HTML and Web Pages

This chapter uses narrative, examples, and hands-on exercises to introduce programming concepts and Web development skills.

HyperText is a way to link and access information of various kinds as a web of nodes in which the user can browse at will.

Tim Berners-Lee

WorldWideWeb: Proposal for a HyperText Project, 1990

The goal of the Web was to be a shared information space through which people (and machines) could communicate. The intent was that this space should span from a private information system to a public information, from high value carefully checked and designed material, to off-the-cuff ideas which make sense only to a few people and may never be read again.

Tim Berners-Lee


If this were a traditional science, Berners-Lee would win a Nobel Prize. What he's done is that significant.

Eric Schmidt, CEO of Novell
Time Magazine, March 29, 1999

The World Wide Web is probably the most pervasive and visible aspect of computing today. Initially designed as a way for researchers to share documents over the Internet, the Web has evolved into a widespread, dynamic medium for communication and commerce. Using a program called a Web browser, people can instantly access documents (commonly referred to as Web pages) stored on computers around the world. With just a few clicks of a mouse, Web users can read newspapers, check sports scores, send email, and play interactive games. In addition, the Web provides an easy-to-use interface for advertising and business transactions, as evidenced by the abundance of corporate Web addresses that appear in television and print ads.

In this chapter, we introduce the basics of Web design and its underlying language, HTML. As you read, you will gain hands-on experience by developing your own Web page, adding new features incrementally as you learn. By the end of the chapter, you will be able to join the Web community by making your Web page publicly available. This means that anyone with a Web browser and an Internet connection will be able to access the document you’ve created.
HTML and Web Pages

Chapter Outline
- HTML Basics
- HTML Tags
- HTML Elements
- Document Formatting
- Text Spacing
- Headings and Alignment
- Font Formatting
- Hyperlink and Multimedia
- Hyperlinks
- Images
- List
- Formatting Lists
- Tables
- Table Borders
- Aligning Images and Text
- Making Pages Accessible
- Looking Ahead
- Chapter Summary
- Supplemental Material and Exercises

This chapter uses narrative, examples, and hands-on exercises to introduce programming concepts and Web development skills.

HyperText is a way to link and access information of various kinds as a web of nodes in which the user can browse at will.

Tim Berners-Lee
WorldWideWeb: Proposal for a HyperText Project, 1990

The goal of the Web was to be a shared information space through which people (and machines) could communicate. The intent was that this space should span from a private information system to a public information, from high value carefully checked and designed material, to off-the-cuff ideas which make sense only to a few people and may never be read again.

Tim Berners-Lee

If this were a traditional science, Berners-Lee would win a Nobel Prize. What he's done is that significant.

Eric Schmidt, CEO of Novell
Time Magazine, March 29, 1999

HTML Basics

A Web page is a text document that contains additional formatting information in a language called HTML (HyperText Markup Language). Using HTML, a Web page creator can identify various features of a page, such as a section heading, a centered table, or an image to be displayed. This information is then interpreted by a Web browser, which formats the page contents appropriately.

HTML Tags

HTML specifies formatting within a page using tags. In its simplest form, a tag is a word or symbol surrounded by angular brackets. For example, every Web page must begin with the tag `<html>` and end with the tag `</html>`. These tags specify that the enclosed text may contain HTML formatting information and that the browser should interpret the contents accordingly. Two required sections must appear inside every page's HTML tags: a HEAD section, which is delimited by the tags `<head>` and `</head>`, and a BODY section, which is delimited by the tags `<body>` and `</body>`.

The HEAD section contains information that the browser uses to determine the look of the page. For now, the only information that we will place in the HEAD will be the title of the page, which is specified by the tags `<title>` and `</title>`. When a browser renders the page, the text occurring between the `<title>` tags is displayed in the title bar at the top of the browser window. The BODY section contains the text that will actually appear in the page, usually separated into paragraphs using the tags `<p>` and `</p>`.

For example, Figure 1 shows a simple Web page named demo1.html. In this example, the title of the page is "Title of the Page", which appears in the title bar of the Web browser (Figure 2). The text in the body, which is displayed within the browser window, is simply "Test that appears in the page."

In this and subsequent figures, HTML tags are highlighted in blue to differentiate them from text that appears in the rendered page.

1. `<html>`
2. `<html>`
3. `<html>`
4. `<p> This is a simple Web page. </p>`
5. `<p>`
6. `<head>`
7. `<title> Title of the Page </title>`
8. `<head>`
9. `<body>`
10. `<p> Test that appears in the page. </p>`
11. `<body>`
12. `<body>`
13. `</html>`

Figure 1 A simple Web page.

Figure 2 demo1.html rendered in a Web browser.
HTML Elements

You may note that the tags you have seen so far—HTML, HEAD, BODY, P, and TITLE—all come in pairs. The first tag marks the beginning of a section; the second tag (which is identical to the first except for the slash) marks the end of the section. For this reason, we will often refer to the first and second tags as opening and closing tags, respectively. For example, <title> is the opening tag for the TITLE and </title> is the closing tag. HTML elements, which form the building blocks of Web pages, are made up of text and the tags that specify the text's role or purpose within the page. For example, <title>Title of the Page</title> can be referred to as a TITLE element, because it represents the piece of the Web page that specifies a title.

Although most HTML tags come in pairs, some do not. A comment element, or comment for short, is defined by a single tag, which begins with <!-- and ends with -->. For example, the Web page in Figure 1 contains three comment elements, occurring immediately after the HTML tag <lines 2 through 4>. You may think of comments as notes to yourself, or to some future Web developer who might review your HTML text. By placing information such as the file name, last modification date, and general purpose of the page in comments, you ensure proper credit and simplify the process of updating or modifying the page. As Figure 2 shows, comments are ignored by the browser, so this additional text does not clutter up the page when it is rendered. You should always place a sequence of comments at the top of any HTML document you create to identify the file name, author, and purpose of the page.

Numerous software tools on the market allow you to design and create elaborate Web pages. To create simple pages, however, all you really need is a text editor. A text editor is a program in which you can type and edit text, then save that text in a file. Common examples are NotePad and WordPad, which come free with Microsoft Windows, and SimpleText, which comes free on Apple computers. Many software manufacturers also offer free or low-cost text editors that provide special features, such as automatic indentation or color coding of HTML elements. Appendix A includes instructions on using some of these common editors.

To develop a Web page, all you need to do is create a text file containing the desired HTML elements. Then, a browser can be used to render the page. Of course, a Web page rarely looks exactly the way its designer intended the first time it is displayed by a browser. For example, you may make a mistake in entering the contents of the page, or you may simply decide that the page would look better with different formatting. Thus, Web page development usually involves having two windows open at the same time—one running a text editor, which allows the creator to update the page contents, and another running a Web browser, which shows the creator the latest version of the rendered page.

EXERCISE 1

Create a simple home page for yourself using some text editor. Your page should have a title, such as "Dave's Home Page," and should contain some basic information about you. Save your page to a file under the name index.html.

Document Formatting

You may have found that, when you displayed your home page, the text didn't appear in the browser window exactly as you typed it. This is inevitable, as the size of the browser window in which the page is rendered may vary. The lines you typed might be too long to fit in the window or might leave wasted white space at the end. To ensure that Web page text looks attractive in various window sizes, the browser automatically adjusts the length of lines to fit the current window. When typing text to the formatting, it ignores any blank lines, extra spaces, and tabs in your text. When displayed in the browser, any consecutive spaces, tabs, or blank lines will appear as a single space.

Text Spacing

Despite browsers' automatic formatting features, Web designers can control some text spacing by using additional HTML tags. A <P> element specifies text that is to be treated as a paragraph. Text enclosed within the opening tag <P> and the closing tag </P> will be displayed starting on a new line that is preceded by a blank line. If you wish to break text explicitly without inserting a blank line, you can place a <BR> element in the text. Unlike the paragraph element, which has opening and closing tags (because there is a logical beginning and end to the paragraph), a <BR> element is defined by a single tag:<BR/>. Text that follows a <BR/> tag will be displayed starting on a new line. By convention, tags such as<BR/> that do not have a corresponding closing tag end with/>. In addition, if you don't want to begin a new line but would like to force consecutive spaces in the text, you can do so by inserting the special symbol &nbsp; in the page. In general, a special symbol is a word in HTML that has special meaning to the browser when it renders the page. For example, the special symbol &nbsp; represents a nonbreaking space. When a Web page is rendered by a browser, each occurrence of the special symbol &nbsp; is displayed as a space, no matter where it appears in the text. Figure 3 demonstrates the use of tags and the &nbsp; symbol to control spacing (with the page rendering shown in Figure 4).

1. <html>
2. <head>
3. <title>Demo of text layout </title>
4. </head>
5. <body>
6. Here is a paragraph
7. that is broken across <br />
8. two lines.
9. </p>
10. <p>
11. Here is another paragraph. <br />
12. This time, the first line is indented two spaces.
13. </p>
14. </body>
15. </html>

Figure 3 A Web page that demonstrates text spacing and layout.

![A screenshot of the rendered page showing text spacing examples.](image)

Figure 4 demo2.html rendered in a Web browser.
HTML Elements
You may note that the tags you have seen so far—HTML, HEAD, BODY, P, and TITLE—all come in pairs. The first tag marks the beginning of a section; the second tag (which is identical to the first except for the slash '/') marks the end of the section. For this reason, we often refer to the first and second tags as opening and closing tags, respectively. For example, <title> is the opening tag for the TITLE and </title> is the closing tag. HTML elements, which form the building blocks of Web pages, are made up of text and the tags that specify the text's role or purpose within the page. For example, <title> of the Page <title> can be referred to as a TITLE element, because it represents the piece of the Web page that specifies a title.

Although most HTML tags come in pairs, some do not. A comment element, or comment for short, is defined by a single tag, which begins with <!-- and ends with -->. For example, the Web page in Figure 1 contains three comment elements, occurring immediately after the HTML tag (lines 2 through 4). You may think of comments as notes to yourself, or to some future Web developer who might review your HTML text. By placing information such as the file name, last modification date, and general purpose of the page in comments, you ensure proper credit and simplify the process of updating or modifying the page. As Figure 2 shows, comments are ignored by the browser, so this additional text does not clutter up the page when it is rendered. You should always place a sequence of comments at the top of any HTML document you create to identify the file name, author, and purpose of the page.

Numerous software tools on the market allow you to design and create elaborate Web pages. To create simple pages, however, all you really need is a text editor. A text editor is a program in which you can type and edit text, then save that text in a file. Common examples are Notepad and WordPad, which come free with Microsoft Windows, and SimpleText, which comes free on Apple computers. Many software manufacturers also offer free or low-cost text editors that provide special features, such as automatic indentation or color coding of HTML elements. Appendix A includes instructions on using some of these common editors.

To develop a Web page, all you need to do is create a text file containing the desired HTML elements. Then, a browser can be used to render the page. Of course, a Web page rarely looks exactly the way its designer intended the first time it is displayed by a browser. For example, you may make a mistake in entering the contents of the page, or you may simply decide that the page would look better with different formatting. Thus, Web page development usually involves having two windows open at the same time—one running a text editor, which allows the creator to update the page contents, and another running a Web browser, which shows the creator the latest version of the rendered page.

EXERCISE 1
Create a simple home page for yourself using some text editor. Your page should have a title, such as "Dave's Home Page," and should contain some basic information about you. Save your page to a file under the name index.html.

Document Formatting
You may have found that, when you displayed your home page, the text didn't appear in the browser window exactly as you typed it. This is inevitable, as the size of the browser window in which the page is rendered may vary. The lines you typed might be too long to fit in the window or might leave wasted white space at the end. To ensure that Web page text looks attractive in various window sizes, the browser automatically adjusts the length of lines to fit the current window, wrapping text to the next line as needed. You may also have noted that, when the browser performs this automatic formatting, it ignores any blank lines, extra spaces, and tabs in your text. When displayed in the browser, any consecutive spaces, tabs, or blank lines will appear as a single space.

Text Spacing
Despite browsers' automatic formatting features, Web designers can control some text spacing by using additional HTML tags. A P element specifies text that is to be treated as a paragraph. Text enclosed within the opening tag <p> and the closing tag </p> will be displayed starting on a new line that is preceded by a blank line. If you wish to break text explicitly without inserting a blank line, you can place a BR element in the text. Unlike the paragraph element, which has opening and closing tags (because there is a logical beginning and end to the paragraph), a BR element is defined by a single tag:<br/>. Text that follows a <br/> tag will be displayed starting on a new line. By convention, tags such as <br/> that do not have a corresponding closing tag end with >. In addition, if you don't want to begin a new line but would like to force consecutive spaces in the text, you can do so by inserting the special symbol &nbsp; in the page. In general, a special symbol is a word in HTML that has special meaning to the browser when it renders the page. For example, the special symbol &nbsp; represents a nonbreaking space. When a Web page is rendered by a browser, each occurrence of the special symbol &nbsp; is displayed as a space, no matter where it appears in the text. Figure 3 demonstrates the use of tags and the &nbsp; symbol to control spacing (with the page rendering shown in Figure 4).

```
<html>
  <head>
    <title> Demo of text layout </title>
  </head>
  <body>
    <p> Here is a paragraph </p>
    <br>    <br>
    <p> That is broken across <br> </p>
    <br>    <br>
    <p> &nbsp; &nbsp; &nbsp; Here is another paragraph. <br> </p>
    &nbsp; &nbsp; &nbsp; This time, the first line is indented two spaces.
  </body>
</html>
```

Figure 3 A Web page that demonstrates text spacing and layout.
Common errors to avoid...

It is important to note that HTML tags are sequences of characters that are placed in a file along with the text they are intended to format. If you enter a malformed tag, omitting either the opening or closing braces, the tag will not be recognized as HTML and will be displayed as plain text within the page. For example, the text

```html
<p>Hello, I'm Dave<br />
My hobbies include hiking and baseball.</p>
```

would be displayed as

```
Hello, I'm Dave /My hobbies include hiking and baseball.
```

By contrast, a browser will typically ignore a well-formed tag whose name is misspelled. For example, if you mistakenly type `<h1 />` instead of the intended `<h1>` tag, the tag will be ignored, and nothing will appear in its place in the page.

EXERCISE 2

Add additional paragraphs to your home page. If you are having trouble thinking of things to write about yourself, consider writing one paragraph that describes biological data, another that describes your hobbies, and a third listing some favorite books or movies. Use `<p>` or `<br />`, and `&nbsp;` appropriately to make your page readable and attractive.

Headings and Alignment

In a large document, it is often useful to divide the text into sections and then provide each with a heading, describing the content that follows. HTML includes special tags that enable Web designers to specify headings of various sizes. For example, text enclosed between the tags `<h1>` and `<h6>` is displayed in large, bold letters above the text that follows it. The tags `<h2>` and `<h3>` can be used to display a heading in slightly smaller letters, and the tags `<h4>` and `<h5>` are used for even smaller letters, and so on down to `<h6>` and `<h6>`. To divide different parts of a document further, designers can separate sections with a horizontal line using the `<hr />` tag (where `HR` stands for horizontal rule).

By default, all text in a Web page is displayed left-justified on the page. This means that each heading, paragraph, or line of text begins at the page's left edge. However, it is possible to align text so that it is centered or even right-justified by including a STYLE attribute in the appropriate opening tag. In general, an attribute is an additional property that can be assigned to an HTML element as part of its tag. The STYLE attribute can be added to an opening tag to specify the alignment of that particular element. For example, the opening tag

```html
<h2 style="text-align:center"> specs a heading that will be centered in the page, whereas
```

```html
<br style="text-align:right">
```

specifies a paragraph that will be right-justified in the page.

Figure 5 demonstrates the alignment of headings and paragraphs within a page (with the page rendering shown in Figure 6). This page also incorporates DIV tags, which can be used to group multiple elements into a single page division. The advantage of arranging multiple elements inside a set of DIV tags is that you can format all enclosed elements at once, as opposed to formatting each individually. In this example, a heading and a paragraph are grouped together and right-justified using DIV tags (lines 25 through 31).

Figure 5 A Web page that demonstrates the alignment of headings and text.

![Figure 5](image_url)

Figure 6 demo3.html rendered in a Web browser.
Common errors to avoid...

It is important to note that HTML tags are sequences of characters that are placed in a file along with the text they are intended to format. If you enter a malformed tag, omitting either the opening or closing braces, the tag will not be recognized as HTML and will be displayed as plain text within the page. For example, the text

```html
<p>Hello, I'm Dave<br/>
   My hobbies include hiking and baseball.</p>
```

would be displayed as

```
Hello, I'm Dave / My hobbies include hiking and baseball.
```

By contrast, a browser will typically ignore a well-formed tag whose name is misspelled. For example, if you mistakenly type `<b1>` instead of the intended `<br />`, the tag will be ignored, and nothing will appear in its place in the page.

**EXERCISE 2**

Add additional paragraphs to your home page. If you are having trouble thinking of things to write about yourself, consider writing one paragraph that provides biographical data, another that describes your hobbies, and a third listing some favorite books or movies. Use `<p>` tags, `<br />`, and `&nbsp;` appropriately to make your page readable and attractive.

**Headings and Alignment**

In a large document, it is often useful to divide the text into sections and then provide each with a heading describing the content that follows. HTML includes special tags that enable Web designers to specify headings of various sizes. For example, text enclosed between the tags `<h1>` and `<h6>` is displayed in large, bold letters above the text that follows it. The tags `<h2>` and `<h3>` can be used to display a heading in slightly smaller letters, `<h4>` and `<h5>` are used for even smaller letters, and so on down to `<h5>` and `<h6>`. To divide different parts of a document further, designers can separate sections with a horizontal line using the `<hr />` tag (where `HR` stands for horizontal rule).

By default, all text in a Web page is displayed left-justified on the page. This means that each heading, paragraph, or line of text begins at the page's left edge. However, it is possible to align text so that it is centered or even right-justified by including a `STYLE` attribute in the appropriate opening tag. In general, an attribute is an additional property that can be assigned to an HTML element as part of its tag. The `STYLE` attribute can be added to an opening tag to specify the alignment of that particular element. For example, the opening tag

```html
<h2 style="text-align:center">specifies a heading that will be centered in the page, whereas
<p style="text-align:right">specifies a paragraph that will be right-justified in the page.
```

Figure 5 demonstrates the alignment of headings and paragraphs within a page (with the page rendering shown in Figure 6). This page also incorporates DIV tags, which can be used to group multiple elements into a single page division. The advantage of arranging multiple elements inside a set of DIV tags is that you can format all enclosed elements at once, as opposed to formatting each individually. In this example, a heading and a paragraph are grouped together and right-justified using DIV tags (lines 25 through 31).
EXERCISE 3
Divide your home page into sections. Use a large heading (delimited by H1 tags) to list your name so that it is centered at the top of the page, followed by a horizontal line. For each subsequent text section, use a smaller heading, e.g., H2 or H3. At least one section of your page should contain multiple paragraphs.

Font Formatting
In printing terminology, a font defines a particular typeface (such as Times Roman, Courier, or Arial) and size (such as 11 pt. or 12 pt.) used to display characters in a document. In general, Web page creators have limited control over the fonts that are used to display their pages. Browsers allow the user to specify his or her own font preferences by selecting default fonts for viewing pages, so a browser may display a page very differently than another would. There is some font formatting that can be forced in a page, however. For example, text enclosed in the tags <b>...</b> will appear in bold, <i>...</i> indicate text in italics, <b><i>...</b><i>...</i> indicate text in a slightly larger font, and <small>...</small> indicate text in a slightly smaller font. In addition, you can change the color of text by enclosing it in SPAN tags of the following form:

```html
<SPAN style="COLOR : DESIRED_COLOR">...</SPAN>
```

SPAN tags serve to group a block of text together for common formatting. The STYLE attribute of the form `style="COLOR : DESIRED_COLOR"` specifies the color of the text within the SPAN tags, where DESIRED_COLOR is a color name such as red, blue, darkblue, or purple. Figure 7 demonstrates text formatting within the page (with the page rendering shown in Figure 8).

EXERCISE 4
Augment your home page by coloring some of the text and using font formatting where appropriate. For example, if you listed favorite books or movies, you could italicize their titles.

```html
1. <html>
2. <title>demo4.html</title>
3. <head>
4. <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
5. </head>
6. <body>
7. <p>Here is some <b>bold text</b>.</p>
8. <p>Here is some <i>italicized text</i>.</p>
9. <p>Here is some <b><i>bigger text</b><i> and <small>smaller text</small></small></p>
10. <p>Here is some <span style="COLOR : blue">blue text</span>.</p>
11. </body>
12. </html>
```

Figure 7 A Web page that demonstrates text formatting.

Hypertext and Multimedia
Although it is always possible to retrieve a Web page by entering its URL into the browser’s Address box, most pages are not accessed this way. The defining feature of the Web is its ability to link pages together through hyperlinks. A hyperlink is an element on a Web page, usually displayed as underlined text, that connects the page to another page or online resource. When a Web user clicks on a hyperlink, the browser loads the connected page, regardless of its physical location on the Web. Thus, by following a chain of hyperlinks, a user can explore related documents, even if the documents are stored on computers thousands of miles apart. Text that contains embedded hyperlinks is referred to as hypertext.

Common errors to avoid...

When specifying HTML elements with opening and closing tags, it is important to type both tags correctly. If you omit or misspell the opening tag of an element, the browser will ignore the closing tag, and no formatting will occur. On the other hand, if you specify the opening tag but misspell the closing tag, the indicated formatting will continue throughout the document. For example, inexperienced programmers often forget to include the </ in closing tags, which in effect specifies two opening tags. If you type <b><i>oops</i></b>, any text occurring after this tag set will appear in bold when rendered in a browser.

As previous examples have demonstrated, HTML elements can contain other elements nested inside them. For example, all HTML elements that format text in the page are nested inside the BODY element (</body>), which is itself nested inside </html>. When nesting HTML elements, be sure that the order in which you specify the closing tags is exactly opposite to that in which you specified the opening tags. Misordering tags in a nested element, such as <b><i>oops</i></b>, is technically incorrect and may cause the browser to display the text in unintended ways.

In programming terminology, any error related to the format of a statement (such as a misspelled tag name or unmatched opening and closing tags) is known as a syntax error. In general, the browser will ignore any HTML syntax error it encounters and continue rendering the page contents as best it can.
Hypertext and Multimedia

Although it is always possible to retrieve a Web page by entering its URL into the browser's Address box, most pages are not accessed this way. The defining feature of the Web is its ability to link pages together through hyperlinks. A hyperlink is an element on a Web page, usually displayed as underlined text, that connects the page to another page or online resource. When a Web user clicks on a hyperlink, the browser loads the connected page, regardless of its physical location on the Web. Thus, by following a chain of hyperlinks, a user can explore related documents, even if the documents are stored on computers thousands of miles apart. Text that contains embedded hyperlinks is referred to as hypertext.

Common errors to avoid...

When specifying HTML elements with opening and closing tags, it is important to type both tags correctly. If you omit or misspell the opening tag of an element, the browser will ignore the closing tag, and no formatting will occur. On the other hand, if you specify the opening tag but forget or misspell the closing tag, the indicated formatting will continue throughout the document. For example, inexperienced programmers often forget to include the / in closing tags, which in effect specifies two opening tags. If you type `<b>oops</b>`, any text occurring after this tag set will appear in bold when rendered in a browser.

As previous examples have demonstrated, HTML elements can contain other elements nested inside them. For example, all HTML elements that format text in the page are nested inside the BODY element `<body>...</body>`, which is itself nested inside `<html>...</html>`. When nesting HTML elements, be sure that the order in which you specify the closing tags is exactly opposite to that in which you specified the opening tags. Misordering tags in a nested element, such as `<b><i>oops</i></b>`, is technically incorrect and may cause the browser to display the text in unintended ways.

In programming terminology, any error related to the format of a statement (such as a misspelled tag name or unmatched opening and closing tags) is known as a syntax error. In general, the browser will ignore any HTML syntax error it encounters and continue rendering the page contents as best it can.
Hyperlinks

When inserting a hyperlink into a document, Web designers use the tags `<a>` and `</a>` to enclose the text that will represent the link. In this context, A is short for anchor, signifying that the hyperlink anchors (e.g., connects) separate documents together. The designer also must indicate the address of the page to which the link connects; this information is specified within the `<a>` tag via an attribute named HREF (short for HyperText Reference).

The general form of a hyperlink element is:

```html
<a href="ADDRESS_OF_PAGE">TEXT LABEL FOR LINK</a>
```

Whatever text you type between `<a>` and `</a>` tags will appear as an underlined link when the page is rendered. If the user clicks that text, the browser will replace the current page with the page specified in the opening tag. The address listed in the HREF attribute can be either an absolute address, meaning that it specifies the URL of a page on the Web, or a relative address, meaning that it specifies a file stored in a location related to that of the Web page itself. For example, the Web page in Figure 9 contains two hyperlinks (with the page rendering shown in Figure 10). The first (spanning lines 12 and 13) uses an absolute address, specifying the URL starting with http:// where that page can be found. When a visitor clicks the link labeled “Home Page for A Balanced Introduction to Computer Science”, the browser will access the linked page and display it in the browser window. The second hyperlink (on line 17) uses a relative address, specifying only the file name of the desired page. Because the http:// prefix is omitted, the browser automatically assumes that this address refers to a file stored in the same directory as the current Web page.

You might note that links within a Web page can appear in different colors. By default, a Web browser displays a link that has been previously visited in one color (usually purple), whereas an unvisited link appears in another color (usually blue).

**EXERCISE 5**

Add a section to your home page containing at least five hyperlinks to sites that you consider useful and/or interesting. To ensure that these hyperlinks work correctly, you should click each one and then view the linked page in your browser.

**Images**

So far, we have discussed ways to format text, font, and spacing within a Web page and to create links between your page and other Web resources. However, anyone who has surfed the Web realizes that pages contain much more than text and hyperlinks. The term multimedia refers to documents that integrate various communication media, such as text, images, movies, and sound clips. In HTML, images are loaded into a page using an `<img>` tag. (Note the `/` at the end of the tag, signifying that there is no corresponding closing tag.) Standard browsers have built-in capabilities for displaying images in GIF® (Graphics Interchange Format®) and JPEG (Joint Photographic Experts Group) formats, which are the most common methods of storing digital images. (The Graphics Interchange Format® is the Copyright property of CompuServe Incorporated. GIF® is a Service Mark property of CompuServe Incorporated.) To render other types of image formats, the browser may require special-purpose extensions called plug-ins. Therefore, if you want an image to be viewable by the majority of Web users, it is advisable to limit yourself to GIF and JPEG formats. An IMG tag typically has two attributes associated with it. The SRC attribute specifies the location of the image file to be displayed, and the ALT attribute specifies alternate text that appears if the image can't be found or displayed properly. The ALT attribute is particularly crucial when a Web page is rendered by certain types of browsers, such as a text-to-speech browser for the visually impaired. Also, some browsers display the ALT text when the mouse moves over an image, enabling Web designers to provide additional context for images.

The general form of an image element is:

```html
<img src="ADDRESS_OF_IMAGE" alt="TEXT_DESCRIPTION" />'
```

After encountering an image element when loading a Web page, the browser accesses the source file specified in the tag and displays that image in the page. As was the case with the HREF attribute for hyperlinks, the address assigned to the SRC attribute of an IMG can be either absolute (a complete URL) or relative (a file stored in a location related to that of the page). For example, the image element in Figure 11 (line 12) uses a relative address, specifying that the browser should display the image file named reed.jpg, that is stored in the same directory or folder as the Web page (the page rendering is shown in Figure 12).

**EXERCISE 6**

Look around the Web, and choose a public-domain image that you would like to include on your Web page. You must be careful not to violate copyrights or otherwise infringe upon the rights of the owner—unless explicitly stated otherwise, you should assume that any image on the Web is private property. Once located, copy the image to your local directory (under Windows, click the right mouse button over the desired image and select Save Image from the resulting menu; under the Mac OS, click on the image while holding down the Control key, then select Download Image to Disk). After saving the image in the directory that contains `index.html`, incorporate the local image into your home page. Be sure to include alternate text for your image via the ALT attribute.
Hyperlinks

When inserting a hyperlink into a document, Web designers use the tags `<a>` and `</a>` to enclose the text that will represent the link. In this context, `A` is short for anchor, signifying that the hyperlink anchors (e.g., connects) separate documents together. The designer also must indicate the address of the page to which the link connects; this information is specified within the `<a>` tag via an attribute named `HREF` (short for Hypertext Reference).

The general form of a hyperlink element is:

```html
<a href="ADDRESS_OF_PAGE">TEXT LABEL FOR LINK</a>
```

Whatever text you type between `<a>` and `</a>` tags will appear as an underlined link when the page is rendered. If the user clicks that text, the browser will replace the current page with the page specified in the opening tag. The address listed in the `HREF` attribute can be either an absolute address, meaning that it specifies the URL of a page on the Web, or a relative address, meaning that it specifies a file stored in a location related to that of the Web page itself. For example, the Web page in Figure 9 contains two hyperlinks (with the page rendering shown in Figure 10). The first (spanning lines 12 and 13) uses an absolute address, specifying the URL (starting with `http://`) where that page can be found. When a visitor clicks the link labeled "Home Page for A Balanced Introduction to Computer Science", the browser will access the linked page and display it in the browser window. The second hyperlink (on line 17) uses a relative address, specifying only the file name of the desired page. Because the `http://` prefix is omitted, the browser automatically assumes that this address refers to a file stored in the same directory as the current Web page.

You might note that links within a Web page can appear in different colors. By default, a Web browser displays a link that has been previously visited in one color (usually purple), whereas an unvisited link appears in another color (usually blue).

EXERCISE 5

Add a section to your home page containing at least five hyperlinks to sites that you consider useful and/or interesting. To ensure that these hyperlinks work correctly, you should click each one and then view the linked page in your browser.

Images

So far, we have discussed ways to format text, font, and spacing within a Web page and to create links between your page and other Web resources. However, anyone who has surfed the Web realizes that pages contain much more than text and hyperlinks. The term *multimedia* refers to documents that integrate various communication media, such as text, images, movies, and sound clips. In HTML, images are loaded into a page using an `<img>` tag. (Note the `/` at the end of the tag, signifying that there is no corresponding closing tag.) Standard browsers have built-in capabilities for displaying images in GIF (Graphics Interchange Format) and JPEG (Joint Photographic Experts Group) formats, which are the most common methods of storing digital images. (The Graphics Interchange Format is the Copyright property of Compuserve Incorporated, GIF is a Service Mark property of Compuserve Incorporated.) To render other types of image formats, the browser may require special-purpose extensions called plug-ins. Therefore, if you want an image to be viewable by the majority of Web users, it is advisable to limit yourself to GIF and JPEG formats.

An IMG tag typically has two attributes associated with it. The SRC attribute specifies the location of the image file to be displayed, and the ALT attribute specifies alternative text that appears if the image can't be found or displayed properly. The ALT attribute is particularly crucial when a Web page is rendered by certain types of browsers, such as a text-to-speech browser for the visually impaired. Also, some browsers display the ALT text when the mouse moves over an image, enabling Web designers to provide additional context for images.

The general form of an image element is:

```html
<img src="ADDRESS_OF_IMAGE" alt="TEXT_DESCRIPTION" />
```

After encountering an image element when loading a Web page, the browser accesses the source file specified in the tag and displays that image in the page. As was the case with the `HREF` attribute for hyperlinks, the address assigned to the SRC attribute of an IMG can be either absolute (a complete URL) or relative (a file stored in a location related to that of the page). For example, the image element in Figure 11 (line 12) uses a relative address, specifying that the browser should display the image file named reed.jpg that is stored in the same directory or folder as the Web page (the page rendering is shown in Figure 12).

EXERCISE 6

Look around the Web, and choose a public-domain image that you would like to include on your Web page. You must be careful not to violate copyrights or otherwise infringe upon the rights of the owner—unless explicitly stated otherwise, you should assume that any image on the Web is private property. Once located, copy the image to your local directory (under Windows, click the right mouse button over the desired image and select Save Image from the resulting menu; under the Mac OS, click on the image while holding down the Control key, then select Download Image to Disk). After saving the image in the directory that contains index.html, incorporate the local image into your home page. Be sure to include alternate text for your image via the ALT attribute.
Lists
When organizing text in a page, it is often convenient to list similar items in sequence. For example, we have mentioned the possibility of listing your hobbies or favorite movies in your home page. HTML provides two different elements for organizing lists. In an unordered list, individual items are preceded by bullets (solid circles), as shown on the left-hand side of Figure 13. In an ordered list, the items are numbered in sequence, as shown on the right-hand side of the figure.

In HTML, an unordered list is identified by the tags `<ul>` and `</ul>`, whereas an ordered list is identified by the tags `<ol>` and `</ol>`. In either type of list, the individual list items are delimited by the tags `<li>` and `</li>`. For example, the Web page in Figure 14 (rendered in Figure 15) contains list elements corresponding to the text lists in Figure 13.

<table>
<thead>
<tr>
<th>Unordered List</th>
<th>Ordered List</th>
</tr>
</thead>
<tbody>
<tr>
<td>old movies</td>
<td>1. And Then There Were None</td>
</tr>
<tr>
<td>Celtic music</td>
<td>2. Bringing Up Baby</td>
</tr>
<tr>
<td>baseball</td>
<td>3. Crouching Tiger, Hidden Dragon</td>
</tr>
<tr>
<td>hacking around the computer</td>
<td>4. Big Trouble in Little China</td>
</tr>
<tr>
<td></td>
<td>5. The Usual Suspects</td>
</tr>
</tbody>
</table>

Figure 13 Examples of unordered and ordered lists.

Formatting Lists
By default, items in an unordered list are preceded by bullets, and items in an ordered list are numbered. However, the page designer can specify alternative formats using the `STYLE` attribute. Items in an unordered list will be preceded by squares, instead of bullets, if you add the attribute `style="list-style-type:square"` to the opening `UL` tag. Likewise, you can specify that an ordered list use letters or roman numerals instead of numbers to identify each list item. The style attributes that correspond to each of these formatting styles are listed in Figure 16.
Lists

When organizing text in a page, it is often convenient to list similar items in sequence. For example, we have mentioned the possibility of listing your hobbies or favorite movies in your home page. HTML provides two different elements for organizing lists. In an unordered list, individual items are preceded by bullets (solid circles), as shown on the left-hand side of Figure 13. In an ordered list, the items are numbered in sequence, as shown on the right-hand side of the figure.

In HTML, an unordered list is identified by the tags `<ul>` and `</ul>`, whereas an ordered list is identified by the tags `<ol>` and `</ol>`. In either type of list, the individual list items are delimited by the tags `<li>` and `</li>`. For example, the Web page in Figure 14 (rendered in Figure 15) contains list elements corresponding to the text lists in Figure 13.

<table>
<thead>
<tr>
<th>Unordered List</th>
<th>Ordered List</th>
</tr>
</thead>
<tbody>
<tr>
<td>- old movies</td>
<td>1. And Then There Were None</td>
</tr>
<tr>
<td>- Celtic music</td>
<td>2. Bringing Up Baby</td>
</tr>
<tr>
<td>- baseball</td>
<td>3. Crouching Tiger, Hidden Dragon</td>
</tr>
<tr>
<td>- hacking around on the computer</td>
<td>4. Big Trouble in Little China</td>
</tr>
</tbody>
</table>

Formatting Lists

By default, items in an unordered list are preceded by bullets, and items in an ordered list are numbered. However, the page designer can specify alternative formats using the `STYLE` attribute. Items in an unordered list will be preceded by squares, instead of bullets, if you add the attribute `style="list-item-type:square"` to the opening `UL` tag. Likewise, you can specify that an ordered list use letters or roman numerals instead of numbers to identify each list item. The style attributes that correspond to each of these formatting styles are listed in Figure 16.
EXERCISE 7

Add a list to your home page. For example, you might list your favorite CDs, your current courses, or the names of your siblings.

1. <html>
2.   <head>
3.     <title>Demo of Lists</title>
4.   </head>
5. <body>
6.   <ol style="list-style-type:square">
7.     <li>Items preceded by squares.</li>
8.   </ol>
9.   <ol style="list-style-type:lower-alpha">
10.    <li>Items preceded by lowercase letters (a, b, c, d, ...).</li>
11.   </ol>
12.   <ol style="list-style-type:upper-alpha">
13.    <li>Items preceded by uppercase letters (A, B, C, D, ...).</li>
14.   </ol>
15.   <ol style="list-style-type:lower-roman">
16.    <li>Items preceded by lowercase roman letters (i, ii, iii, iv, ...).</li>
17.   </ol>
18.   <ol style="list-style-type:upper-roman">
19.    <li>Items preceded by uppercase roman letters (I, II, III, ...).</li>
20.   </ol>

Figure 16 Style attributes to format lists.

The ability to specify an ordered list's format is especially useful when you wish to nest lists. For example, the Web page in Figure 17 (rendered in Figure 18) displays an algorithm for finding the oldest person in a room. An algorithm is a sequence of step-by-step instructions for carrying out some task. In this case, the algorithm has four main steps, which appear in the page as items in an ordered list. The creator of this page has broken the third step into parts "a" and "b" by nesting an ordered list (with appropriate STYLE attribute) inside the third list item.

Figure 17 A Web page that demonstrates nested ordered lists.

Tables

Although it is usually preferable to let the browser determine how text is laid out within a page, there are times when you want things to line up just so. For example, suppose that you want to display a collection of related information about a student. Aligning such data into columns can make a page more attractive and easier to read.

Name: Chris
Age: 20
Hometown: Chicago
Major: Computer Science

In HTML, table elements are used to organize text and other elements into rows and columns. Elements in the same row appear on the same line when rendered by a browser (as is the case with "Name:" and "Chris" above). Elements in the same column are aligned on the left-hand side (as is the case with "Name:" and "Age:" above). "Hometown:" and "Major:" are aligned on the right-hand side (as is the case with "Chris" and "Computer Science"). Web designers specify a table element using the tags <table> and </table>. Individual rows in the table are identified by <tr> (short for table row), and the number of columns is determined by the number of data items in the rows, as identified by <td> (short for table data). Every time you end one row element and begin another, the browser places the first data item in the first column of that new row. For example, the Web page in Figure 19 (rendered in Figure 20) contains a table element that displays the aligned text from our sample student information record.

As this example demonstrates, the browser automatically adjusts the width of each column to accommodate its largest entry. In this case, the width of the first column is determined by the space needed to fit "Hometown." If the table included a fifth row containing a longer entry in the first column—say "Favorite Movie!"—then the browser would adjust the column size to fit that entry.

Table Borders

If you would like to insert borders between the rows and columns in a table, this is accomplished by assigning a number to the BORDERS attribute in the opening TABLE tag. For example, the opening tag <table border="1"> would begin a table in which a border separated each row and column. The number value assigned to the BORDERS attribute refers to the width of the border, as measured in pixels (or picture elements, the dots that make up the display on a computer screen). By increasing the number, the page designer can increase the width of the border. For example, the page in Figure 21 displays the same table as in Figure 19, except that each entry is surrounded by a border 1 pixel wide.
The ability to specify an ordered list's format is especially useful when you wish to nest lists. For example, the Web page in Figure 17 (rendered in Figure 18) displays an algorithm for finding the oldest person in a room. An algorithm is a sequence of step-by-step instructions for carrying out some task. In this case, the algorithm has four main steps, which appear in the page as items in an ordered list. The creator of this page has broken the third step into parts "a" and "b" by nesting an ordered list (with appropriate STYLE attributes) inside the third list item.

**EXERCISE 7**

Add a list to your home page. For example, you might list your favorite CDs, your current courses, or the names of your siblings.

```
1. <html>
2. <head>
3.  <title> Demo of Lists </title>
4. </head>
5. <body> 
6.   <p>To find the oldest person in a room:</p>
7.   <ol> 
8.     <li> Line up all the people along one wall. </li>
9.     <li> Ask the first person to state his or her name and birthday, then write this information down on a piece of paper. </li>
10.    <li> For each successive person in line: </li>
11.     <ol style="list-style-type:lower-alpha"> 
12.         <li> Ask the person his or her name and birthday. </li>
13.         <li> If the stated birthday is earlier than the date written on the paper, cross out the old information and write down the name and birthday of this person. </li>
14.     </ol> 
15.     <li> When you have reached the end of the line, the name and birthday of the oldest person will be written on the paper. </li>
16.   </ol> 
17. </body> 
18. </html>
```

Figure 17 A Web page that demonstrates nested ordered lists.

### Tables

Although it is usually preferable to let the browser determine how text is laid out within a page, there are times when you want things to line up just so. For example, suppose you want to display a collection of related information about a student. Aligning such data into columns can make a page more attractive and easier to read.

Table: Student Information

<table>
<thead>
<tr>
<th>Name</th>
<th>Chris</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>20</td>
</tr>
<tr>
<td>Hometown</td>
<td>Chicago</td>
</tr>
<tr>
<td>Major</td>
<td>Computer Science</td>
</tr>
</tbody>
</table>

In HTML, table elements are used to organize text and other elements into rows and columns. Elements in the same row appear on the same line when rendered by a browser (as is the case with "Name:" and "Chris" above). Elements in the same column are aligned on the left-hand side (as is the case with "Age:" and "20"). Web designers specify a table element using the tags `<table>` and `</table>`.

Individual rows in the table are identified by `<tr>` (short for table row), and the number of columns is determined by the number of data items in the rows, as identified by `<td>` (short for table data). Every time you end one row element and begin another, the browser places the first data item in the first column of that new row. For example, the Web page in Figure 19 (rendered in Figure 20) contains a table element that displays the aligned text from our sample student information record.

As this example demonstrates, the browser automatically adjusts the width of each column to accommodate its largest entry. In this case, the width of the first column is determined by the space needed to fit "Hometown": If the table included a fifth row containing a longer entry in the first column—say "Favorite Movie"—then the browser would adjust the column size to fit that entry.

### Table Borders

If you would like to insert borders between the rows and columns in a table, this is accomplished by assigning a number to the BORDER attribute in the opening `<TABLE>` tag. For example, the opening tag `<table border="1">` would begin a table in which a border separated each row and column. The number value assigned to the BORDER attribute refers to the width of the border, as measured in pixels (or picture elements, the dots that make up the display on a computer screen). By increasing the number, the page designer can increase the width of the border. For example, the page in Figure 21 displays the same table as in Figure 19, except that each entry is surrounded by a border 1 pixel wide.
Aligning Images and Text

In addition to organizing text into columns, tables are useful for aligning a variety of HTML elements. For example, a common use of tables is to align an image with text that refers to it. The Web page in Figure 23 (rendered in Figure 24) contains a table with only one row. An image appears...
HTML and Web Pages

1. `<html>`
2. `<file demo10.html>`
3. `<file This page demos a table for alignment. -->`
4. `<file ------------------------- -->`
5. `<file>`
6. `<head>`
7. `<title> Demo of tables </title>`
8. `</head>`
9. `<body>`
10. `<table border="1">`
11. `<tr><td> Name: </td> <td> Chris </td></tr>`
12. `<tr><td> Age: </td> <td> 20 </td></tr>`
13. `<tr><td> Hometown: </td> <td> Chicago </td></tr>`
14. `<tr><td> Major: </td> <td> Computer Science </td></tr>`
15. `</table>`
16. `</body>`
17. `</html>`

Figure 30 demo10.html rendered in a Web browser.

Aligning Images and Text

In addition to organizing text into columns, tables are useful for aligning a variety of HTML elements. For example, a common use of tables is to align an image with text that refers to it. The Web page in Figure 23 (rendered in Figure 24) contains a table with only one row. An image appears in the first column, and text appears in the second column. The result is that the text appears to the right of the image, centered vertically.

EXERCISE 8

Use a table to align text and/or images into rows and columns. For example, you might organize your class schedule into a table, designating a row for each class and columns for listing the course name, room number, and meeting times. Alternatively, you might align the names and email addresses of your friends into columns.
**Designer Secrets**

The Web has become an important information and communication source for millions of people worldwide. When designing a Web page, it is important to keep all users in mind, especially those who might have disabilities that limit their accessibility to online content. The World Wide Web Consortium has founded a Web Accessibility Initiative to encourage the design of Web pages that are accessible to all users, including vision-impaired users who may utilize devices to read Web content aloud. In 1999, the initiative produced the Web Content Accessibility Guidelines 1.0, a document that provides practical advice on creating accessible Web pages (accessible online at http://www.w3.org/TR/WCAG10). Guidelines from this document include:

1. Use headings, lists, and consistent structure to organize the page.
2. For each link, be sure to select text that makes sense when read out of context.
3. For each image, be sure to use the ALT attribute to describe its contents.
4. For each table, be sure that a line-by-line reading of the contents is sensible.

**Making Pages Accessible**

As you have learned in this chapter, developing Web pages does not require an Internet connection. You have been able to create pages, store them on your computer, and then view the pages in a browser. However, in order for your pages to be retrievable via the Web, you will need access to a Web server connected to the Internet. The easiest way to accomplish this will vary depending on your situation. If you are at a college or university, your school most likely has a Web server available for student use. After obtaining permission to store pages on such a server, you will receive specific instructions on how to copy your pages to the appropriate directories on that server. If your Internet access is supplied through an Internet Service Provider (ISP), such as AOL or MSN, the company will probably offer space on its Web servers for customers to use. You will need to contact the company's customer support in order to determine the steps for uploading your pages.

When you copy a page to a publicly accessible Web server, remember that you must also copy any supporting pages or files needed by the page. For example, in Exercise 6, you copied an image onto your disk and incorporated that image into your page. Because you employed a relative address to identify the image, a browser will assume that the image is stored in the same location as the page, so it is necessary to copy the image along with the page.

**EXERCISE 9**

If you have access to a Web server, copy both your home page and the downloaded image file to the server. After you have done this, anyone should be able to access your page, regardless of his or her location on the Internet.

**Common errors to avoid...**

Most Web pages refer to supporting documents, such as images or other Web pages, which are stored in the original page's directory or folder. Whenever you move the page from one directory or folder to another, you must be sure to move the supporting documents along with it. If the browser is unable to locate an image file in the expected location, the alternate text specified in the ALT attribute will be displayed (along with box containing an 'X' when using Internet Explorer). If a supporting HTML document (i.e., a linked Web page) cannot be found, then a default screen containing an error message will appear when a user clicks the link.

**Looking Ahead...**

In this chapter, you learned to create your own Web pages with HTML. By using (1) HTML elements to format and align text, (2) images and links to incorporate visuals and related elements, and (3) lists and tables to organize elements, you created a home page comparable to most found on the Web. In fact, the act of placing your home page on a publicly accessible Web server has effectively made your transition from Web user to Web contributor. Your home page now serves to introduce you to the Web, or at least to anyone in the world with Internet access.

As you have learned via the chapter exercises, creating a Web page involves both artistry (in organizing the layout of elements in the page) and careful, logical thinking (in properly formatting the HTML tags). In practice, most Web designers use a variety of technologies in addition to HTML when developing their pages. These technologies include XML (an extension of HTML in which the Web designer can define his own tags), Macromedia Flash (a program that integrates moving images into a page), and programming languages such as PHP, Java, and JavaScript. Although the mastery of all of these technologies is beyond the scope of this book, in subsequent chapters you will learn the fundamentals of JavaScript, the simplest and most popular language for extending HTML. Building upon the skills you have developed in this chapter, you will learn to use JavaScript in the development of dynamic Web pages—pages that not only display text and images, but also interact with the user and react to actions such as mouse clicks and keyboard entries.

**Chapter Summary**

- A Web page is a text document that contains additional formatting information in a language called HTML (HyperText Markup Language). A Web browser is a program that displays Web pages by interpreting the HTML and formatting the page accordingly. A Web server is a computer that runs special software for storing and retrieving pages. A Web address (formally known as a Uniform Resource Locator, or URL) specifies the location of a particular Web page.
- HTML specifies formatting within a page using tags. An HTML element, the building block of Web pages, is made up of text enclosed in tags that indicate the text's role or purpose within the page.
- Every HTML document must begin with the tag `<html>` and end with the tag `</html>`.
- An HTML document has two main sections, the HEAD and the BODY. The HEAD contains the TITLE of the page, which appears at the top of the browser window when that page is displayed. The BODY contains the text and formatting that you want to appear within the page.
- Comments can be placed at any point in the page and are delimited by <!--[ ] and -->. By placing key information about a page in comments, you ensure proper credit and simplify the process of updating or modifying the page. The browser ignores comments when the page is loaded.
- Typically, a browser formats text to fit the current window, ignoring blank lines and extra spacing within the HTML document. Designers can control spacing in Web pages by using
Designers not only have to be familiar with the basics of HTML and CSS, but also need to consider accessibility features such as ARIA roles and properties. ARIA (Accessible Rich Internet Applications) is a set of web design patterns and associated HTML attributes that can be used to improve the accessibility of complex user interfaces. For example, a header role can be used to identify the main content of a page, while a button role can be used to describe the action performed when the button is clicked. By using these ARIA labels, designers can make content more accessible to people using assistive technologies such as screen readers.

Looking Ahead...

In this chapter, you learned to create your own Web pages with HTML. By using (1) HTML elements to format and align text, (2) images and links to incorporate visuals and related documents, and (3) lists and tables to organize elements, you created a home page comparable to most found on the Web. In fact, the act of placing your home page on a publicly accessible Web server has affected your transition from Web user to Web contributor. Your home page now serves to introduce you to the world, or at least to anyone in the world with Internet access.

As you have learned via the chapter exercises, creating a Web page involves both artistry (in organizing the layout of elements in the page) and careful, logical thinking (in properly formatting the HTML tags). In practice, most Web designers use a variety of technologies in addition to HTML when developing their pages. These technologies include XML (an extension of HTML in which the Web designer can define his own tags), Macromedia Flash (a program that integrates moving images into a page), and programming languages such as PHP, Java, and JavaScript. Although the mystery of all of these technologies is beyond the scope of this book, in subsequent chapters you will learn the fundamentals of JavaScript, the simplest and most popular language for extending HTML. Building upon the skills you have developed in this chapter, you will learn to use JavaScript in the development of dynamic Web pages—pages that not only display text and images, but also interact with the user in a variety of ways, such as mouse clicks and keyboard entries.

Chapter Summary

- A Web page is a text document that contains additional formatting information in a language called HTML (HyperText Markup Language). A Web browser is a program that displays Web pages by interpreting the HTML and formatting the page accordingly. A Web server is a computer that runs special software for storing and retrieving pages. A Web address (formally known as a Uniform Resource Locator, or URL) specifies the location of a particular Web page.
- HTML specifies formatting within a page using tags. An HTML element, the building block of Web pages, is made up of text enclosed in tags that indicate the text’s role or purpose within the page.
- Every HTML document must begin with the tag <html> and end with the tag </html>.
- An HTML document has two main sections, the HEAD and the BODY. The HEAD contains the TITLE of the page, which appears at the top of the browser window when that page is displayed. The BODY contains the text and formatting that you want to appear within the page.
- Comments can be placed at any point in the page and are delimited by <!-- and -->. By placing key information about a page in comments, you ensure proper credit and simplify the process of updating or modifying the page. The browser ignores comments when the page is loaded.
- Typically, a browser formats text to fit the current window, ignoring blank lines and extra spacing within the HTML document. Designers can control spacing in Web pages by using
the `<br />` element, which explicitly breaks a line of text. Similarly, `<p>` and `</p>` tags specify a new paragraph (preceded by a blank line). In addition, the `&nbsp;` symbol inserts a space into text.

- The tags `<h1>`/`<h1>`, `<h2>`/`<h2>`, and `<h3>`/`<h3>` can be used to display section headings of decreasing sizes; `<hr />` draws a horizontal line across the page. Designers can add STYLE attributes to headings or paragraphs to control text alignment. For example, `<h2 style="text-align:center">` creates a centered heading.

- Numerous other HTML tags can be used to format text, such as `<b>`/`<b>` for bold text, `<i>`/`<i>` for italicized text, `<big>`/`<big>` for larger text, `<small>`/`<small>` for smaller text, and `<span style="color:red">`/`</span>` for red-colored text.

- Hyperlinks to other HTML documents can be embedded in a Web page using an anchor element, e.g., `<a href="http://www.prenhall.com">Prentice Hall</a>`. The HREF attribute specifies the address of the linked page.

- GIF and JPEG images can be embedded in a Web page using an image tag, e.g., `<img src="reed.gif" alt="Dave Reed" />`. The SRC attribute specifies the address of the image file to be displayed, and the ALT attribute specifies alternate text to be displayed if the image doesn't load.

- A list can be used to organize items in a page. By default, an unordered list (identified by `<ul>`/`</ul>`) displays each item (identified by `<li>`/`</li>`) preceded by a bullet. An ordered list (identified by `<ol>`/`</ol>`) displays each item (identified by `<li>`/`</li>`) preceded by a number.

- A table can be used to organize items into rows and columns. A table is identified by `<table>`/`</table>`, with individual rows identified by `<tr>`/`</tr>` and entries within that row identified by `<td>`/`</td>`.

### Supplemental Material and Exercises

#### HTML Standards
To ensure that all browsers are able to read and display Web content consistently, the World Wide Web Consortium maintains language standards that all browsers must abide by. The current standard for HTML, which defines the different tags and their meaning, is HTML 4.01. Thus, if you adhere to the HTML 4.01 standard when you write your Web page, you can be assured that any standards-compliant browser will display it correctly.

In 2002, the World Wide Web Consortium merged HTML with XML, a more general-purpose markup language for representing data. The merger of XML and HTML was named XHTML 1.0. HTML, as a defined by the HTML 4.01 standard, continues to be a part of the XHTML standard. However, to adhere to the broader XHTML 1.0 standard, additional XML content must be added to a Web page.

1. The following tag must be added at the top of the document.

   ```html
   <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
   ```

2. The opening HTML tag must have attributes added to it.

   ```html
   <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
   ```

3. The following tag must be added to the HEAD of the document:

   ```html
   <meta http-equiv="Content-Type" content="text/html;charset=utf-8" />
   ```
the `<br>` element, which explicitly breaks a line of text. Similarly, `<p>` and `</p>` tags specify a new paragraph (preceded by a blank line). In addition, the `&nbsp;` symbol inserts a space into text.

- The tags `<h1>`,`h2>`,`h3>`, and `<h4>` can be used to display section headings of decreasing sizes; `<br>` draws a horizontal line across the page. Designers can add `STYLE` attributes to headings or paragraphs to control text alignment. For example, `<h2 style="text-align:center">` creates a centered heading.

- Numerous other HTML tags can be used to format text, such as `<b>` for bold text, `<i>` for italicized text, `<p>` for larger text, `<small>` for smaller text, and `<span style="color:red">` for red-colored text.

- Hyperlinks to other HTML documents can be embedded in a Web page using an anchor element, e.g., `<a href="http://www.prenticehall.com">Prentice Hall</a>`. The `HREF` attribute specifies the address of the linked page.

- GIF and JPEG images can be embedded in a Web page using an image tag, e.g., `<img src="reed.gif" alt="Dave Reed" />`. The `SRC` attribute specifies the address of the image file to be displayed, and the `ALT` attribute specifies alternate text to be displayed if the image doesn't load.

- A list can be used to organize items in a page. By default, an unordered list (identified by `<ul>`) displays each item (identified by `<li>`) preceded by a bullet. An ordered list (identified by `<ol>`) displays each item (identified by `<li>`) preceded by a number.

- A table can be used to organize items into rows and columns. A table is identified by `<table>` and entries within that row identified by `<tr>` and entries within that row identified by `<td>`.

### Supplemental Material and Exercises

#### HTML Standards

To ensure that all browsers are able to read and display Web content consistently, the World Wide Web Consortium maintains a set of standards that all browsers must abide by. The current standard for HTML, which defines the different tags and their meaning, is HTML 4.01. Thus, if you adhere to the HTML 4.01 standard when you write your Web page, you can be assured that any standards-compliant browser will display it correctly.

In 2002, the World Wide Web Consortium merged HTML with XML, a more general-purpose markup language for representing data. The merger of XML and HTML was named XHTML 1.0, HTML, as defined by the HTML 4.01 standard, continues to be a part of the XHTML standard. However, to adhere to the broader XHTML 1.0 standard, additional XML content must be added to a Web page.

1. The following tag must be added at the top of the document.

```html
<!DOCTYPE html PUBLIC "+//W3C//DTD XHTML 1.0 Strict//EN"
 http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

2. The opening HTML tag must have attributes added to it.

```html
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
```

3. The following tag must be added to the HEAD of the document:

```html
<meta http-equiv="Content-Type" content="text/html;charset=utf-8" />
```

For developing simple Web pages, the additional generality of XHTML is not needed and so these complex tags can be omitted (as is done throughout this book). If XHTML 1.0-compliance is desired, however, the addition of these three XML tags to the HTML document suffices.

#### EXERCISE 10

The Web site for the World Wide Web Consortium includes a tool for verifying whether a Web page adheres to the XHTML 1.0 standard. To test your home page, first add the three tags listed above. Then, go to http://validator.w3.org, where you can validate your page either by specifying its URL or else by uploading the file itself. Make whatever revisions are necessary to successfully validate your page as XHTML 1.0 compliant.

#### Browser Settings

A fundamental idea behind HTML is that, although the creator of a page can use tags to identify various text elements, the browser handles the text's final formatting. This increases the universal accessibility of Web documents—if each specific browser controls a page's formatting, the page can be viewed in different browsers running on different type of machines and operating systems. In addition, this approach allows individual users to customize the way in which pages are displayed in their personal browsers. For example, users can easily increase or decrease the size of text fonts through the "View" menu.

Users can also customize other viewing preferences, such as selecting default fonts and colors and a default home page. If you are using Mozilla Firefox, select "Options" under the "Tools" menu to see all of the possible options. In Internet Explorer, you can change your preferences by selecting "Internet Options" under the "Tools" menu.

#### EXERCISE 11

If you are working on a private computer, customize the browser so that its settings match your personal preferences. If you are working on a publicly accessible computer, you may experiment with the settings, but be sure to reset them to their default values when you are done. It is inconsiderate to impose your preferences on others!

#### Background Color and Images

By default, most browsers display Web pages with a white background (although individual users can reset this default value via preferences). However, a little color can sometimes make a page more attractive and interesting. To change a page's background color, you must assign a specific color value within the STYLE attribute of the BODY tag. For example, the opening tag `<body style="background-color:lightgrey">` would cause the background of the page to appear light gray.

#### EXERCISE 12

Add a background color to your home page. Be careful in your choice of colors, though. Subtle colors, such as lightgray or lightblue, can be attractive, but bright or dark colors, such as yellow or darkblue, can make reading the page's text extremely difficult. Remember that readability is the primary goal when designing a Web page—if users can't read the contents, it doesn't matter how pretty the page is!

As you surf the Web, you may have noticed that some Web pages have backgrounds comprised of textured patterns or even photographic images. A Web designer can specify an image as the background of a Web page using the STYLE attribute of the BODY tag. For example, the opening tag `<body style="background-image: url(\"http://mp.netscape.com/assist/net_sites/bj/rock/yellow_rock.gif\")">`
would cause the image located at the specified URL to be displayed in the background of the page. In this case, the background would have a yellow, stone-like texture. Note that, if the image is too small to fill the entire browser window (which is usually the case), then the image is repeated to fill the space. This is why simple, repetitive patterns make the best backgrounds.

**EXERCISE 13**

If desired, add a background image to your page (sample images can be found at various Web sites, such as http://wp.netscape.com/assist/net_sites/bg/backgrounds.html and http://www.free-backgrounds.com/). However, be very selective when choosing your image. Plastering your picture in the background of your page might seem like a neat idea, but complex background images almost always detract from the page’s readability.

**Images as Links**

In all the examples you have seen so far, links have been labeled with text—that is, only text appeared between the opening and closing A tags. However, Web designers can anchor a link to any HTML element, including an image. The following example defines an image link, in which an IMG element is placed inside the A tags.

```html
<a href="http://dave-reed.com/book/cover.jpg"
   alt="A Balanced Introduction to Computer Science">
   <img src="http://dave-reed.com/book/cover.jpg"
        alt="A Balanced Introduction to Computer Science" />
</a>
```

Images that serve as hyperlinks display the same behavior that hypertext does. When a user clicks the image, the browser loads the page specified in the HREF attribute. By default, images enclosed in A tags are surrounded by colored borders to identify the images as links. However, if you do not want the image to appear with a border, you can add attributes to the ANCHOR and IMG tags to remove the border, as in the following:

```html
   alt="A Balanced Introduction to Computer Science" border="0">
   <img src="http://dave-reed.com/book/cover.jpg"
        alt="A Balanced Introduction to Computer Science" />
</a>
```

**EXERCISE 14**

Add images that serve as links to your home page. The images that you select should have relevance to the link destinations—don’t use a picture unless a user could intuitively connect the image to the document represented by the link. Also, do not remove the borders from images unless you have a good reason for doing so. These borders help Web users identify and locate links.

**Cascading Style Sheets**

As we have seen, several of the tags shown in this chapter have an optional STYLE attribute that can be used to control the way the page element is displayed. For example, adding the attribute style="text-align:center" to an H1 tag causes that header to be centered on the page. Likewise, adding the attribute style="color:red" to a P or DIV tag causes the text in that paragraph or page division to appear in red.

When developing a Web site with numerous interconnected pages, consistency is important. Having some of your pages with centered headers and light blue background, while other pages have left-justified headers and a white background, produces a disconnected, unprofessional appearance. Fortunately, HTML provides a mechanism for applying consistent STYLE attributes across multiple

```css
h1 {text-align:center}
body {background-color:blue; color:white}
```

If this text were saved in a file named mystyle.css, it could then be linked into any HTML document by placing the following tag in the HEAD of the page:

```html
<link href="mystyle.css" rel="stylesheet" type="text/css" />
```

As a result, every page that linked this cascading style sheet would apply the STYLE attributes consistently. That is, all pages linking to mystyle.css would have a blue background, white text, and centered H1 elements. Plus, any subsequent changes made to the cascading style sheet, say changing the background color to dark blue, would automatically be applied to all pages that are linked to it.

**EXERCISE 15**

Create a cascading style sheet that defines STYLE attributes for your home page. Add a LINK element to your page to link to that cascading style sheet and produce the desired appearance. Next, link that same cascading style sheet to another HTML document and verify that it maintains a consistent look to the page.
would cause the image located at the specified URL to be displayed in the background of the page. In this case, the background would have a yellow, stone-like texture. Note that, if the image is too small to fill the entire browser window (which is usually the case), then the image is repeated to fill the space. This is why simple, repetitive patterns make the best backgrounds.

**EXERCISE 13**

If desired, add a background image to your page (sample images can be found at various Web sites, such as http://ap.netscape.com/assist/net_sites/bg/backgrounds.html and http://www.free-backgrounds.com/). However, be very selective when choosing your image. Placing your picture in the background of your page might seem like a neat idea, but complex background images almost always detract from the page's readability.

**Images as Links**

In all the examples you have seen so far, links have been labeled with text—that is, only text appeared between the opening and closing A tags. However, Web designers can anchor a link to any HTML element, including an image. The following example defines an image link, in which an IMG element is placed inside the A tag:

```
</a>
```

Images that serve as hyperlinks display the same behavior that hypertext does. When a user clicks the image, the browser loads the page specified in the HREF attribute. By default, images enclosed in A tags are surrounded by colored borders to identify the images as links. However, if you do not want the image to appear with a border, you can add attributes to the ANCHOR and IMG tags to remove the border, as in the following:

```
  <img src="http://dave-reed.com/book/cover.jpg" alt="A Balanced Introduction to Computer Science" border="0" />
</a>
```

**EXERCISE 14**

Add images that serve as links to your home page. The images that you select should have relevance to the link destinations—don't use a picture unless a user could intuitively connect the image to the document represented by the link. Also, do not remove the borders from images unless you have a good reason for doing so. These borders help Web users identify and locate links.

**Cascading Style Sheets**

As we have seen, several of the tags shown in this chapter have an optional STYLE attribute that can be used to control the way the page element is displayed. For example, adding the attribute style="text-align:center" to a H1 tag causes that header to be centered on the page. Likewise, adding the attribute style="color:red" to a P or DIV tag causes the text in that paragraph or page division to appear in red.

When developing a Web site with numerous interconnected pages, consistency is important. Having some of your pages with centered headers and light blue background, while other pages have left-justified headers and a white background, produces a disconnected, unprofessional appearance. Fortunately, HTML provides a mechanism for applying consistent STYLE attributes across multiple pages.