find the `Style.css` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.54 shows the code of the `Style.css` style sheet:

Listing 2.54: Creating an External Style Sheet

```
body {background-color: #f0f8ff; font-family: Arial}
a:link {color: #808080}
a:visited {color: #0000ff}
a:hover {color: #00ff00}
a:active {color: #ff0000}
```

Linking a style to an HTML Document

After creating an external style sheet, you can use the styles added to the style sheet, in an HTML document. To do so, you need to link the style sheet to the HTML document (Web page) by using the `<link>` element.

Let's create a Web page, named `ExternalStyleSheets.html` to learn how to link the external style sheet to an HTML document. Listing 2.55 shows the code of the `ExternalStyleSheets.html` page:

Listing 2.55: External Style Sheet Example

```
<html>
<head>
<title>External Style Sheets</title>
</head>
<body>
<h1>External Style Sheet Example</h1>
<a href= Page1.html target="_blank">
<h2>Page 1</h2>
<a href= Page2.html target="_blank">
<h2>Page 2</h2>
</body>
</html>
```

When you open this page, the output appears, as shown in Figure 2.61:

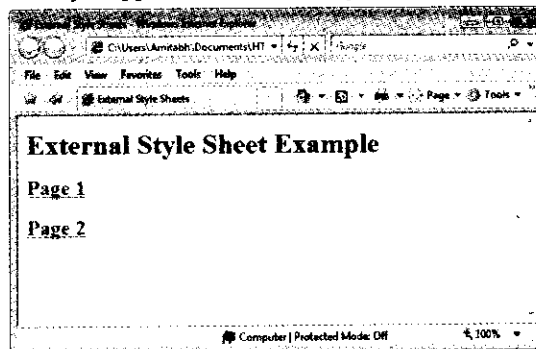


Figure 2.61: Page Without Style Sheet

Now, again open the `ExternalStyleSheets.html` page in Notepad and add the following highlighted code given in Listing 2.56 to the document:

Listing 2.56: Linking an HTML Document to an External Style Sheet

```
<html>
<head>
<title>External Style Sheets</title>
<link href= Style.css type= text/css rel= stylesheet>
</head>
<body>
```

```

<h1>External Style Sheet Example</h1>
<a href= Page1.html target="_blank">
<h2>Page 1</h2>
<a href= Page2.html target="_blank">
<h2>Page 2</h2>
</body>
</html>

```

You can find the LinkColors.html file in the Code\HTML\Chapter 2 folder on the CD. At this time, when you open the ExternalStyleSheets.html page, the output appears, as shown in Figure 2.62:

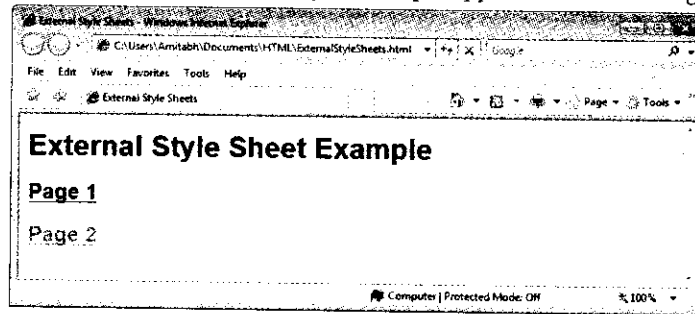


Figure 2.62: Applying CSS Using External Style Sheet

You can notice in Figure 2.62 how the background color of the Web page and the text color of the links present in it changes as you move your mouse pointer over the links.

Internal Style Sheets

Unlike external style sheets, internal style sheets are not separate documents rather they are styles, created inside an HTML document. In other words, an internal style sheet is a set of styles that is created as a part of an HTML document. These style sheets are useful when you want to apply similar styles to all the elements present on a Web page. Internal style sheets are created using the <style> element that is added inside the <head> element of the HTML document. Table 2.11 lists all the attributes of the <style> element:

Attribute	Description
dir	Gives the direction to directionally neutral text. You can set this attribute to ltr for left to right text direction or rtl for right to left text direction.
disabled	Specifies that the styles should not be applied initially. This attribute is a standalone attribute meaning that it is specified without assigning it a value.
lang	Indicates base language used for the element.
media	Sets the media for style sheet definitions (multiple destinations are specified by separating each pair of destinations by a comma). You can set this attribute to screen (default value for the attribute), print, projection, braille, speech, or all.
title	Allows the browser to build a menu of alternative style sheets. You can set this attribute to an alphanumeric value.
type	Specifies an essential attribute, which indicates the MIME type of the <style> element content. You can set this attribute to either text/css or text/javascript.

Let's create a Web page, named InternalStyleSheets.html to learn how the internal style sheet works in the HTML document. Listing 2.57 shows the code of the InternalStyleSheets.html page:

Listing 2.57: Creating Example for Internal Style Sheet

```

<html>
<head>
<title>Internal Style Sheets</title>
</head>
<body>
<h1>Internal Style Sheet Example</h1>
<a href= Page1.html target="_blank">
<h2>Page 1</h2>
<a href= Page2.html target="_blank">
<h2>Page 2</h2>
</body>
</html>

```

When you open this page, the output appears, as shown in Figure 2.63:

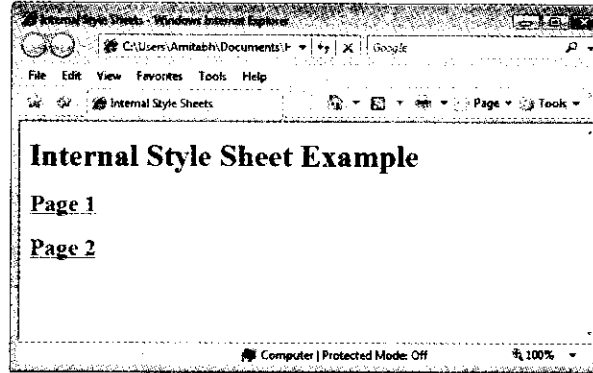


Figure 2.63: A Web Page Without Style Sheet

In Figure 2.63, you can observe that the styles are applied on a Web page using the internal style sheets.

Now, again open the `InternalStyleSheets.html` page in Notepad and add the following highlighted code given in Listing 2.58 to the document:

Listing 2.58: Creating an Internal Style Sheet

```

<html>
<head>
<title>Internal Style sheets</title>
<style>
h1 {color: red; text-align: center;}
h2 {color: blue; text-align: center;}
a {color: green; text-align: center;}
</style>
</head>
<body>
<h1>Internal Style Sheet Example</h1>
<a href= Page1.html target="_blank">
<h2>Page 1</h2>
<a href= Page2.html target="_blank">
<h2>Page 2</h2>
</body>
</html>

```

You can find the `InternalStyleSheets.html` file in the `Code\HTML\Chapter 2` folder on the CD. At this time, when you open the `InternalStyleSheets.html` page, the output appears, as shown in Figure 2.64:

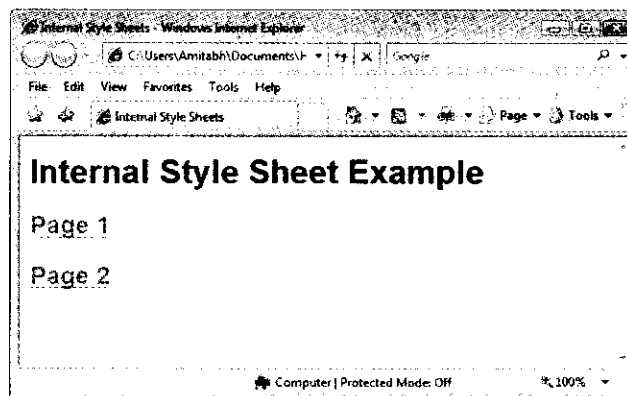


Figure 2.64: Applying CSS Using Internal Style Sheet

You can notice in Figure 2.64 that the styles you defined in the `<style>` element have been applied to your Web page.

Style Classes

In Style classes method, you can create styles in the form of style classes in external or embedded style sheets. To apply a style defined in a style class to an HTML element, you can assign the `class` attribute of the HTML element to the name of the style class. You can create two types of style classes: universal and element specific. A universal style class starts with a dot operator (.) followed by the class name. The syntax to define a universal style class is as follows:

```
<style>
  class name {class definition}
</style>
```

An element specific style class starts with the element name, followed by a dot operator, which is followed by the class name. The syntax to define an element specific style class is as follows:

```
<style>
  Element name.class name {class definition}
</style>
```

Let's create a Web page, named `StyleClasses.html` to learn how to apply styles in the HTML document by using the style classes. You can find the `StyleClasses.html` file in the `Code\HTML\Chapter 2` folder on the CD. Listing 2.59 shows the code of the `StyleClasses.html` page:

Listing 2.59: Applying the Style Classes

```
<html>
<head>
<title>Style Classes</title>
<style>
  body {background-color: #f0f8ff}
  th.color {background-color: #800000}
  .green {background-color: #008000}
</style>
</head>
<body>
<table border="1">
<caption><h2>Student Details</h2></caption>
<th class="color">Name</th>
<th class="color">Date of Birth</th>
<th class="color">Address</th>
<tr>
<td class="green">Sumit Saxena</td>
```