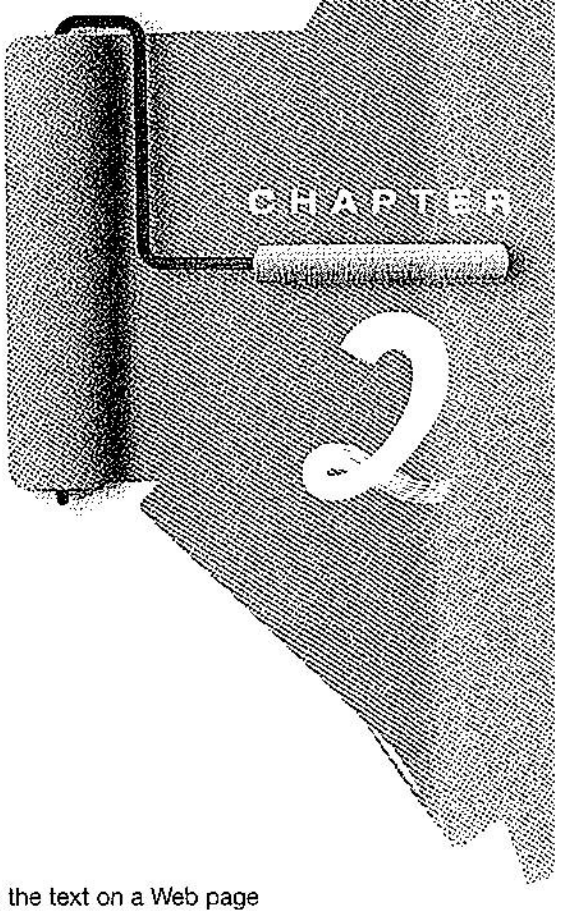


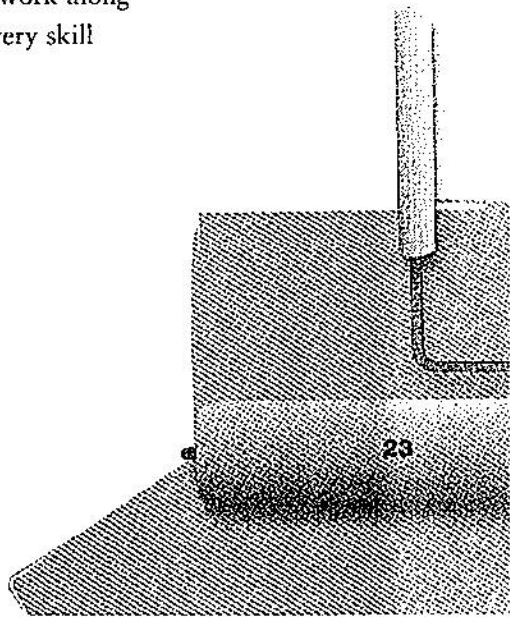
XHTML Basics

Chapter Objectives

In this chapter, you will learn about ...

- The development of HTML
- The transition from HTML to XHTML
- XHTML syntax, elements, and document type definitions
- The anatomy of a Web page
- Formatting the body of a Web page
- Formatting the text on a Web page
- Physical and logical style elements
- Special characters
- Using the anchor element to link from page to page
- Creating absolute, relative, and e-mail links

This chapter introduces Hypertext Markup Language (HTML), the language used to create Web pages, and eXtensible Hypertext Markup Language (XHTML), the latest version of HTML. The chapter begins with an introduction to the syntax of XHTML, continues with the anatomy of a Web page, and introduces block-level and inline formatting and demonstrates hyperlinks as sample pages are created. You will learn more if you work along with the sample pages in the text. Coding XHTML is a skill and every skill improves with practice.



2.1 What Is HTML?

The World Wide Web is composed of files containing Hypertext Markup Language (HTML) and other markup languages that describe Web pages. HTML was developed using Standard Generalized Markup Language (SGML). SGML prescribes a standard format for embedding descriptive markup within a document and for describing the structure of a document. SGML is not in itself a document language, but rather a description of how to specify one and create a **document type definition (DTD)**.

The W3C, http://w3c.org, sets the standards for HTML and its related languages. HTML (like the Web itself) is in a constant state of change.

HTML is the set of markup symbols or codes placed in a file intended for display on a Web browser page. These markup symbols and codes identify structural elements such as paragraphs, headings, and lists. HTML can also be used to place media (such as graphics, video, and audio) on a Web page and describe fill-in forms. The browser interprets the markup code and renders the page. HTML permits the platform-independent display of information across a network. That is, no matter what type of computer a Web page was created on, any browser running on any operating system can display the page.

Each individual markup code is referred to as an **element** or **tag**. Each tag has a purpose. Tags are enclosed in angle brackets, the < and > symbols. Most tags come in pairs: an opening tag and a closing tag. These tags act as containers and are sometimes referred to as container tags. For example, the text that is between the `<title>` and `</title>` tags on a Web page would display in the title bar on the browser window.

Some tags are used alone and are not part of a pair. For example, a tag that displays a horizontal line on a Web page, `<hr />`, is a stand-alone or self-contained tag and does not have a closing tag. You will become familiar with these as you use them. Most tags can be modified with **attributes** that further describe their purpose.

2.2 Why XHTML and Not HTML?

The newest version of HTML used today is actually **eXtensible HyperText Markup Language (XHTML)**. XHTML uses the tags and attributes of HTML along with the syntax of XML. While many Web pages and Web authoring tools still use HTML, as a Web developer you must learn about XHTML because you will be seeing a lot of it in the future.

Why was a new version needed? HTML was originally developed to provide access to electronic documents via a Web browser. Web browsers that evolved along with HTML were written to forgive coding errors, ignore syntax errors, and allow "sloppy" HTML code. Web browsers contain many program instructions that are designed to ignore mistakes such as missing ending tags and to guess how the developer meant the page to display. This is not a problem for a personal computer, which has relatively large processing power. However, this could be an issue for electronic devices with fewer resources, such as a personal digital assistant (PDA) or mobile phone.

Also, as new versions of Web browsers were developed and competed for market share, they often created their own proprietary extensions to HTML—tags that were not part of the standard and supported by one browser only. This created a lot of nonstandard HTML pages, and browsers are coded to accept this and ignore tags they don't recognize. However, this extra processing is not efficient, especially for devices with limited resources.

Finally, HTML is a structural language—it was originally intended to mark up printed documents for online viewing. It describes the structure of the document instead of the contents or information contained in the document. The Web has changed from a medium used to display electronic versions of paper documents to a medium that provides diverse information for a variety of devices. HTML does not fit this need. How will a table 600 pixels wide be displayed on a mobile phone? With the expansion of the Web to include devices other than personal computers, the need for a descriptive rather than structural language became evident and XHTML was created.

The purpose of XHTML was to provide a foundation for device-independent Web access. XHTML was developed by the W3C to be the reformulation of HTML as an application of XML. Tim Berners-Lee, the W3C director and inventor of the Web, stated in a press release (http://www.w3.org/2000/01/xhtml-pressrelease), "XHTML 1.0 connects the present Web to the future Web. It provides the bridge to page and site authors for entering the structured data, XML world, while still being able to maintain operability with user agents that support HTML 4." XHTML combines the formatting strengths of HTML and the data structure and extensibility strengths of XML. Since XHTML was designed using XML, let's take a quick look at XML.

XML (eXtensible Markup Language) is the W3C standard method for creating new markup languages that will support the display of nontraditional content such as mathematical notation, as well as support newer display devices such as PDAs and mobile phones. XML can fulfill these diverse needs because it is an extensible language—it is designed to allow the definition of new tags or markup. The syntax of XML is very exacting because the portable devices will not have to waste processing power guessing how the document should display, but will be able to display information efficiently. XHTML, which combines the language of HTML with the syntax of XML, is a markup language that should adapt to future needs.

An XML document must be well formed. A **well-formed document** is a document that adheres to the syntax rules of the language. The XHTML examples in the text will guide you in creating well-formed Web pages using XHTML. As a starting point, it is recommended that XML documents begin with an XML declaration. The basic form of this declaration is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
```

This XML declaration indicates that the document is based on the XML 1.0 standard. It also indicates the character encoding (the internal representation of letters, numbers, and symbols) in this document is UTF-8, a form of Unicode. This XML declaration will be the first line in each Web page that you write. See Appendix D, Comparison of HTML and XHTML, for a list of the key syntax rules of XML.

2.3 Document Type Definition

Because multiple versions and types of HTML and XHTML exist, the W3C recommends identifying the type of markup language used in a Web page document. Recall from Chapter 1 that XHTML 1.1 is the most recent version of HTML. However, XHTML 1.1 is not yet well supported by popular browsers. For this reason, we will follow the W3C XHTML 1.0 Recommendation. The three types of XHTML 1.0, **XHTML 1.0 Transitional**, **XHTML 1.0 Strict**, and **XHTML 1.0 Frameset**, are defined in Table 2.1.

Table 2.1 XHTML document types

Document Type Definition	Description
XHTML 1.0 Transitional	The least strict specification for XHTML 1.0; allows the use of Cascading Style Sheets and traditional formatting instructions such as fonts; used for most of the coding in this book
XHTML 1.0 Strict	Requires the exclusive use of Cascading Style Sheets; not used in this book
XHTML 1.0 Frameset	Required for pages using XHTML framesets; not used in this book

The version and type of XHTML is listed in the Document Type Definition (DTD) tag (commonly called the DOCTYPE). The DTD identifies the version and type of XHTML contained in your document. Browsers and HTML code validators can use the information in the DTD when processing the Web page. The DTD tag is placed at the top of a Web page document, even before the `<html>` tag. The DTD for XHTML 1.0 Transitional is as follows:

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

You will place the DTD as the second line in each Web page document you create. Are you ready to create your first Web page?

2.4 Your First Web Page

After the XML declaration and the DTD, each Web page begins with an opening `<html>` tag and ends with a closing `</html>` tag. These tags indicate that the text between them is HTML formatted. It tells the browser how to interpret the document.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html> an opening tag
... page information goes here
</html> a closing tag
```

There are two sections on a Web page: the **head** and the **body**. The head section, sometimes called the **header**, contains information that describes the Web page document.

Tags that are located in the head section include the title of the Web page, keywords that can be used by search engines, and references to scripts and styles. Many of these do not show directly on the Web page. The head section begins with the **<head>** tag and ends with the </head> tag.

The body section contains text and elements that do show directly on the Web page. The purpose of the body section is to describe the contents of the Web page. You will spend most of your time coding XHTML in the body of a Web page. If you type text in the body section, it will appear directly on the page.

The body section begins with the **<body>** tag and ends with the </body> tag.

The following code sample describes the anatomy of a Web page: a header section followed by a body section.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html>
<head>
... header information goes here
</head>
<body>
... body information goes here
</body>
</html>
```

Notice that the XHTML tags are lowercase. This conforms to XML syntax. Notice also that the DTD statement does not follow this syntax. The DTD statement indicates the markup language being used and has its own formatting—mixed case.

In XHTML, the <html> tag also needs to describe the XML namespace (**xmlns**), which is the location of the documentation for the elements being used. This additional information is added to the <html> tag in the form of an attribute. The `xmlns` attribute points to the URL of the XHTML namespace used in the document, the standard http://www.w3.org/1999/xhtml. The optional `lang` and `xml:lang` attributes (described in Appendix A, XHTML Reference) specify the spoken language of the document. Search engines and screen readers may access these attributes.

What are Web page editors?

No special software is needed to create an XHTML document—all you need is a text editor. Notepad is a text editor that is included with Microsoft Windows. TextEdit is distributed with the Mac OS X operating system. BBEdit is another popular editing program for Mac users. An alternative to using a simple text editor or word processor is to use a commercial Web authoring tool, such as Microsoft Expression Web or Adobe Dreamweaver. There are also many free or shareware editors available, including PageBreeze and Emacs. Netscape Composer is a Web page editor built into the Netscape Navigator browser. Regardless of the tool you use, having a solid foundation in XHTML will be useful. The examples in this text use Notepad.

The final version of the basic anatomy of a Web page follows. Note that the first five lines will usually be the same on every Web page that you create.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
... header information goes here
</head>
<body>
... body information goes here
</body>
</html>
```

HANDS-ON PRACTICE 2.1

Launch Notepad or another text editor and type in the following XHTML:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>My First Web Page</title>
</head>
<body>
Hello World
</body>
</html>
```

Notice that the first lines in the file contain the XML declaration and the DTD. The XHTML code begins with an opening <html> tag and ends with a closing </html> tag. The purpose of these tags is to indicate that the content between the tags makes up a Web page. The head section is delimited by <head> and </head> tags and happens to contain a pair of title tags with the words "My First Web Page" in between. The body section is delimited by <body> and </body> tags. The words "Hello World" are typed on a line between the body tags. See Figure 2.1 for a screenshot of the code as it would appear in Notepad. You have just created the source code for a Web document.

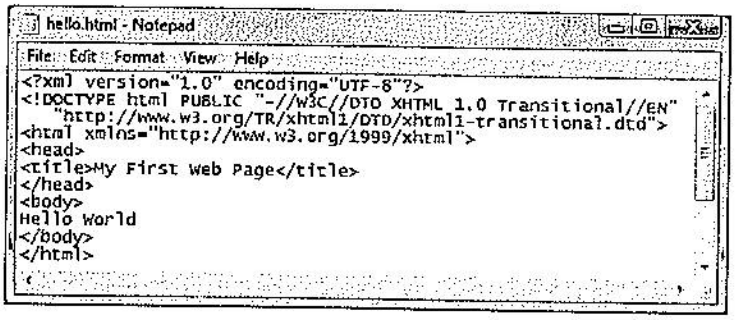


Figure 2.1
Source code of hello.html

FAQ

Do I have to start each tag on its own line?

No. A browser can display a page even if all the tags follow each other on one line with no spaces. Humans, however, find it easier to write and read XHTML if line breaks and indentation (more on this later) are used.

Save Your File

You will save your file with the name of hello.html. Web pages use either an .htm or .html file extension. Select File from the menu bar, and then select Save As. The Save As dialog box appears. Using Figure 2.2 as an example, type the file name.

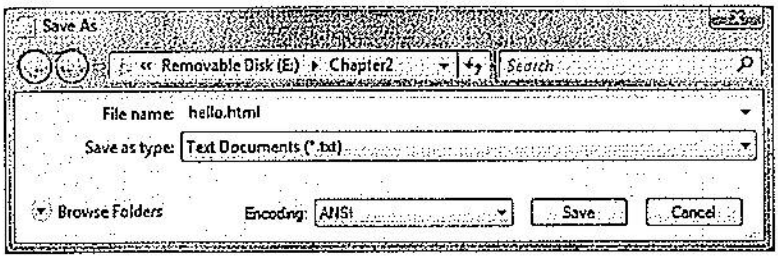


Figure 2.2
The Save As dialog box

FAQ

Why does my file have a .txt file extension?

In some older versions of Windows, Notepad will automatically append a .txt file extension. If this happens, type the name of the file within quotes; "hello.html", and save your file again.

Click the Save button after you type the file name. Sample solutions for the Hands-On Practice exercises are available in the student files. If you would like, compare your work with the solution (Chapter2/hello.html) before you test your page.

Test Your Page

There are two ways to test your page as follows:

1. **Launch Windows Explorer.** Navigate to your hello.html file. Double-click hello.html. The default browser will launch and will display your hello.html page. Your page should look similar to the one shown in Figure 2.3.

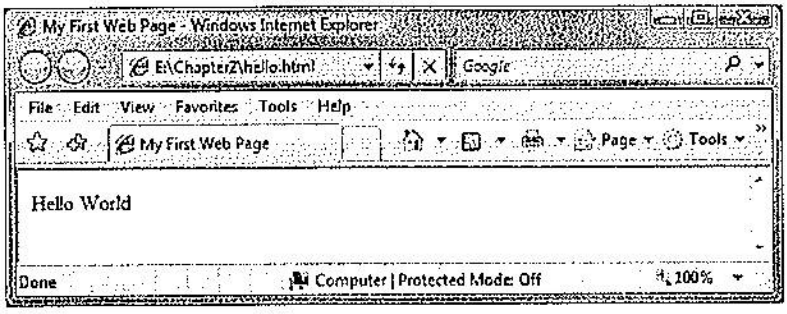


Figure 2.3
Web page displayed by Internet Explorer

2. **Launch Internet Explorer.** (If you are using Internet Explorer 7, select Tools, Menu Bar.) Select File, Open, Browse, My Computer, and then select your drive. Double-click hello.html and click OK. If you used Internet Explorer, your page should look similar to the one shown in Figure 2.3. A display of the page using Firefox is shown in Figure 2.4.

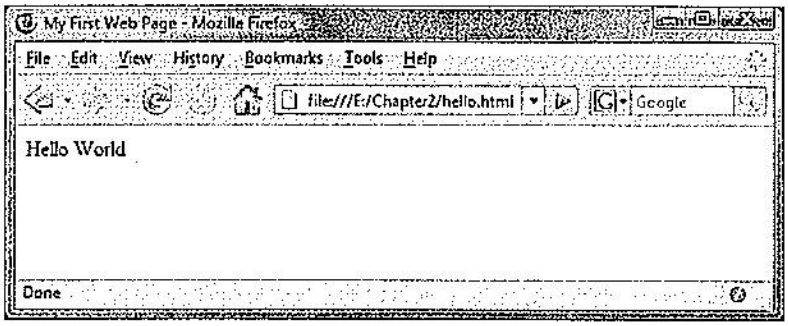
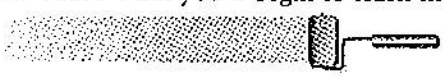
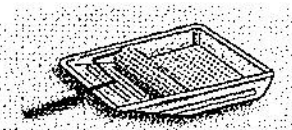


Figure 2.4
Web page displayed by Firefox

Examine your page. Look carefully at the browser window. What do you think is the purpose of the `<title>` tag? If you guessed that it's to display the title in the browser window, you are correct! Some search engines need the text surrounded by the `<title>` and `</title>` tags to help determine relevancy of keyword searches, so make certain that your pages contain descriptive titles. The `<title>` tag is also used when viewers bookmark your page or add it to their Favorites. An engaging and descriptive page title may entice a visitor to revisit your page. If your Web page is for a company or an organization, it's a good idea to include the name of the company or organization in the title.

You might be thinking "Hmmm ... white background, black text, no images, can't we make the page look more interesting?" Sure we can. That's what you'll begin to learn in the next section.





CHECKPOINT 2.1

1. Describe the origin, purpose, and features of HTML.
2. Explain why you would use XHTML instead of HTML.
3. Describe the purpose of the header and body sections of a Web page.

2.5 XHTML—Body and Text Basics

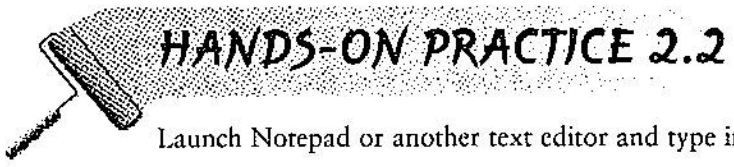
Have you noticed the wide variety of page designs on Web sites? Whether a Web page contains mostly text, uses blocks of color, displays images, employs animation, or is interactive, the foundation of the page is the `<body>` tag.

The Body Element

The purpose of the <body> element is to contain the text and XHTML elements that will display in the browser window. As you noticed when you created your first Web page, any text that you type in the body section of a Web page document will be displayed by the browser in the actual Web page. Often, this text is organized by structural elements that indicate important headings, text paragraphs, and lists. These structural elements are **block-level elements**—they control blocks of text such as headings, paragraphs, and lists. Tags that affect individual sections of text are called **inline-level elements**. Web development is a skill—the more you practice, the better you get. Why not try each example as you read?

The Heading Element

Headings are block-level elements that are organized into levels h1 through h6. The size of the text is largest for **<h1>** and smallest for **<h6>**. Depending on the font being used (more on fonts in Chapter 3), text contained in <h5> and <h6> tags may be displayed smaller than the default text size.



Launch Notepad or another text editor and type in the following XHTML:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Sample Heading Tags</title>
</head>
<body>
  <h1>Heading Level 1</h1>
  <h2>Heading Level 2</h2>
  <h3>Heading Level 3</h3>
  <h4>Heading Level 4</h4>
  <h5>Heading Level 5</h5>
  <h6>Heading Level 6</h6>
</body>
</html>
```

Save the file as heading.html. Launch a browser such as Internet Explorer or Firefox to test your page. It should look similar to the page shown in Figure 2.5. You can compare your work with the solution found in the student files (Chapter2/heading.html).

Notice that each heading in Figure 2.5 is on its own line and that there is a blank line between headings. The heading tag is a container tag. Notice how there are always corresponding opening <h#> and closing </h#> tags. It's a good idea to use headings to emphasize important topics or sections on a Web page.

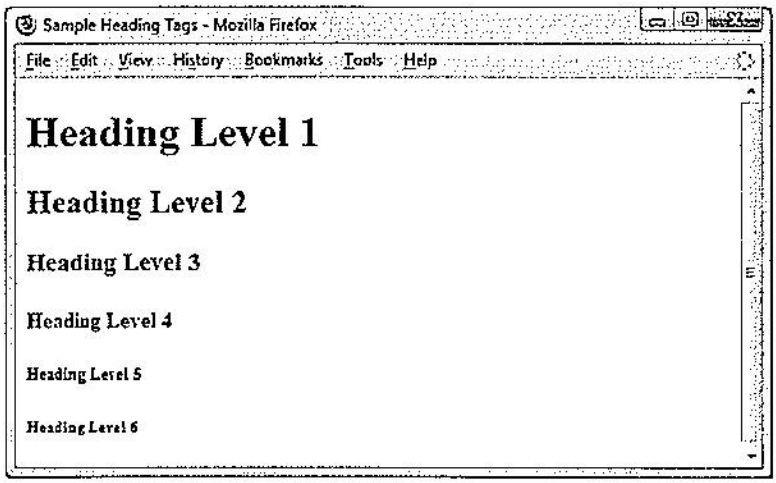


Figure 2.5
Sample heading.html

Accessibility and Headings

Focus on Accessibility

Heading tags can help to make your pages more accessible and usable. To indicate areas within a page hierarchically, code heading tags numerically as appropriate (h1, h2, h3, and so on) and include page content in block-level elements such as paragraphs and lists. Visually challenged visitors who are using a screen reader can configure the software to display a list of the headings used on a page to focus on the topics that interest them. Your well-organized page will be more usable for every visitor to your site, including those who are visually challenged.

The Paragraph Element

Paragraph elements are block-level elements used to group sentences and sections of text together. Text that is contained by and tags will have a blank line above and below it.

HANDS-ON PRACTICE 2.3

Open your heading.html file in a text editor. Use the following sample code and add a paragraph of text to your page below the line with the <h1> tags and above the line with the <h2> tags. Save your page as heading2.html.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Sample Heading Tags</title>
</head>
<body>
```

```
  <h1>Heading Level 1</h1>
  <p>This is a sample paragraph about HTML and XHTML. XHTML is the
newest version of HTML. XHTML uses the tags and attributes of HTML
along with the syntax of XML.</p>
  <h2>Heading Level 2</h2>
  <h3>Heading Level 3</h3>
  <h4>Heading Level 4</h4>
  <h5>Heading Level 5</h5>
  <h6>Heading Level 6</h6>
</body>
</html>
```

Launch a browser to test your page. It should look similar to the page shown in Figure 2.6 and to the solution in the student files (Chapter2/heading2.html).

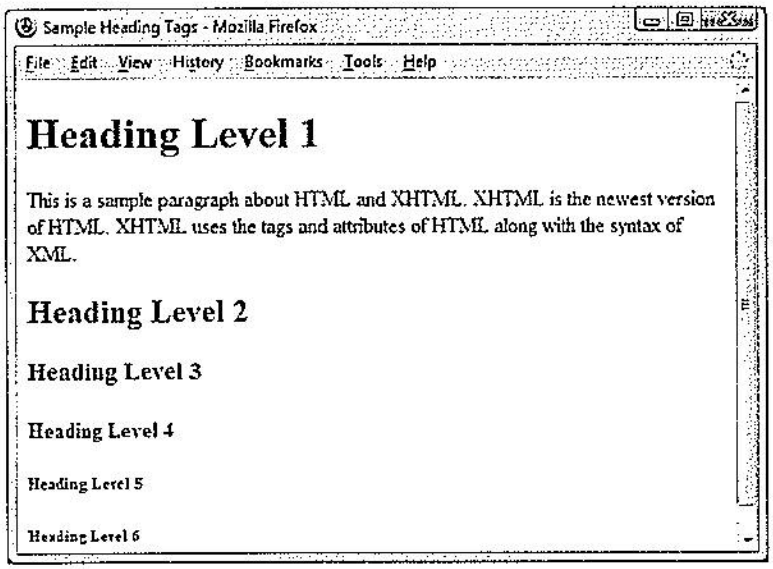
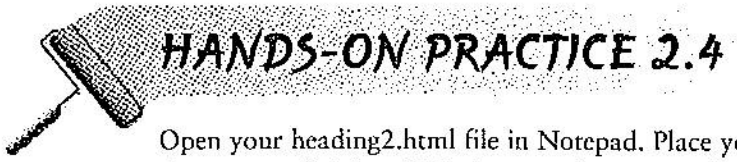


Figure 2.6
Web page using headings and a paragraph

Notice how the text wraps automatically as you resize your browser window. If you wanted to have the second sentence in the paragraph begin on its own line, you would need to add a line break. The next section describes `<br />`, the tag used to create line breaks.

The Line Break Element

The line break tag, `<br />`, is used to force a new line when the text on the Web page document is displayed by a browser. The line break tag is used alone—it is not used as a pair of opening and closing tags. It is considered to be a stand-alone or self-contained tag. If you were using HTML syntax, the line break tag would be coded as `<br>`. Because you are using XHTML (which follows XML syntax), the line break tag is coded as `<br />` (the ending `/>` indicates a self-contained tag).



Open your heading2.html file in Notepad. Place your cursor after the first sentence in the paragraph (after "This is a sample paragraph about HTML and XHTML."). Press the Enter key. Save your page. Test your page in a browser and notice that even though your source code showed the "This is a sample paragraph about HTML and XHTML." sentence on its own line, the browser did not render it that way. A `<br />` tag is needed to configure the browser. Open the heading2.html file in Notepad and add a `<br />` tag after the first sentence in the paragraph. Save your page as heading3.html. Your source code should look similar to the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Sample Heading Tags</title>
</head>
<body>
  <h1>Heading Level 1</h1>
  <p>This is a sample paragraph about HTML and XHTML.<br /> XHTML is
the newest version of HTML. XHTML uses the tags and attributes of
HTML along with the syntax of XML.</p>
  <h2>Heading Level 2</h2>
  <h3>Heading Level 3</h3>
  <h4>Heading Level 4</h4>
  <h5>Heading Level 5</h5>
  <h6>Heading Level 6</h6>
</body>
</html>
```

Launch a browser to test your page. It should look similar to the page shown in Figure 2.7. You can compare your work with the solution found in the student files (Chapter2/heading3.html).

As you tested your Web pages, you may have noticed that the headings and text begin near the left margin. This is called left alignment and is the default alignment for Web pages. There are times when you want a paragraph or heading to be centered or right-aligned (justified). The `align` attribute can be used for this. The purpose of an **attribute** is to modify the properties of an XHTML element. In this case, the align attribute modifies the element's horizontal alignment (left, center, or right) on a Web page. To center an element on a Web page use the attribute `align="center"`. To right-justify an element on a Web page, use `align="right"`. The default alignment is left. The `align` attribute can be used with a number of block level elements, including the paragraph (`<p>`) and heading (`<h1>` through `<h6>`) tags.

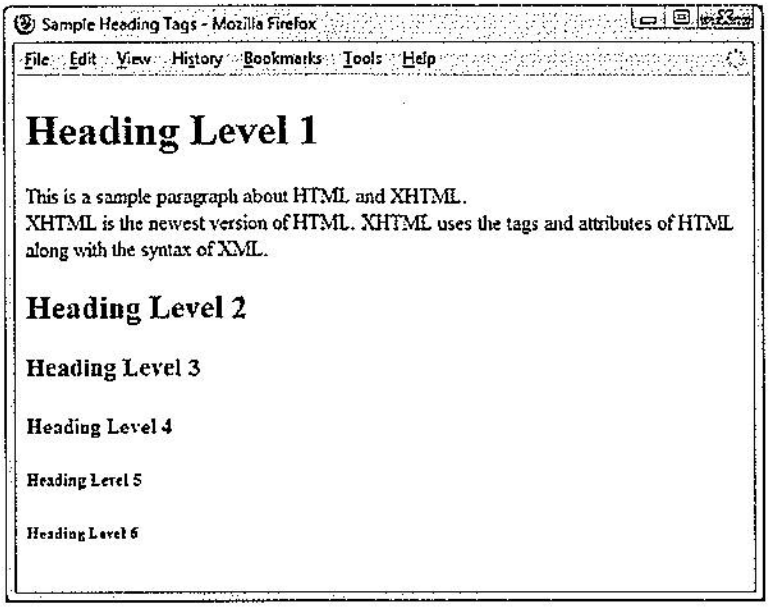


Figure 2.7
A `<br />` tag creates the line break after the first sentence

FAQ

Why does my Web page still look the same?

Often, students make changes to a Web page but get frustrated because their browser shows an older version of the page. The following troubleshooting tips are helpful when you know you modified your Web page but the changes do not show up in the browser:

1. Make sure you save your page after you make the changes.
2. Verify the location that you are saving your page to—the hard drive, a particular folder.
3. Verify the location that your browser is requesting the page from—the hard drive, a particular folder.
4. Be sure to click the Refresh or Reload button in your browser.

Open your heading3.html file in Notepad. Modify the heading to be centered. Change the `<h1>` tag to `<h1 align="center">` but do not change the closing `</h1>` tag. Also modify the paragraph to be centered on the Web page. Change the `<p>` tag to `<p align="center">`, but do not change the closing `</p>` tag. Save your page as heading4.html and test it in a browser. Your page should look similar to the page shown in Figure 2.8. You can compare your work with the solution found in the student files (Chapter2/heading4.html).

Legacy Alert. You will find many Web pages that use the `align` attributes to center block-level elements such as paragraphs and headings. Be aware that the `align` attribute is supported in XHTML 1.0 Transitional but is deprecated. In Chapters 3 and 6, you will learn to use Cascading Style Sheets (CSS) to configure alignment of text on a Web page.

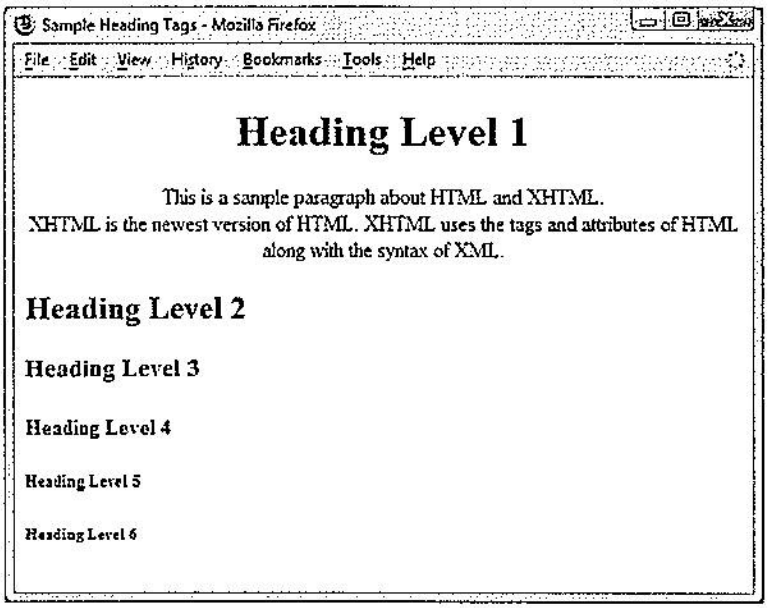


Figure 2.8
Using the `align` attribute to center the first heading and paragraph

The Blockquote Element

Besides organizing text in paragraphs and lists, sometimes it is useful to indent a block of text for special emphasis. Items often found indented in this manner include quotations, lists, and instructions. A block of indented text begins with a **<blockquote>** tag and ends with a </blockquote> tag.

HANDS-ON PRACTICE 2.5

Launch Notepad or another text editor and type in the following XHTML:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Blockquote Example</title>
</head>
<body>
  <h1>Markup Languages</h1>
    <blockquote>
      HTML HyperText Markup Language<br />
      DHTML Dynamic HyperText Markup Language<br />
      XHTML eXtensible HyperText Markup Language<br />
      XML eXtensible Markup Language
    </blockquote>
```

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

Save your file as blockquote.html. Launch a browser and test your file. Your page should look similar to the page shown in Figure 2.9 and the solution in the student files (Chapter2/blockquote.html). Notice how the text that was entered between `<blockquote>` tags is indented.

Figure 2.9
Sample blockquote.html

2.6 XHTML—List Basics

Lists are used on Web pages to organize information. When writing for the Web, remember that headings and bulleted lists make your pages clear and easy to read. XHTML can be used to create three types of lists: **definition lists**, **ordered lists**, and **unordered lists**.

Definition Lists

Definition lists help to organize terms and their definitions. The terms stand out and their definitions can be as long as needed to convey your message.

Definition lists are also handy for organizing Frequently Asked Questions (FAQs) and their answers. The questions and answers are offset with indentation. Each defined term begins on its own line at the margin. Each definition begins on its own line and is indented. See Figure 2.10 for an example of a Web page that uses a definition list.

Any type of information that consists of a number of corresponding terms and longer descriptions is well suited to being organized in a definition list.

Definition lists begin with the `<dl>` tag and end with the `</dl>` tag. Each defined term in the list begins with the `<dt>` tag and ends with the `</dt>` tag. Each term definition (data definition) begins with the `<dd>` tag and ends with the `</dd>` tag. A definition list is created in the following Hands-On Practice.

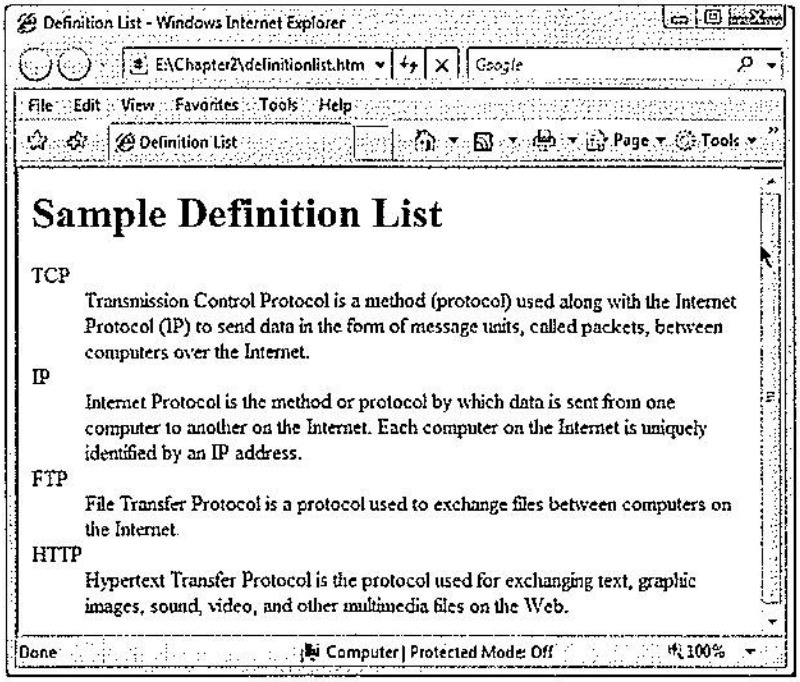


Figure 2.10
Sample definitionlist.html

HANDS-ON PRACTICE 2.6

Open a new file in Notepad. Use the following sample code to create a definition list.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Definition List</title>
</head>
<body>
  <h1>Sample Definition List</h1>
    <dl>
      <dt>TCP</dt>
        <dd>Transmission Control Protocol is a method (protocol) used
along with the Internet Protocol (IP) to send data in the form of
message units, called packets, between computers over the
Internet.</dd>
      <dt>IP</dt>
        <dd>Internet Protocol is the method or protocol by which data
is sent from one computer to another on the Internet. Each computer
on the Internet is uniquely identified by an IP address.</dd>
      <dt>FTP</dt>
```

```
        <dd>File Transfer Protocol is a protocol used to exchange
files between computers on the Internet.</dd>
      <dt>HTTP</dt>
        <dd>Hypertext Transfer Protocol is the protocol used for
exchanging text, graphic images, sound, video, and other multimedia
files on the Web.</dd>
    </dl>
</body>
</html>
```

Save your file as definitionlist.html and test it in a browser. Your page should look similar to the one shown in Figure 2.10 and to the solution in the student files (Chapter2/definitionlist.html). Don't worry if the word wrap is a little different—the important formatting is that each `<dt>` term should be on its own line and the corresponding `<dd>` definition should be indented under it. Try resizing your browser window and notice how the word wrap on the definition text changes.

Ordered Lists

Ordered lists use a numbering or lettering system to organize the information contained in the list. An ordered list can be organized by the use of numerals (the default), uppercase letters, lowercase letters, uppercase Roman numerals, and lowercase Roman numerals. See Figure 2.11 for a sample ordered list.

Figure 2.11
Sample ordered list

Web Server Intro in an Ordered List

1. Apache Web Server
2. Microsoft IIS
3. iPlanet

Ordered lists begin with an `<ol>` tag and end with an `</ol>` tag. Each list item begins with an `<li>` tag and ends with an `</li>` tag. The `type` attribute can be used to change the symbol used for ordering the list. For example, to create an ordered list organized by uppercase letters, use `<ol type="A">`. Table 2.2 documents the `type` attribute and its values for ordered lists.

Table 2.2 `type` attributes for ordered lists

Attribute	Value	Symbol
`type`	`1`	Numerals (the default)
	`A`	Uppercase letters
	`a`	Lowercase letters
	`I`	Roman numerals
	`i`	Lowercase Roman numerals

FAQ **Why is the XHTML code in the Hands-On Practice examples indented?**

It doesn't matter to the browser if XHTML code is indented, but humans find it easier to read and maintain code when it is logically indented. Review the definition list created in Hands-On Practice 2.6. Notice how each tag level (`<dl>`, `<dt>`, and `<dd>`) is indented two spaces. This makes it easier for you or another Web developer to understand the source code in the future. There is no "rule" as to how many spaces to indent, although your instructor or the organization you work for may have a standard. Consistent indentation helps to create more easily maintainable Web pages.

The XHTML code to create the ordered list shown in Figure 2.11 follows:

```
Web Server Intro in an Ordered List
<ol>
  <li>Apache Web Server</li>
  <li>Microsoft IIS</li>
  <li>iPlanet</li>
</ol>
```

Unordered Lists

Unordered lists show a bullet before each entry in the list. This bullet can be one of several types: disc (the default), square, and circle. See Figure 2.12 for a sample unordered list.

Figure 2.12
Sample unordered list

Web Server Intro in an Unordered List

- Apache Web Server
- Microsoft IIS
- iPlanet

Unordered lists begin with an `<ul>` tag and end with an `</ul>` tag. Each list item begins with an `<li>` tag and ends with an `</li>` tag. The `type` attribute can be used to change the type of bullet. For example, to create an unordered list organized with square bullets, use `<ul type="square">`. Table 2.3 documents the `type` attribute and its values for unordered lists.

Table 2.3 `type` attributes for unordered lists

Attribute	Value
`type`	`disc` (the default)
	`square`
	`circle`

The XHTML code to create the unordered list shown in Figure 2.12 follows:

```
Web Server Intro in an Unordered List
<ul>
  <li>Apache Web Server</li>
  <li>Microsoft IIS</li>
  <li>iPlanet</li>
</ul>
```

HANDS-ON PRACTICE 2.7

In this Hands-On Practice you will use heading tags and lists on the same page. Launch Notepad or another text editor and type in the following XHTML:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Headings and Lists</title>
</head>
<body>
  <h1>Web Servers and Web Browsers</h1>
    <h2>Popular Web Servers</h2>
      <ol>
        <li>Apache Web Server</li>
        <li>Microsoft IIS</li>
        <li>iPlanet</li>
      </ol>
    <h2>Popular Web Browsers</h2>
      <ul>
        <li>Internet Explorer</li>
        <li>Firefox</li>
        <li>Opera</li>
      </ul>
</body>
</html>
```

Save your file as heading5.html. Launch a browser and test your page. It should look similar to the pages shown in Figure 2.13. Notice that the Firefox browser configures unordered lists with a diamond bullet point by default instead of the disc as specified by the W3C. You can compare your work with the solution in the student files (Chapter2/heading5.html).

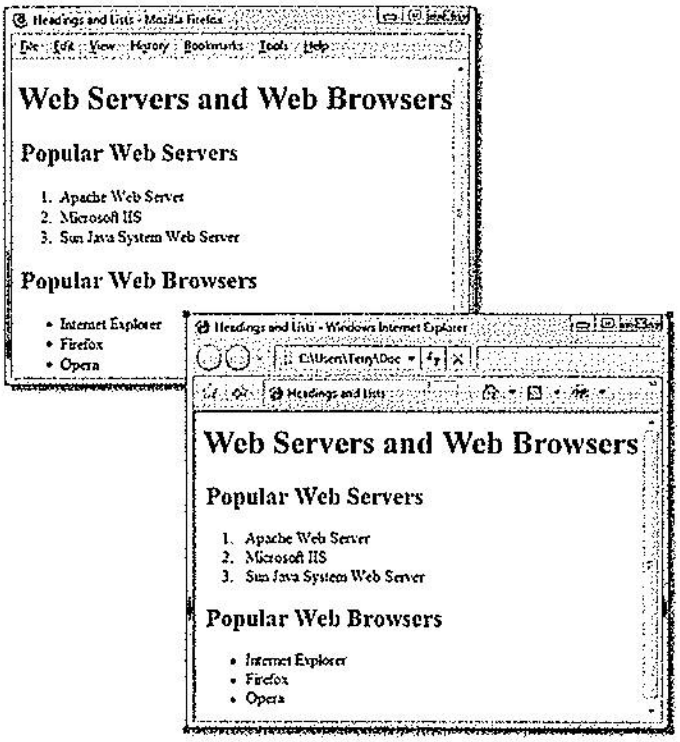


Figure 2.13
Web page using an ordered and unordered list

Take a few minutes to experiment with the `type` attribute. Configure the unordered list to use square bullets. Configure the ordered list to use uppercase letters instead of numerals. Save your page as heading6.html. Test your page in a browser. It should look similar to the page shown in Figure 2.14 and the solution in the student files (Chapter2/heading6.html).

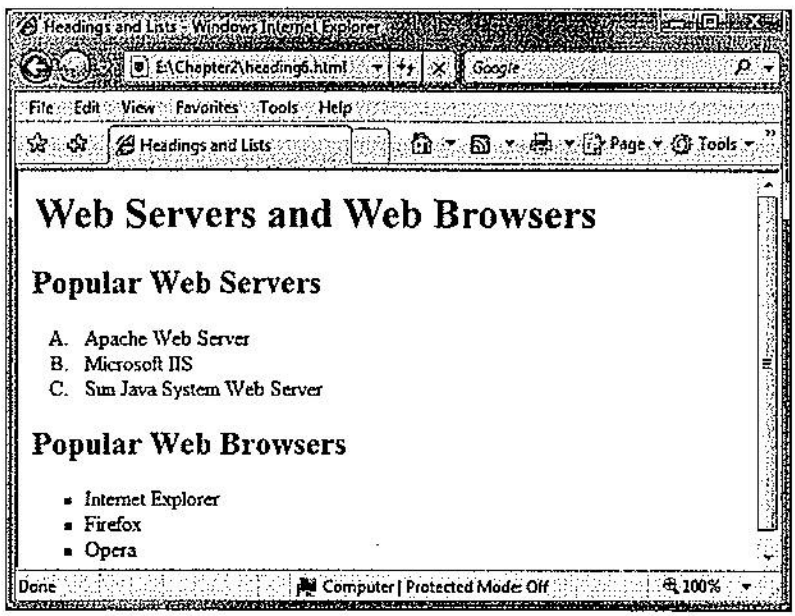
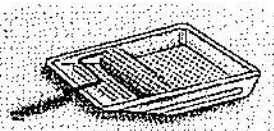


Figure 2.14
Using the `type` attribute with unordered and ordered lists



CHECKPOINT 2.2

1. Describe the features of a heading element and how it configures the text.
2. Describe the difference between ordered lists and unordered lists.
3. Describe the purpose of the blockquote element.

2.7 XHTML—Text Formatting

Text can be formatted in various ways using logical style elements, physical style elements, and special characters. These are considered to be inline-level elements because they can apply to either a section of text or a single character of text. This is not the only method for formatting text; Cascading Style Sheets (introduced in Chapter 3) is commonly used for this purpose.

FAQ

What about the font tag?

The `<font>` tag allows you to configure the typeface, color, and size of the text between the `<font>` and `</font>` container tags. However, the `<font>` tag is deprecated. A deprecated XHTML element or attribute is still supported by XHTML 1.0 Transitional and currently popular browsers, but may not be supported in the future. The W3C recommends using Cascading Style Sheets (see Chapter 3) to format text instead of using the `<font>` tag. If you'd like more information about the `<font>` tag, see Appendix A, XHTML Reference.

XHTML Logical Style Elements

Logical style elements, sometimes called phrase elements, indicate the logical style used to display the text between the container tags. It is up to each browser to interpret that style. For example, the `<strong>` element indicates that the text associated with it be displayed in a "strong" manner in relation to normal text on the page. Usually, but not always, the browser (or other user agent) will display `<strong>` text in bold. A screen reader, such as Jaws or Window-Eyes, might interpret `<strong>` text to indicate that the text should be more strongly spoken. With more and more devices used to access the Web, the use of logical style elements instead of physical style elements (whenever possible) is preferred. Both are still used on the Web.

Note that all logical style elements are container tags—an opening and a closing tag should be used. For example, if you wanted the phone number in the following line to have a strong logical style

```
Call for a free quote for your Web development needs: 888.555.5555
```

the XHTML would look like

```
<p>Call for a free quote for your Web development needs:
<strong>888.555.5555</strong></p>
```

Notice that the opening and closing tags are contained within the paragraph tags (and). This XHTML code is nested properly, follows XML syntax, and is considered to be well formed. An example of improper nesting follows:

```
<p>Call for a free quote for your Web development needs:
<strong>888.555.5555</p></strong>
```

When improperly nested, the and tag pairs overlap each other instead of being nested within each other. Appendix D, Comparison of HTML and XHTML, contains a list of the key syntax rules of XML.

Table 2.4 lists logical style tags and examples of their use.

Table 2.4 Logical style elements

Element	Example	Usage
	strong text	Causes text to be emphasized or to stand out from surrounding text; usually displayed in bold
	emphasized text	Causes text to be emphasized in relation to other text; usually displayed in italics
<cite>	*cite* text	Identifies a citation or reference; usually displayed in italics
<code>	code text	Identifies program code samples; usually a fixed-space font
<dfn>	*dfn* text	Identifies a definition of a word or term; usually displayed in italics
<kbd>	kbd text	Identifies user text to be typed; usually a fixed-space font
<samp>	samp text	Shows program sample output; usually a fixed-space font
<var>	*var* text	Identifies and displays a variable or program output; usually displayed in italics

FAQ

Why do the displays look so similar?

As you look at Table 2.4, you may notice that some tags, such as <cite> and <dfn>, result in the same type of display (italics) as the tag in today's browsers. These tags are logically describing the text as a citation or definition, but the physical display is usually italics in both cases. Cascading Style Sheets (see Chapter 3) are a better way to format elements than logical style tags. However, logical style tags are preferred over physical style tags. If you find this a little confusing and think that there are too many tags with similar purposes, you are correct. Please keep in mind that Cascading Style Sheets is the preferred method to format text—not physical style and logical style elements. However, we introduce physical style and logical style elements in this chapter because they are still used on the Web.

XHTML—Physical Style Elements

Physical style elements are sometimes called font style elements because they provide specific font instructions for the browser. This type of tag is still commonly used and generated by some Web authoring tools. Be aware that logical style elements and Cascading Style Sheets provide for a wider range of Web access. Physical style elements

are covered in this book because many existing Web pages use them. Table 2.5 lists physical style tags and examples of their use.

Table 2.5 Physical style elements

Element	Example	Usage
`<b>`	**bold** text	Displays text as bold
`<i>`	*emphasized* text	Displays text in italics
`<big>`	big text	Displays text larger than normal size
`<small>`	small text	Displays text smaller than normal size
`<sub>`	$_{sub}$text	Displays small text below the baseline
`<sup>`	^{sup}text	Displays small text above the baseline
`<strike>`	~~strike~~ text	Displays text with a line through it (deprecated)
`<u>`	u text	Displays text underlined; avoid using this because underlined text can be confused with hyperlinks (deprecated)
`<tt>`	`teletype` text	Displays text in teletype or fixed-space font

You may have noticed that the `<strong>` logical style tag usually has the same effect as the `<b>` physical style tag. Also, the `<em>` logical style tag usually has the same effect as the `<i>` physical style tag. In order to create XHTML that describes logical styles instead of font instructions for browsers, use `<strong>` instead of `<b>` and use `<em>` instead of `<i>`. As you continue to study Web development, you will learn about Cascading Style Sheets and their use in text formatting.

Special XHTML Characters

In order to use special characters such as quotation marks, greater than (>), lesser than (<), and the copyright symbol (©) in your Web document, you need to use special characters, sometimes called entity characters. For example, if you wanted to include a copyright line on your page as follows:

```
© Copyright 2008 My Company. All rights reserved.
```

You would use the special character `©` to display the copyright symbol. The XHTML would look as follows:

```
<p>&copy; Copyright 2008 My Company. All rights reserved.</p>
```

Another useful special character is ` `, which stands for nonbreaking space. You may have noticed that Web browsers treat multiple spaces as a single space. If you need a small number of spaces in your text, you may use ` ` multiple times to indicate multiple blank spaces. This is acceptable if you simply need to tweak the position of an element a little. If you find that your Web pages contain many ` ` special characters in a row, you should use a different method to align elements, such as a table or Cascading Style Sheets.

See Table 2.6 and Appendix B, Special Characters, for a description of special characters and their codes.

Table 2.6 Common special characters

Character	Entity Name	Code
"	Quotation mark	`"`
©	Copyright symbol	`©`
&	Ampersand	`&`
Empty space	Nonbreaking space	` `

HANDS-ON PRACTICE 2.8

Figure 2.15 shows the Web page you will create. Launch Notepad and open one of the Web page files that you have already created, such as blockquote.html from Hands-On Practice 2.5. Modify the title of the Web page by changing the text between the `<title>` and `</title>` tags to Web Design Steps. Since our Web Design Steps page will be quite different from the previous page you created, delete the code between the `<body>` and `</body>` tags. Save the file as design.html.

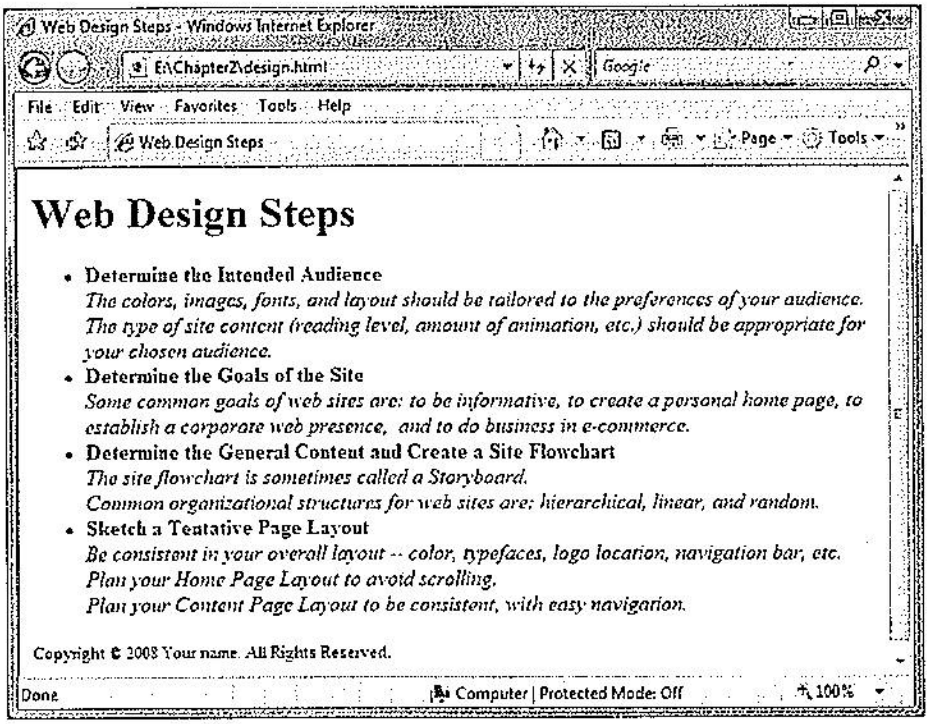


Figure 2.15
Sample design.html

The sample page shown in Figure 2.15 contains a heading, an ordered list, and copyright information.

Configure the heading Web Design Steps as a level 1 heading (`<h1>`) as follows:

```
<h1>Web Design Steps</h1>
```

Now create the unordered list. The first line of each bulleted item is the title of the Web design step. In the sample, each step title should be strong, or stand out from the rest of the text. The subsequent lines in each bulleted item should be emphasized. The code for the beginning of the unordered list follows:

```
<ul>
  <li>Determine the Intended Audience</strong><br />
  <em>The colors, images, fonts, and layout should be tailored to the
preferences of your audience. The type of site content (reading
level, amount of animation, etc.) should be appropriate for your
chosen audience.</em>
  </li>
```

Edit your design.html file and code the entire ordered list. Remember to code the closing `</ul>` tag at the end of the list. Don't worry if your text wraps a little differently—your screen resolution or browser window size may be different.

Finally, configure the copyright information. This should be smaller than the rest of the text. Use the special character, `©`, for the copyright symbol. The code for the copyright line follows:

```
<p><small>Copyright &copy; 2008 Your name. All Rights Reserved.
</small></p>
```

How did you do? Compare your work to the sample in the student files (Chapter2/design.html).

2.8 XHTML—Hyperlinks

The Anchor Element

The anchor element can be used to specify a hyperlink reference (`href`) to a Web page you want to display. Each hyperlink begins with an `<a>` tag and ends with an `</a>` tag. The opening and closing anchor tags surround the text to click to perform the hyperlink.

You have probably seen many links on the Web but may have never thought about how they are created. To create an absolute link to a Web site such as Yahoo!, you would create a hyperlink with the URL for Yahoo! for the value of the **`href`** attribute as follows:

```
<a href="http://yahoo.com">Yahoo!</a>
```

"Yahoo!", the text contained between the anchor tags, is displayed in the browser window. By default, this text is underlined and blue. Figure 2.16 shows an example of a hyperlink to the Yahoo! Web site in a browser.

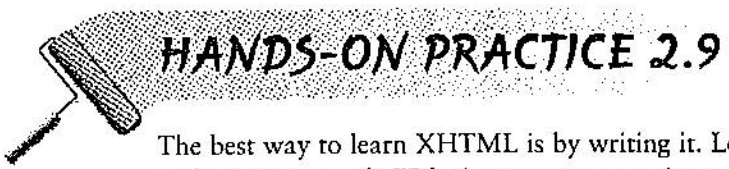
Figure 2.16
A hyperlink to the Yahoo! Web site

Absolute and Relative Links

The link to Yahoo! you just created is an absolute link. Notice that the XHTML code for the link indicates the protocol being used, http://, and continues with the domain name, yahoo.com. This indicates the absolute location of the Web resource. Use absolute links when you are creating links to other Web sites.

When you need to link to Web pages within your site, use a relative link. This link does not begin with http://. It only contains the file name or file name and folder of the Web page you want to display. The link location is relative to the page currently being displayed. For example, if you had a home page called index.html and wanted to link to a page with your resumé (called resume.html) located in the same folder as index.html, the XHTML for the relative link would be as follows:

```
<a href="resume.html">My Resume</a>
```



The best way to learn XHTML is by writing it. Let's experiment with the anchor tag and create a sample Web site to use to practice creating hyperlinks.

First, create a new folder called mywebsite.

- **Windows XP Users.** Launch Windows Explorer by selecting Start, Programs, Windows Explorer with your pointing (mouse) cursor. Click your drive to select it. Select File, New, Folder.
- **Windows Vista Users.** Launch the Computer folder by selecting Start, Computer with your pointing (mouse) cursor. Click your drive to select it. Select Organize, New Folder.

Name your folder mywebsite. This site could be a personal Web site. It will contain a home page called index.html and two content pages called resume.html and favorites.html. A sample site map that was created using Adobe Dreamweaver (see Figure 2.17) shows the architecture of the site—a home page (index.html) with major links to two pages (resume.html and favorites.html).

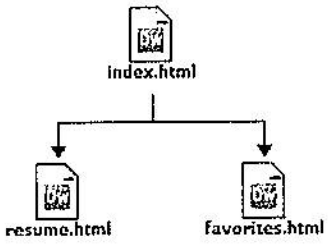


Figure 2.17
Site map

Now create the home page for your mywebsite Web site. Launch Notepad or another text editor and type in the tags found on every Web page (XML declaration, `DTD`, `html`, `head`, `title`, `body`). In the body of the Web page create the following:

- A heading: My Web Site
- An absolute link to Yahoo!

- An absolute link to the Web site of your school
- A relative link to resume.html
- A relative link to favorites.html

Save your page as index.html in the mywebsite folder. Display your page in a browser. It should look similar to the page shown in Figure 2.18. Compare your work to the sample in the student files (Chapter2/2.9/index.html). *Hint*: Check Appendix B, Special Characters, for the XHTML code for the "é".

Figure 2.18
Sample index.html

Test your page by clicking each link. When you click the absolute links to Yahoo! and your school you should see those pages displayed if you are connected to the Internet. The relative links should not work yet—let's create those pages next.

Create the resume.html page. Launch Notepad or another text editor and type in the tags found on every Web page (XML declaration, `DTD`, `html`, `head`, `title`, `body`). In the body of the Web page place the following:

- A heading of Resumé
- Some text that describes your job objective
- A navigation bar that contains a relative link to the Home page (index.html), and a relative link to the Favorites page (favorites.html)

FAQ

What if my absolute links don't work?

Check the following:

- Are you currently connected to the Internet?
- Are you certain that you spelled the URLs of the Web sites correctly?
- Did you begin with http://?
- When you place your mouse over a link, the URL will display in the status bar in the lower edge of the browser window. Verify that this is the correct URL.
- *Hint*: When you are about to put an absolute link in a Web page, display the Web site in a browser, then copy and paste the URL. Don't rely on typing the URL accurately.

See Figure 2.19 for a sample Resumé page. Save your Resumé page as resume.html in your mywebsite folder.

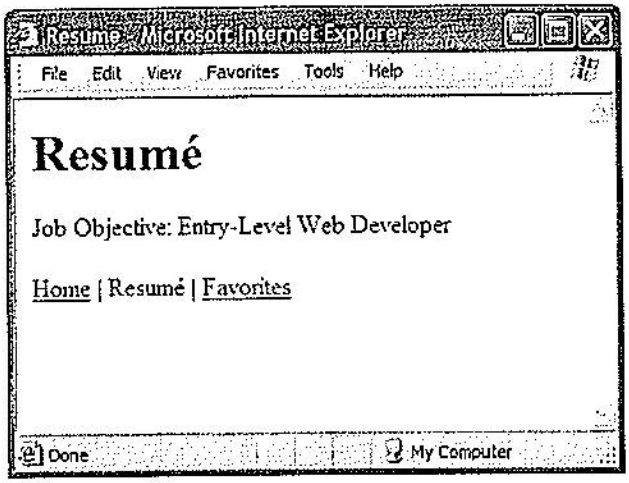


Figure 2.19
Sample resume.html

Test your index.html page again. This time when you click the Resumé link, your new page should display. Use the Home link on your resume.html page to redisplay your home page.

Create the favorites.html page. Launch Notepad or another text editor and type in the tags found on every Web page (XML declaration, `DTD`, `html`, `head`, `title`, `body`). In the body of the Web page place the following:

- A heading: Favorite Sites
- An unordered list that contains the following categories:

  Hobbies
  XHTML
  JavaScript
  Professional Groups
- A navigation bar that contains a relative link to the Home page (index.html) and a relative link to the Resumé page (resume.html)

See Figure 2.20 for a sample Favorites page. Save your page as favorites.html in your mywebsite folder.

Figure 2.20
Sample favorites.html

Test your index.html page again and try the links between the Home page, Resumé page, and Favorites pages. Don't worry if the links don't work perfectly the first time. If you have problems, carefully examine the source code of the pages and verify the existence of the files using Windows Explorer.

FAQ

What if my relative links don't work?

Check the following:

- Did you save your index.html and resume.html pages in your mywebsite folder?
- Did you save the files with the names as requested? Use Windows Explorer or My Computer to verify the actual names of the files you saved.
- Did you type the file names correctly in the link's `href` property? Check for typographical errors.
- When you place your mouse over a link, the file name of a relative link will display in the status bar in the lower edge of the browser window. Verify that this is the correct file name.
- On many operating systems such as UNIX or Linux, the use of uppercase and lowercase in file names matters—make sure that the file name and the reference to it are in the same case. It's a good practice to always use lowercase for file names used on the Web.
- *Hint*: Tiny details such as spelling file names correctly and consistently are very important in Web development.

E-Mail Links

The anchor tag can also be used to create e-mail links. An e-mail link will automatically launch the default mail program configured for the browser. It is similar to an external hyperlink with the following two exceptions:

- It uses mailto: instead of http://.
- It launches the default e-mail application for the visitor's browser with your e-mail address as the recipient.

For example, to create an e-mail link to the e-mail address help@terrymorris.net, code the following:

```
<a href="mailto:help@terrymorris.net">help@terrymorris.net</a>
```

It is good practice to place the e-mail address both on the Web page and within the anchor tag. Not everyone has an e-mail program configured with his or her browser. By placing the e-mail address in both places, you increase usability for all your visitors.

HANDS-ON PRACTICE 2.10

Open the home page of your mywebsite Web site and add an e-mail link to the bottom of the page. Save and test it in a browser. The page should look similar to the page shown in Figure 2.21. Compare your work with the sample in the student files (Chapter2/2.10/index.html).

Figure 2.21
E-mail link added to index.html

This section provided a quick introduction to the anchor element. You should now be able to code different types of text hyperlinks: e-mail links, links relative to a Web page, and absolute links to other Web sites. As you continue to study, you will learn to use images as hyperlinks (Chapter 4), to code links internal to a Web page (Chapter 7), and to target specific windows (Chapter 7).

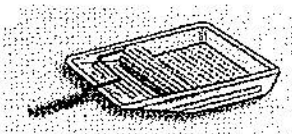
Accessibility and Links

Focus on Accessibility

Visually challenged visitors who are using a screen reader can configure the software to display a list of the hyperlinks in the document. In addition, some popular browsers, such as Opera (visit http://www.opera.com for free download information), provide this feature as a convenience for all users. However, a list of links is only useful if the text describing each link is actually helpful and descriptive. For example, on your college Web site a "Search the course schedule" link would be more useful than a link that simply says "More information." Keep this in mind as you are coding hyperlinks in your Web pages.

Can you share some tips on using links?

- Make your link names descriptive and brief to minimize possible confusion.
- Avoid using the phrase "Click here for" in your links. In the beginning of the Web, this phrase was needed because clicking links was a new experience for Web users. Now that the Web is a daily part of our lives, this phrase seems slightly redundant and almost archaic.
- Try not to bury links in large blocks of text—use bullets or definition lists. It is more difficult to read Web pages than printed pages.
- Be careful when linking to external Web sites. The Web is dynamic and it's possible that the external site may change the name of the page or even delete the page. If this happens, your link will be broken.



CHECKPOINT 2.3

1. Provide a reason for using logical style tags rather than physical style tags.
2. Describe the purpose of special characters.
3. Describe when to use an absolute link. Is the http protocol used in the `href` value?
4. Describe when to use a relative link. Is the http protocol used in the `href` value?

2.9 XHTML Validation

The W3C has a free Markup Validation Service available at http://validator.w3.org/ that will validate your XHTML code and check it for syntax errors. XHTML validation provides students with quick self-assessment—you can prove that your code uses correct syntax. In the working world, XHTML validation serves as a quality assurance tool. Invalid code may cause browsers to render the pages slower than otherwise.

HANDS-ON PRACTICE 2.11

In this Hands-On Practice you will use the W3C Markup Validation Service to validate a Web page. This example uses the design.html page completed in Hands-On Practice 2.8 (student files Chapter2/design.html). Locate design.html and open it in Notepad. We will add an error to the design.html page. Delete the first closing `</li>` tag. This modification should generate several error messages. The first error message will be a direct result of the incorrect syntax.

Next, attempt to validate the design.html file. At the time this was written, the W3C warns that their validator may not work with all versions of Internet Explorer browsers. Therefore, launch a Mozilla-based browser such as Firefox or Netscape and visit the W3C Markup Validation Service file upload page at http://validator.w3.org/#validate_by_upload. Click the Browse button and select the Chapter2/design.html file from your computer. Select More Options. Verify that the check boxes next to Show Source and Verbose Output are checked, as shown in Figure 2.22. Click the Check button to upload the file to the W3C site.

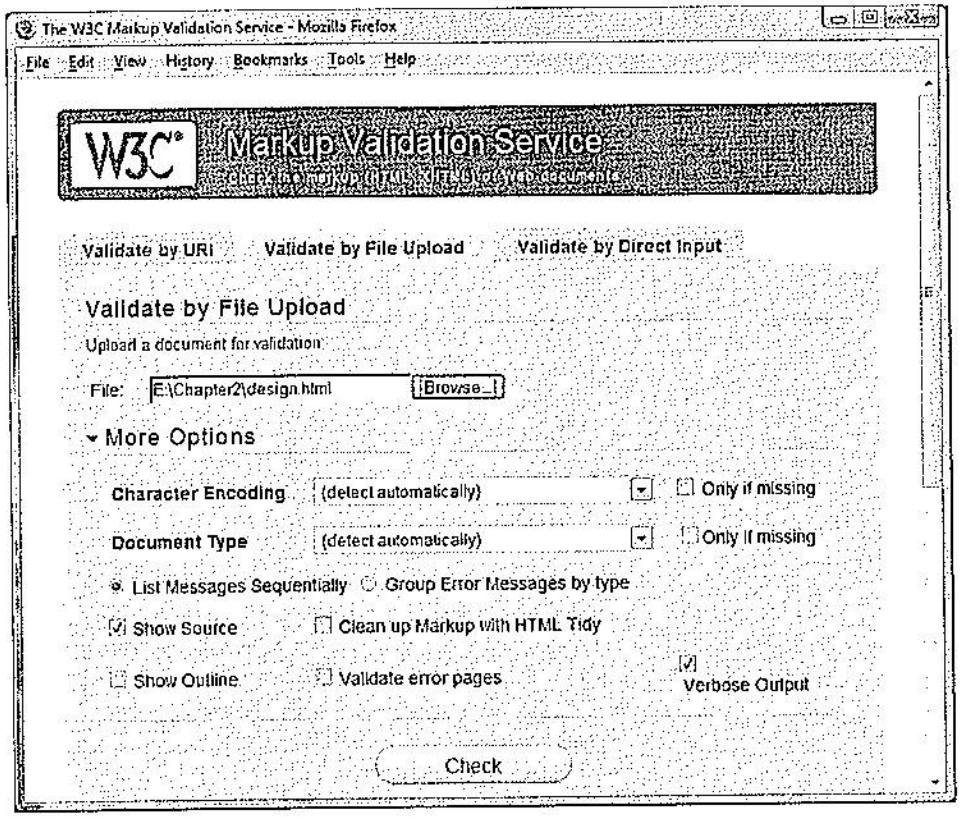


Figure 2.22
Use Firefox or Netscape to validate your page

Your display should be similar to that shown in Figure 2.23. Notice the "Result: Failed validation, 8 Errors" message.

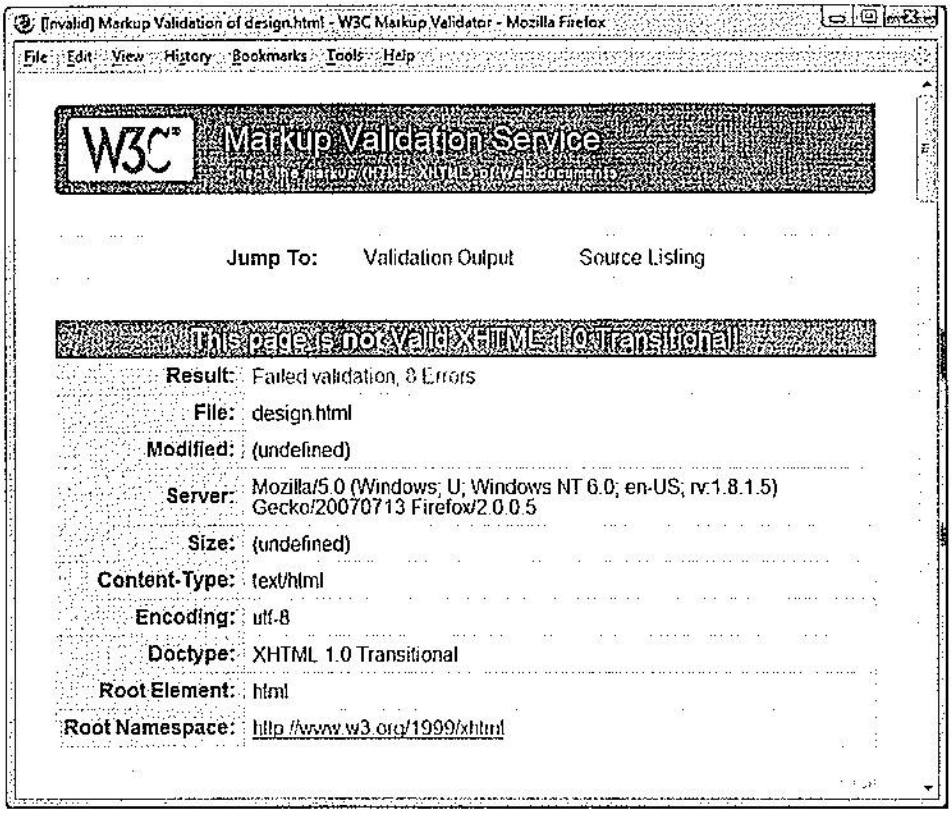


Figure 2.23 The validation results indicate errors

You can view the errors by scrolling down the page, as shown in Figure 2.24.

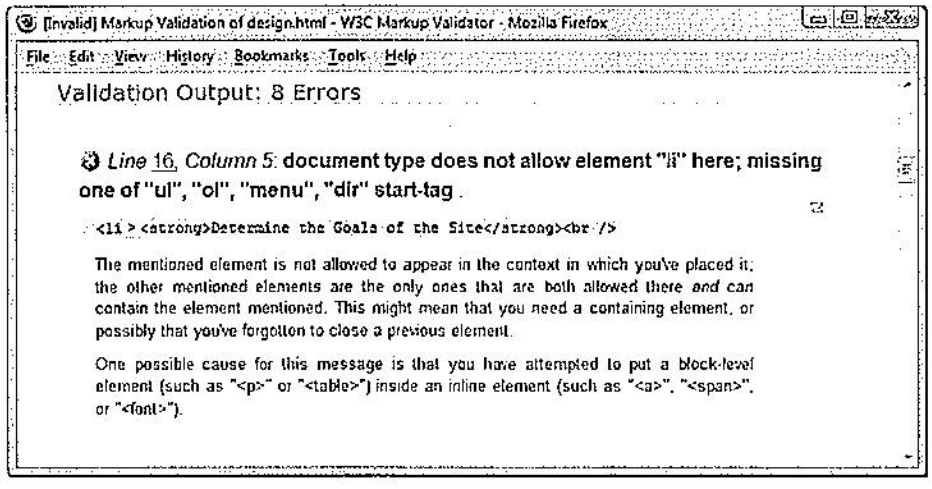


Figure 2.24 The error indicates line 16

Notice that the message indicates line 16—which is the first line after the missing closing `</li>` tag. XHTML error messages often point to the line that follows the error. The text of the message 'document type does not allow element "li" here; missing one of "ul", "ol", "menu", "dir" start-tag' lets you know that something is wrong. It's up

to you to figure out what it is. A good place to start is to check your container tags and make sure they are in pairs. In this case, that is the problem. You can scroll down to view the other errors. However, since multiple error messages are often displayed after a single error occurs, it's a good idea to fix one item at a time and then revalidate.

Edit the design.html file in Notepad and add the missing `</li>` tag. Save the file. Launch Firefox or Netscape and visit http://validator.w3.org/#validate_by_upload. Select your file, select More Options, and verify the Show Source and Verbose Output check boxes are checked. Click the Check button to begin the validation.

Your display should be similar to that shown in Figure 2.25. Notice the "Result: Passed validation" message. This means your page passed the validation test. Congratulations, your design.html page is a valid XHTML page! It's a good practice to validate your Web pages. However, when validating code use common sense. Since Web browsers still do not completely follow W3C recommendations, there will be situations, such as when adding multimedia to a Web page, when XHTML code configured to work reliably across a variety of browsers and platforms will not pass XHTML validation.

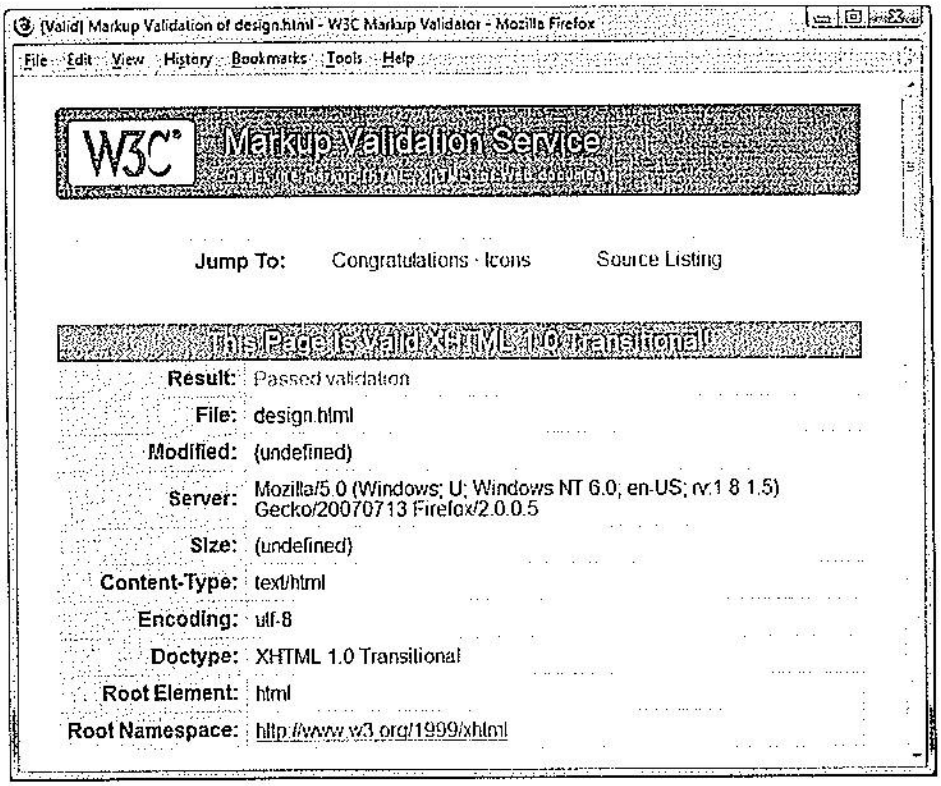
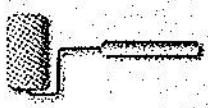


Figure 2.25 The page has passed the validation test

CHAPTER SUMMARY



This chapter provided an introduction to XHTML. It began with an introduction to HTML, discussed the transition to XHTML, continued with the anatomy of a Web page, introduced inline- and block-level formatting, and demonstrated coding anchor elements to link Web pages. If you worked along with the samples in the chapter, you should be ready to create some Web pages on your own. The Hands-On Exercises and Web Case Studies that follow will provide some practice.

Visit the textbook Web site at http://www.webdevfoundations.net for the links listed in this chapter and for updated information.

Key Terms

`©`
` `
`<a>`
`<blockquote>`
`<body>`
`<br />`
`<dd>`
`<dl>`
`<dt>`
`<em>`
`<font>`
`<h1>`
`<h6>`
`<head>`
`<html>`
`<li>`
`<ol>`
`<p>`
`<strong>`
`<title>`
`<ul>`
absolute link
anchor element
attribute
block-level element
body
definition list
deprecated
`DOCTYPE`
document type definition (DTD)
element
e-mail link
eXtensible HyperText Markup Language (XHTML)
head
header
headings
`href` attribute
hyperlink
Hypertext Markup Language (HTML)
inline-level element
left alignment
logical style element
ordered list
paragraph element
physical style element
relative link
special characters
tag
unordered list
well-formed document
XHTML 1.0 Frameset
XHTML 1.0 Strict
XHTML 1.0 Transitional
XHTML validation
XML (eXtensible Markup Language)
XML namespace (xmlns)

Review Questions

Multiple Choice

1. Which tag pair is used to create a new paragraph?
   a. `<new paragraph> </new paragraph>`
   b. `<paragraph> </paragraph>`
   c. `<p> </p>`
   d. `<para> </para>`

2. Which tag pair is used to create the smallest heading?
   a. `<h1> </h1>`
   b. `<h9> </h9>`
   c. `<h type="smallest"> </h>`
   d. `<h6> </h6>`

3. Which tag is used to force the browser to display the next text or element on a new line?
   a. `<new line />`
   b. `<nl />`
   c. `<br />`
   d. `<line />`

4. Which tag is used to link Web pages to each other?
   a. `<link>` tag
   b. `<hyperlink>` tag
   c. `<a>` tag
   d. `<body>` tag

5. What is the default alignment for headings and paragraphs?
   a. `center`
   b. `left`
   c. `right`
   d. wherever you type them in the source code

6. Which type of XHTML list will automatically number the items for you?
   a. numbered list
   b. ordered list
   c. unordered list
   d. definition list

7. Which type of XHTML list contains bullets?
   a. bullet list
   b. ordered list
   c. unordered list
   d. definition list

8. When do you need to use a fully qualified URL in a hyperlink?
   a. always
   b. when linking to a Web page file on the same site
   c. when linking to a Web page file on an external site
   d. never

9. Which tag pair contains the items in an ordered or unordered list?
   a. `<item> </item>`
   b. `<li> </li>`
   c. `<dd> </dd>`
   d. all of the above

10. What does an e-mail link do?
    a. automatically sends you an e-mail message with the visitor's e-mail address as the reply-to field
    b. launches the default e-mail application for the visitor's browser with your e-mail address as the recipient
    c. displays your e-mail address so that the visitor can send you a message later
    d. links to your mail server

Fill in the Blank

11. The `<blockquote>` element is used to _____.

12. _____ can be used to display characters such as the copyright symbol.

13. The _____ is the preferred element to use when you need to emphasize text.

14. The _____ is used to place a nonbreaking space on a Web page.

Short Answer

15. Explain why it is good practice to place the e-mail address on the Web page and within the anchor tag when creating an e-mail link.

Apply Your Knowledge

1. **Predict the Result.** Draw and write a brief description of the Web page that will be created with the following XHTML code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Predict the Result</title>
</head>
<body>
  <h1><em>Favorite Sites</em></h1>
  <ol>
    <li><a href="http://myspace.com">My Space</a></li>
    <li><a href="http://google.com">Google</a></li>
  </ol>
  <p><small>Copyright &copy; 2008 Your name here</small></p>
</body>
</html>
```

2. **Fill in the Missing Code.** This Web page should display a heading and a definition list, but some XHTML tags, indicated by <_> are missing. Fill in the missing code.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Door County Wild Flowers</title>
</head>
<body>
  <_>Door County Wild Flowers<_>
  <dl>
    <dt>Trillium<_>
      <_>This white flower blooms from April through June in
wooded areas.<_>
      <_>Lady Slipper<_>
      <_>This yellow orchid blooms in June in wooded areas.</dd>
    <_>
</body>
</html>
```

3. **Find the Error.** Why won't this page display in a browser?

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Find the Error<title>
</head>
<body>
  <h1>Why don't I display?</h1>
</body>
</html>
```

Hands-On Exercises

1. Write the XHTML to display your name in the largest size heading element.
2. Write the XHTML to create an absolute link to a Web site whose domain name is yahoo.com.
3. Write the XHTML for an unordered list to display the days of the week.
4. Write the XHTML for an ordered list that uses uppercase letters to order the item. This ordered list will display the following terms: HTML, XML, and XHTML.
5. Think of a favorite quote by someone you admire. Write the XHTML code to display the person's name in a heading and the quote in a paragraph.
6. Modify the following code snippet to use logical style tags instead of physical style tags.

```
<p>A diagram of the organization of a web site is called a <b>site
map</b> or <b>storyboard</b>. <i>Creating the <b>site map</b> is
one of the initial steps in developing a web site.</i></p>
```

7. Create a Web page about your favorite movie. Include the name of the movie, the actors and actresses, a hyperlink to a Web site that displays a review of the movie, and a brief description of the movie. Use an unordered list to organize the names of the actors and actresses. Save the page as movie.html. Open your file in Notepad and print the source code for the page. Display your page in a browser and print the page. Hand in both printouts to your instructor.
8. Create a Web page that uses a definition list to display three network protocols (see Chapter 1) and their descriptions. Include a hyperlink to a Web site that provides information about the protocols. Add an appropriate heading to the page. Save the page as network.html. Open your file in Notepad and print the source code for the page. Display your page in a browser and print the page. Hand in both printouts to your instructor.
9. Create a Web page about your favorite musical group. Include the name of the group, the individuals in the group, a hyperlink to the group's Web site, your favorite three (or fewer if the group is new) CD releases, and a brief review of each CD.
   - Use an unordered list to organize the names of the individuals.
   - Use a definition list for the names of the CDs and your reviews.

   Save the page as band.html. Open your file in Notepad and print the source code for the page. Display your page in a browser and print the page. Hand in both printouts to your instructor.
10. Create a Web page about your favorite recipe. Use an unordered list for the ingredients and an ordered list to describe the steps needed to prepare the food. Include a hyperlink to a Web site that offers free recipes. Save the page as recipe.html. Open your file in Notepad and print the source code for the page. Display your page in a browser and print the page. Hand in both printouts to your instructor.

Web Research

1. There are many HTML and XHTML tutorials on the Web. Use your favorite search engine to discover them. Choose two that are helpful. For each, print out the home page or other pertinent page and create a Web page that contains the answers to the following questions:
   a. What is the URL of the Web site?
   b. Is the tutorial geared toward the beginner level, intermediate level, or both levels?
   c. Would you recommend this site to others? Why or why not?
   d. List one or two concepts that you learned from this tutorial.

   Open your file in Notepad and print the source code for the page. (*Hint*: Select File, Print.) Display your page in a browser and print the page. Hand in both printouts to your instructor.

Focus on Web Design

1. You are learning the syntax of XHTML. However, coding alone does not make a Web page—design is very important. Surf the Web and find two Web pages—one that is appealing to you and one that is unappealing to you. Print each page. Create a Web page that answers the following questions for each of your examples.
   a. What is the URL of the Web site?
   b. Is the page appealing or unappealing? List three reasons for your answer.
   c. If the page is unappealing, what would you do to improve it?

   Open your file in Notepad and print the source code for the page. Display your page in a browser and print the page. Hand in both printouts to your instructor.

WEB SITE CASE STUDY

Each of the following case studies continues throughout most of the text. This chapter introduces each Web site scenario, presents the site map or storyboard, and directs you to create two pages for the site.

JavaJam Coffee House

Julio Perez is the owner of the JavaJam Coffee House, a gourmet coffee shop that serves snacks, coffee, tea, and soft drinks. Local folk music performances and poetry readings are held a few nights during the week. The customers of JavaJam are mainly college students and young professionals. Julio would like a Web presence for his shop that will display his services and provide a calendar for the performances. He would like a home page, menu page, music performance schedule page, and job opportunities page.

A site map for the JavaJam Coffee House Web site is shown in Figure 2.26.

The site map describes the architecture of the Web site, a Home page with three main content pages: Menu, Music, and Jobs.

Figure 2.26
JavaJam site map

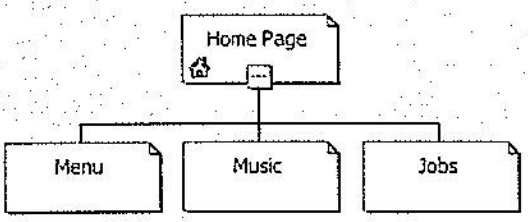
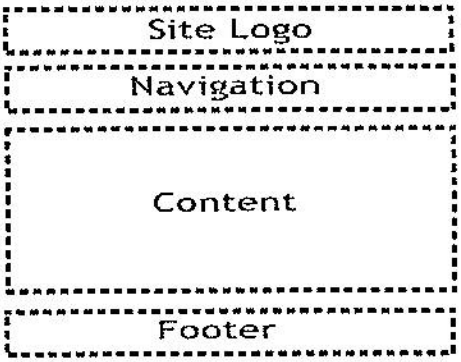


Figure 2.27 displays a sample layout for the pages. It contains a site logo, a navigation area, a content area, and a footer area for copyright information.

Figure 2.27
JavaJam page layout



You have two tasks in this case study:

1. Create the Home page: index.html.
2. Create the Menu page: menu.html.

Hands-On Practice Case

Create a folder called javajam to contain your JavaJam Web site files.

1. **The Home Page.** You will use Notepad to create the Home page for the JavaJam Coffee House Web site. The Home page is shown in Figure 2.28.

Figure 2.28
JavaJam index.html

Launch Notepad and create a Web page with the following specifications:

- **Web page:** Use a descriptive page title—the company name is a good choice for a business Web site.
- **Logo area:** Use `<h1>` for the JavaJam Coffee House logo.
- **Navigation:** Place the following text within a paragraph:

  Home Menu Music Jobs

  Code anchor tags so that Home links to index.html, Menu links to menu.html, Music links to music.html, and Jobs links to jobs.html.
- **Content:** Place the following content in an unordered list:

  Specialty Coffee and Teas

  Bagels, Muffins, and Organic Snacks

  Music and Poetry Readings

  Open Mic Night
- **Contact information:** Place the address and phone number information within a paragraph below the unordered list. *Hint*: Use line break tags to help you configure this area.

  12312 Main Street

  Mountain Home, CA 93923

  1-888-555-5555
- **Footer:** Place the following information in a small text size (use the `<small>` physical style element) and emphasized font style (use the `<em>` logical style element):

  Copyright © 2008 JavaJam Coffee House

  Place your name in an e-mail link on the line under the copyright information.

The page in Figure 2.28 may seem a little sparse, but don't worry, as you gain experience and learn to use more advanced techniques, your pages will look more professional. White space (blank space) on the page can be added with `<br />` tags where needed. Your page does not need to look exactly the same as the sample. Your goal at this point should be to practice and get comfortable using XHTML.

Save your page in the javajam folder and name it index.html.

2. **The Menu Page.** Create the Menu page shown in Figure 2.29. A productivity technique is to create new pages based on existing pages—so you can benefit from your previous work. Your new Menu page will use the index.html page as a starting point.

   Open the index.html page for the JavaJam Web site in Notepad. Select File, Save As, and save the file with the new name of menu.html in the javajam folder. Now you are ready to edit the page.

   - Modify the page title. Change the text contained between the `<title>` and `</title>` tags to JavaJam Coffee House Menu.
   - Delete the unordered list and the contact information.
   - Add the menu content to the page using a definition list. Use the `<dt>` element to contain each menu item name. Configure the menu item name to display in bold text (use the `<strong>` element). Use the `<dd>` element to contain the menu item description.

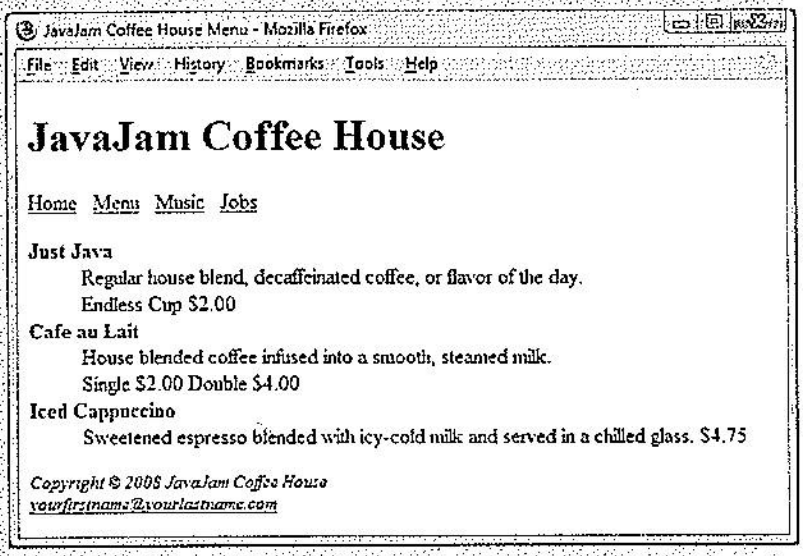


Figure 2.29
JavaJam menu.html

- The menu items names and descriptions are as follows:

  Just Java
  Regular house blend, decaffeinated coffee, or flavor of the day.
  Endless Cup $2.00

  Cafe au Lait
  House blended coffee infused into a smooth, steamed milk.
  Single $2.00 Double $4.00

  Iced Cappuccino
  Sweetened espresso blended with icy-cold milk and served in a chilled glass.
  $4.75

Save your page and test it in a browser. Test the hyperlink from the menu.html page to index.html. Test the hyperlink from the index.html page to menu.html. If your links do not work, review your work with close attention to these details:

- Verify that you have saved the pages with the correct names in the correct folder.
- Verify your spelling of the page names in the anchor tags.
- After you make changes, test again.

Fish Creek Animal Hospital

Magda Patel is a veterinarian and owner of the Fish Creek Animal Hospital. Her customers are local pet owners who range in age from children to senior citizens. Magda would like a Web site to provide information to her current and potential customers. She has requested a home page, a services page, an ask the vet page, and a contact page.

A site map for the Fish Creek Animal Hospital Web site is shown in Figure 2.30.

The site map describes the architecture of the Web site, a Home page with three main content pages: Services, Ask the Vet, and Contact.

Figure 2.30
Fish Creek site map

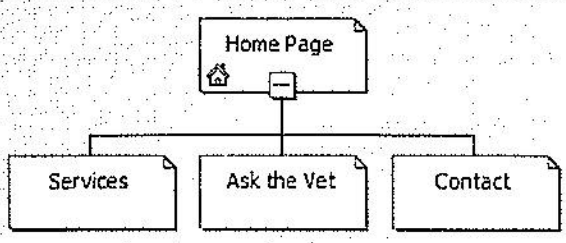
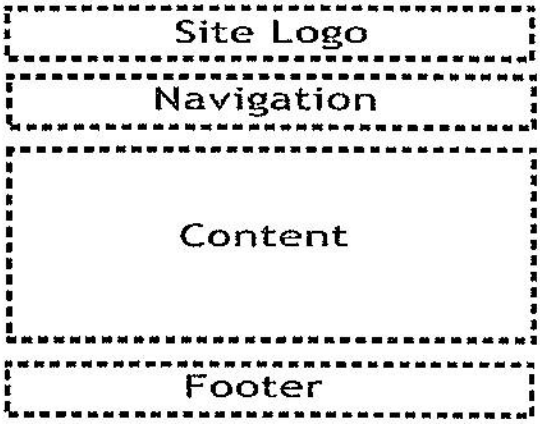


Figure 2.31 displays a sample page layout. It contains a site logo, a navigation area, a content area, and a footer area for copyright information.

Figure 2.31
Fish Creek page layout



You have two tasks in this case study:

1. Create the Home page: index.html.
2. Create the Services page: services.html.

Hands-On Practice Case

Create a folder called fishcreek to contain your Fish Creek Web site files.

1. **The Home Page.** You will use Notepad to create the Home page for the Fish Creek Animal Hospital Web site. The Home page is shown in Figure 2.32.

   Launch Notepad and create a Web page with the following specifications:

   - Web page: Use a descriptive page title—the company name is a good choice for a business Web site.
   - Logo area: Use `<h1>` for the Fish Creek Animal Hospital logo.
   - Navigation: Place the following text using the `<strong>` logical style element within a paragraph:

     Home Services Ask the Vet Contact

     Code anchor tags so that Home links to index.html, Services links to services.html, Ask the Vet links to askvet.html, and Contact links to contact.html.

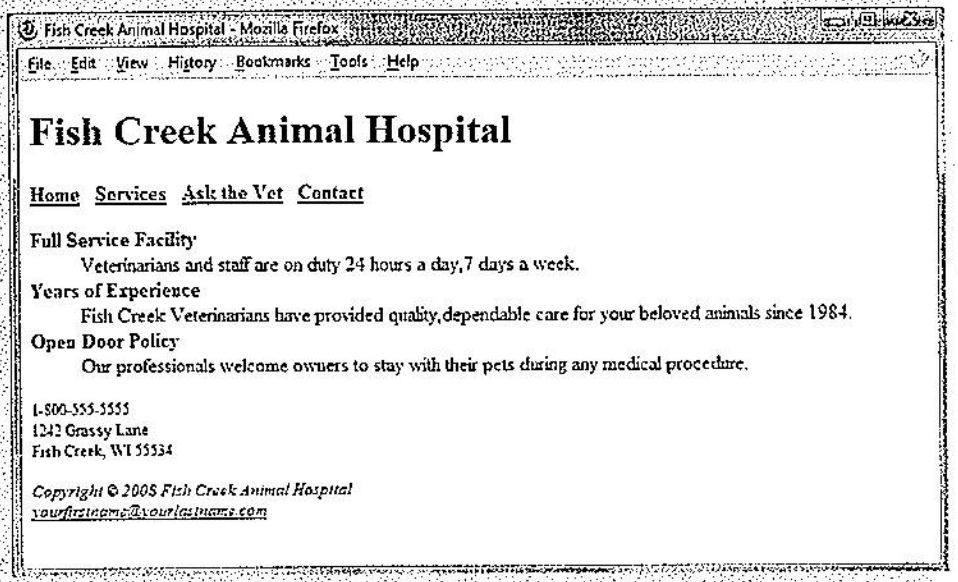


Figure 2.32
Fish Creek index.html

- **Content:** Place the following content in a definition list:

  Full Service Facility

  Veterinarians and staff are on duty 24 hours a day, 7 days a week.

  Years of Experience

  Fish Creek Veterinarians have provided quality, dependable care for your beloved animals since 1984.

  Open Door Policy

  Our professionals welcome owners to stay with their pets during any medical procedure.

- **Contact information:** Place the address and phone number information within a paragraph below the unordered list. *Hints:* Use line break tags to help you configure this area. The text size is configured with the `<small>` physical style element.

  1-800-555-5555

  1242 Grassy Lane

  Fish Creek, WI 55534

- **Footer:** Place the following information in a small text size (use the `<small>` physical style element) and emphasized font style (use the `<em>` logical style element):

  Copyright © 2008 Fish Creek Animal Hospital

  Place your name under the copyright information.

The page in Figure 2.32 may seem a little sparse, but don't worry, as you gain experience and learn to use more advanced techniques, your pages will look more professional. White space (blank space) on the page can be added with `<br />` tags where needed. Your page does not need to look exactly the same as the sample. Your goal at this point should be to practice and get comfortable using XHTML.

Save your page in the fishcreek folder and name it index.html.

2. **The Services Page.** Create the Services page shown in Figure 2.33. A productivity technique is to create new pages based on existing pages—so you can benefit from your previous work. Your new Services page will use the index.html page as a starting point.

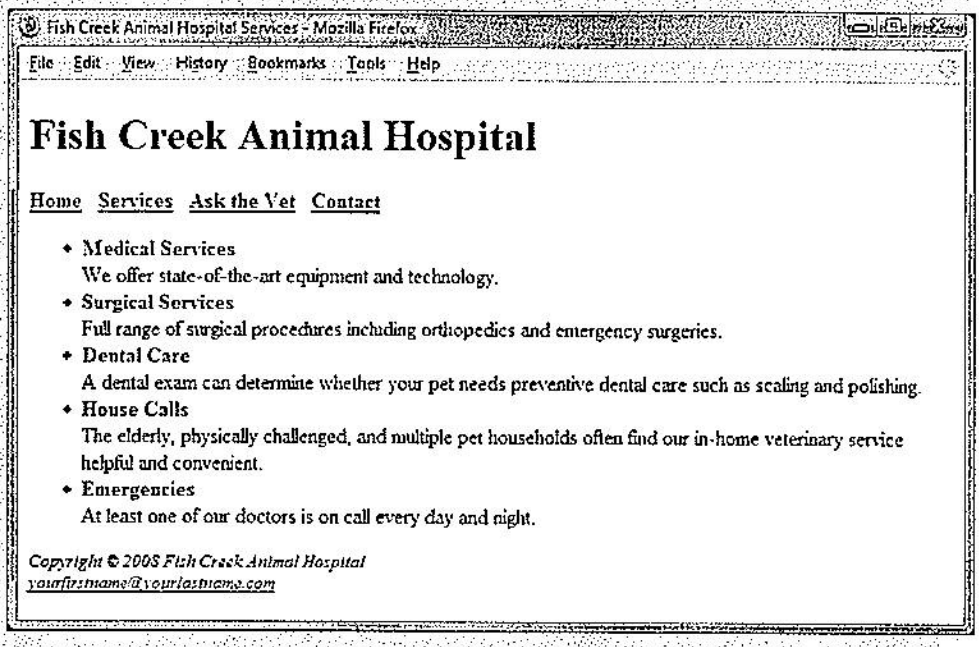


Figure 2.33
Fish Creek services.html

Open the index.html page for the Fish Creek Web site in Notepad. Select File, Save As, and save the file with the new name of services.html in the fishcreek folder. Now you are ready to edit the page.

- Modify the page title. Change the text contained between the `<title>` and `</title>` tags to Fish Creek Animal Hospital Services.
- Delete the definition list and the contact information.
- Add the services content to the page using an unordered list. Configure the name of each services category to be bold (use the `<strong>` logical style element). Code line breaks after each category name.
- The service categories and descriptions are as follows:

  Medical Services
  We offer state-of-the-art equipment and technology.

  Surgical Services
  Full range of surgical procedures including orthopedics and emergency surgeries.

  Dental Care
  A dental exam can determine whether your pet needs preventive dental care such as scaling and polishing.

  House Calls
  The elderly, physically challenged, and multiple pet households often find our in-home veterinary service helpful and convenient.

  Emergencies
  At least one of our doctors is on call every day and night.

Save your page and test it in a browser. Test the hyperlink from the services.html page to index.html. Test the hyperlink from the index.html page to services.html. If your links do not work, review your work with close attention to these details:

- Verify that you have saved the pages with the correct names in the correct folder.
- Verify your spelling of the page names in the anchor tags.
- After you make changes, test again.

Pete the Painter

Pete Johnson is an independent home painter and decorator. He would like to have a Web site to advertise his business. His clients are mainly homeowners in the middle-class suburbs of a large city. They range in age from thirties to fifties. Pete would like a site that contains a home page, a services page, a free estimates page, and a testimonial page.

A site map for the Pete the Painter Web site is shown in Figure 2.34.

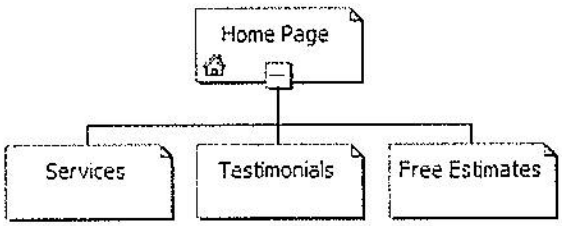


Figure 2.34
Pete the Painter site map

The site map describes the architecture of the Web site, a Home page with three main content pages: Services, Testimonials, and Free Estimates.

Figure 2.35 displays a sample page layout. It contains a site logo, a navigation area, a content area, and a footer area for copyright information.

You have two tasks in this case study:

1. Create the Home page: index.html.
2. Create the Services page: services.html.

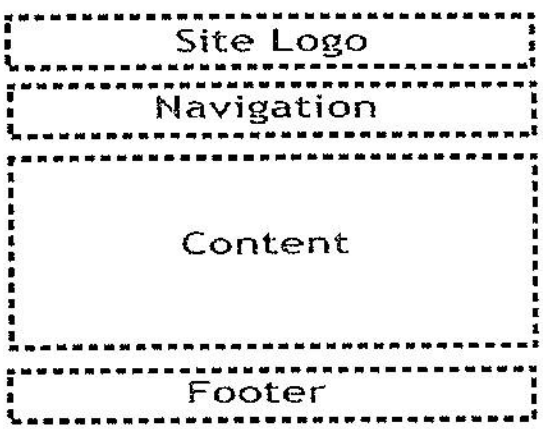


Figure 2.35
Pete the Painter page layout

Hands-On Practice Case

Create a folder called painter to contain your Pete the Painter Web site files.

1. **The Home Page.** You will use Notepad to create the Home page for the Pete the Painter Web site. The Home page is shown in Figure 2.36.

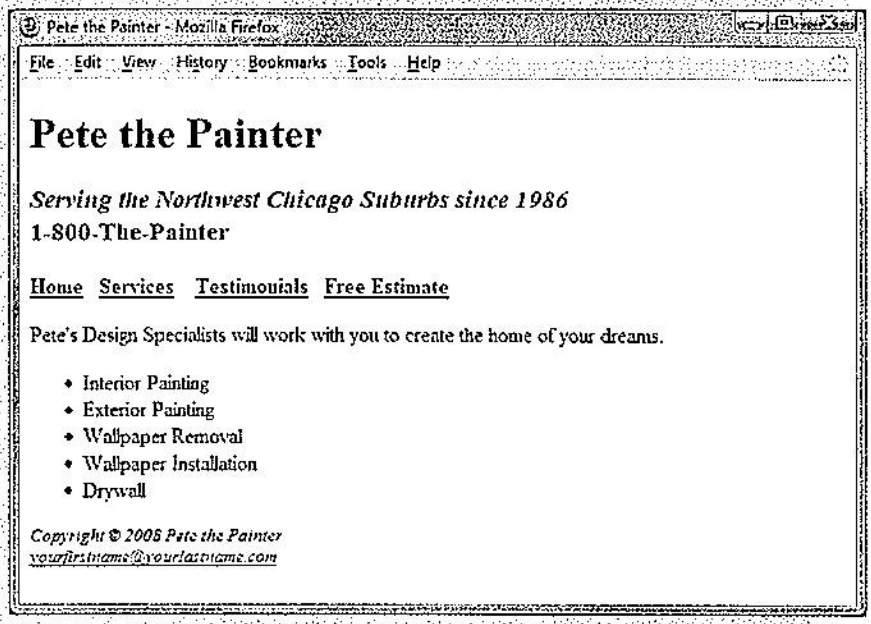


Figure 2.36
Pete the Painter index.html

Launch Notepad and create a Web page with the following specifications:

- Web page: Use a descriptive page title—the company name is a good choice for a business Web site.
- Logo area: Use `<h1>` for the Pete the Painter logo. The motto and phone number should be contained within an `<h3>` element. Emphasize the motto: Serving the Northwest Chicago Suburbs since 1986.
- Navigation: Place the following text using the `<strong>` logical style element within a paragraph:

  Home Services Testimonials Free Estimate

  Code anchor tags so that Home links to index.html, Services links to services.html, Testimonials links to testimonials.html, and Free Estimate links to estimate.html.
- Content: Place this sentence in a paragraph: Pete's Design Specialists will work with you to create the home of your dreams.

  Place the following content in an unordered list:

  Interior Painting
  Exterior Painting
  Wallpaper Removal
  Wallpaper Installation
  Drywall

- **Footer:** Place the following information in a small text size (use the <small> physical style element) and emphasized font style (use the logical style element):

  Copyright © 2008 Pete the Painter

  Place your name in an e-mail link on the line under the copyright information.

The page in Figure 2.36 may seem a little sparse, but don't worry, as you gain experience and learn to use more advanced techniques, your pages will look more professional. White space (blank space) on the page can be added with <br /> tags where needed. Your page does not need to look exactly the same as the sample. Your goal at this point should be to practice and get comfortable using XHTML.

Save your page in the painter folder and name it index.html.

2. **The Services Page.** Create the Services page shown in Figure 2.37. A productivity technique is to create new pages based on existing pages—so you can benefit from your previous work. Your new Services page will use the index.html page as a starting point.

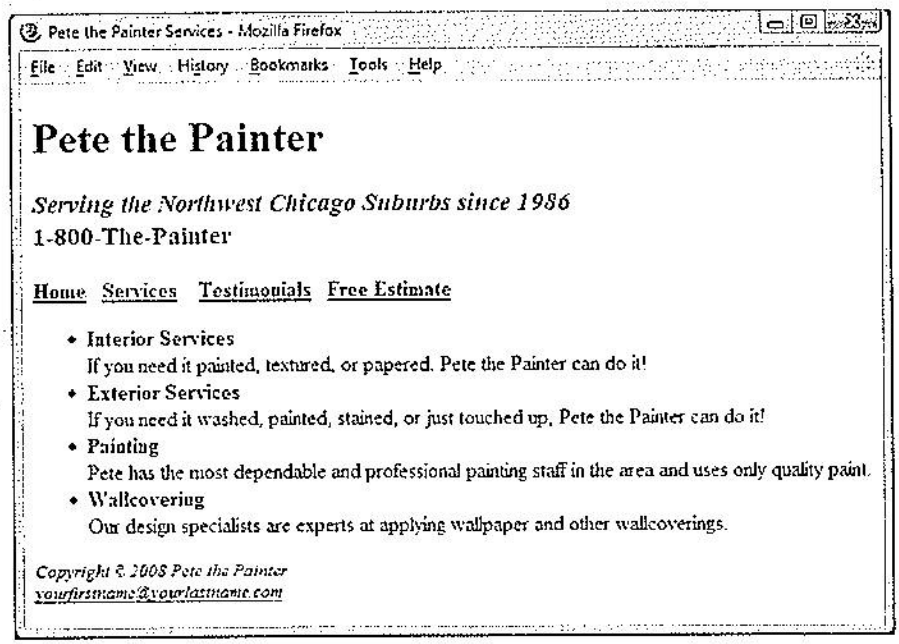


Figure 2.37
Pete the Painter services.html

Open the index.html page for the Pete the Painter Web site in Notepad. Select File, Save As, and save the file with the new name of services.html in the painter folder. Now you are ready to edit the page.

- Modify the page title. Change the text contained between the <title> and </title> tags to Pete the Painter Services.
- Delete the content paragraph and unordered list.
- Add the services content to the page using an unordered list. Configure the name of each services category to be bold (use the logical style element). Code line breaks after each category name.

- The service categories and descriptions are as follows:

  Interior Services
  If you need it painted, textured, or papered, Pete the Painter can do it!

  Exterior Services
  If you need it washed, painted, stained, or just touched up, Pete the Painter can do it!

  Painting
  Pete has the most dependable and professional painting staff in the area and uses only quality paint.

  Wallcovering
  Our design specialists are experts at applying wallpaper and other wallcoverings

Save your page and test it in a browser. Test the hyperlink from the services.html page to index.html. Test the hyperlink from the index.html page to services.html. If your links do not work, review your work with close attention to these details:

- Verify that you have saved the pages with the correct names in the correct folder.
- Verify your spelling of the page names in the anchor tags.
- After you make changes, test again.

Prime Properties

Prime Properties is a small real estate company that specializes in residential properties. The owner, Maria Valdez, would like a Web site to showcase her listings and provide a point of contact for her clients, who are mainly middle-class working adults who are looking for a home in the northwest Chicago suburbs. Maria would like a home page, a listings page that contains information about her properties, a financing page, and a contact page.

A site map for the Prime Properties Web site is shown in Figure 2.38.

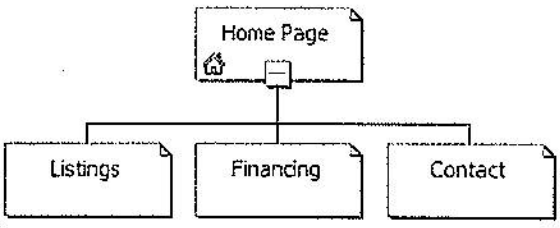


Figure 2.38
Prime Properties site map

The site map describes the architecture of the Web site, a Home page with three main content pages: Listings, Financing, and Contact.

Figure 2.39 displays a sample page layout. It contains a site logo, a navigation area, a content area, and a footer area for copyright information.

You have two tasks in this case study:

1. Create the Home page: index.html.
2. Create the Financing page: financing.html.

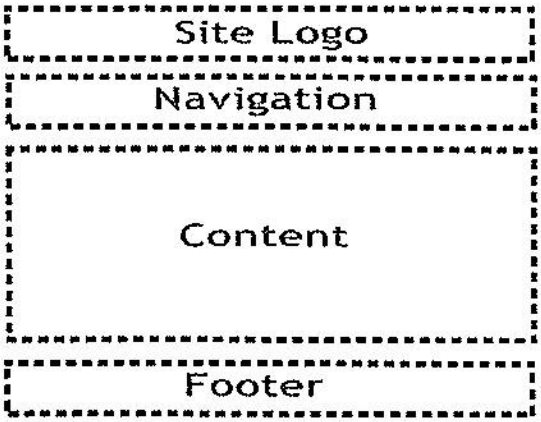


Figure 2.39
Prime Properties page layout

Hands-On Practice Case

Create a folder called prime to contain your Prime Properties Web page files.

1. **The Home Page.** You will use Notepad to create the Home page for the Prime Properties Web site. The Home page is shown in Figure 2.40.

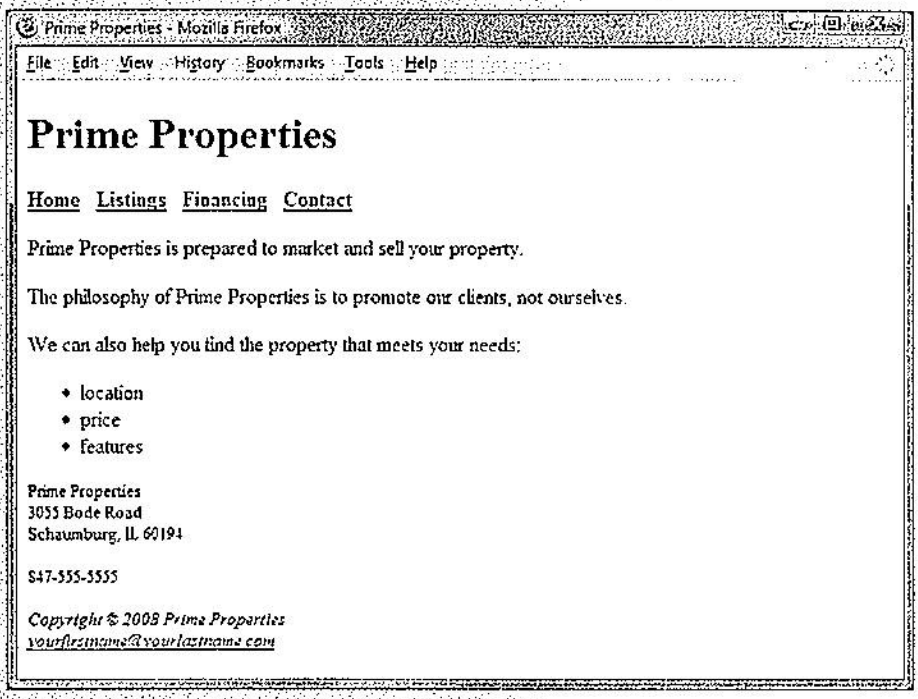


Figure 2.40
Prime Properties index.html

Launch Notepad and create a Web page with the following specifications:

- **Web page:** Use a descriptive page title—the company name is a good choice for a business Web site.
- **Logo area:** Use `<h1>` for the Prime Properties logo.
- **Navigation:** Configure the following text to display in bold font (use the `<strong>` logical style element) within a paragraph:

  Home Listings Financing Contact

Code anchor tags so that Home links to index.html, Services links to services.html, Financing links to financing.html, and Contact links to contact.html.

- Content: Place each line of text shown below in its own paragraph.

  Prime Properties is prepared to market and sell your property.

  The philosophy of Prime Properties is to promote our clients, not ourselves.

  We can also help you find the property that meets your needs:

  Place the following content in an unordered list:

  location

  price

  features

- Contact information: The address and phone information should display in small text (use the `<small>` physical style element).

  Prime Properties

  3055 Bode Road

  Schaumburg, IL 60194

  847-555-5555

- Footer: Place the following information in a small text size and emphasized font style (use `<small>` and `<em>` elements):

  Copyright © 2008 Prime Properties

  Place your name in an e-mail link on the line under the copyright information.

The page in Figure 2.40 may seem a little sparse, but don't worry, as you gain experience and learn to use more advanced techniques, your pages will look more professional. White space (blank space) on the page can be added with `<br />` tags where needed. Your page does not need to look exactly the same as the sample. Your goal at this point should be to practice and get comfortable using XHTML.

Save your page in the prime folder and name it index.html.

2. **The Financing Page.** Create the Financing page shown in Figure 2.41. A productivity technique is to create new pages based on existing pages—so you can benefit from your previous work. Your new Financing page will use the index.html page as a starting point.

   Open the index.html page for the Prime Properties Web site in Notepad. Select File, Save As, and save the file with the new name of financing.html in the prime folder. Now you are ready to edit the page.

   - Modify the page title. Change the text contained between the `<title>` and `</title>` tags to Prime Properties :: Financing.
   - Delete the content paragraphs, unordered list, and contact information.
   - Add the financing content to the page.

     First, configure the following text in an `<h2>` element: Financing.

     Next, place the following sentence in a paragraph:

     We work with many area mortgage and finance companies.

     Finally, configure the following phrase with an `<h4>` element:

     Mortgage FAQs

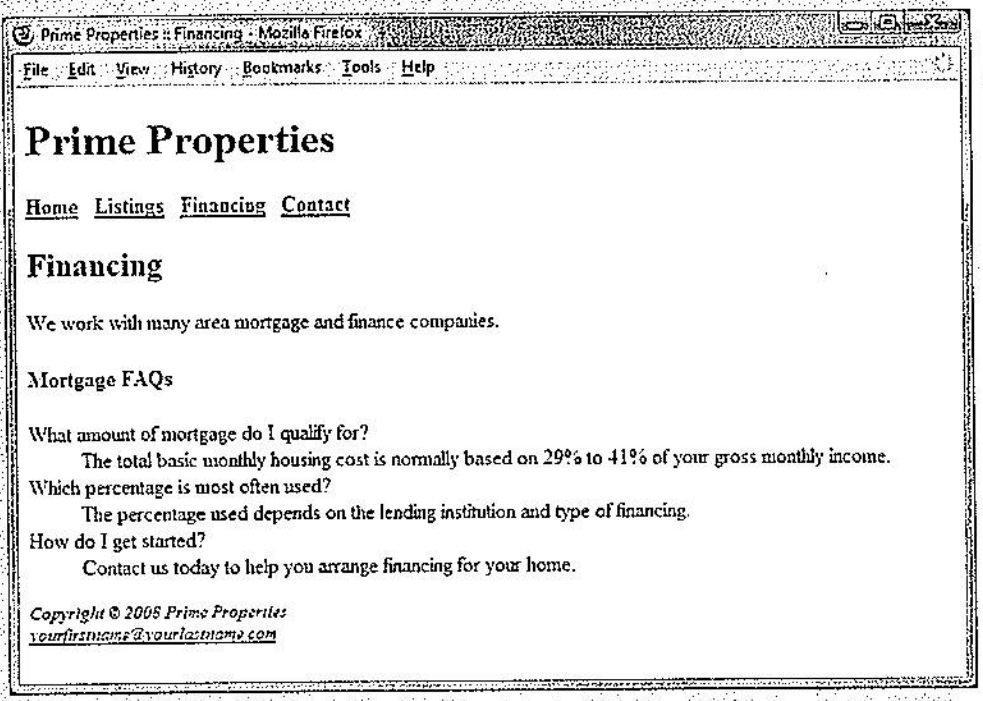


Figure 2.41
Prime Properties financing.html

Use a definition list to configure the FAQs. Use <dt> elements for the questions and <dd> elements for the answers. The FAQ questions and answers are as follows:

What amount of mortgage do I qualify for?

The total basic monthly housing cost is normally based on 29% to 41% of your gross monthly income.

Which percentage is most often used?

The percentage used depends on the lending institution and type of financing.

How do I get started?

Contact us today to help you arrange financing for your home.

Save your page and test it in a browser. Test the hyperlink from the financing.html page to index.html. Test the hyperlink from the index.html page to financing.html. If your links do not work, review your work with close attention to these details:

- Verify that you have saved the pages with the correct names in the correct folder.
- Verify your spelling of the page names in the anchor tags.
- After you make changes, test again.