

4

Introduction to XHTML: Part 1

Objectives

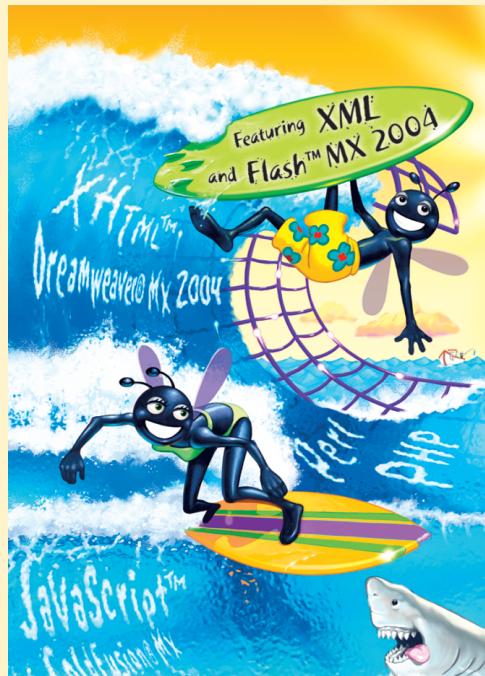
- To understand important components of XHTML documents.
- To use XHTML to create Web pages.
- To be able to add images to Web pages.
- To understand how to create and use hyperlinks to navigate Web pages.
- To be able to mark up lists of information.

To read between the lines was easier than to follow the text.

Henry James

High thoughts must have high language.

Aristophanes



Outline

-
- 4.1 Introduction
 - 4.2 Editing XHTML
 - 4.3 First XHTML Example
 - 4.4 W3C XHTML Validation Service
 - 4.5 Headers
 - 4.6 Linking
 - 4.7 Images
 - 4.8 Special Characters and More Line Breaks
 - 4.9 Unordered Lists
 - 4.10 Nested and Ordered Lists
 - 4.11 Web Resources

Summary • Terminology • Self-Review Exercises • Answers to Self-Review Exercises • Exercises

4.1 Introduction

Welcome to the world of opportunity created by the World Wide Web. The Internet is now three decades old, but it was not until the Web became popular in the 1990s that the explosion of opportunity that we are still experiencing began. Exciting new developments occur almost daily—the pace of innovation is unprecedented by any other technology. In this chapter, you will develop your own Web pages. As the book proceeds, you will create increasingly appealing and powerful Web pages. In the later portion of the book, you will learn how to create complete Web-based applications.

This chapter begins unlocking the power of Web-based application development with **XHTML**—the **Extensible HyperText Markup Language**. In later chapters, we introduce more sophisticated XHTML techniques, such as **tables**, which are particularly useful for structuring information from **databases** (i.e., software that stores structured sets of data), and **Cascading Style Sheets (CSS)**, which make Web pages more visually appealing.

Unlike procedural programming languages such as C, Fortran, Cobol and Pascal, **XHTML** is a **markup language** that specifies the format of the text that is displayed in a Web browser such as Microsoft’s *Internet Explorer* or *Netscape*.

One key issue when using XHTML is the separation of the **presentation** of a document (i.e., the document’s appearance when rendered by a browser) from the **structure** of the document’s information. XHTML is based on HTML (HyperText Markup Language)—a legacy technology of the World Wide Web Consortium (W3C). In HTML, it was common to specify the document’s content, structure and formatting. Formatting might specify where the browser placed an element in a Web page or the fonts and colors used to display an element. XHTML 1.1 (W3C’s latest version of W3C XHTML Recommendation at the time of publication) allows only a document’s content and structure to appear in a valid XHTML document, and not its formatting. Normally, such formatting is specified with Cascading Style Sheets (Chapter 6). All our examples in this chapter are based upon the XHTML 1.1 Recommendation.

4.2 Editing XHTML

In this chapter, we write XHTML in its **source-code form**. We create **XHTML documents** by typing them in a text editor (e.g., *Notepad*, *Wordpad*, *vi*, *emacs*) and saving them with either an **.html** or an **.htm** file-name extension.



Good Programming Practice 4.1

Assign documents file names that describe their functionality. This practice can help you identify documents faster. It also helps people who want to link to a page, by giving them an easy-to-remember name. For example, if you are writing an XHTML document that contains product information, you might want to call it products.html.

Machines running specialized software called **Web servers** store XHTML documents. Clients (e.g., Web browsers) request specific **resources** such as the XHTML documents from the Web server. For example, typing `www.deitel.com/books/downloads.html` into a Web browser's address field requests `downloads.html` from the Web server running at `www.deitel.com`. This document is located on the server in a directory named `books`. We discuss Web servers in detail in Chapter 21. For now, we simply place the XHTML documents on our machine and open them using Internet Explorer.

4.3 First XHTML Example

In this chapter and the next, we present XHTML markup and provide screen captures that show how Internet Explorer renders (i.e., displays) the XHTML.¹ Every XHTML document we show has line numbers for the reader's convenience. These line numbers are not part of the XHTML documents.

Our first example (Fig. 4.1) is an XHTML document named `main.html` that displays the message "Welcome to XHTML!" in the browser.

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
3   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
4
5 <!-- Fig. 4.1: main.html -->
6 <!-- Our first Web page -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>Internet and WWW How to Program - Welcome</title>
11   </head>
12
13  <body>
14    <p>Welcome to XHTML!</p>
15  </body>
16 </html>
```

Fig. 4.1 First XHTML example. (Part 1 of 2.)

1. All the examples presented in this book are available at www.deitel.com and on the CD-ROM that accompanies the book.

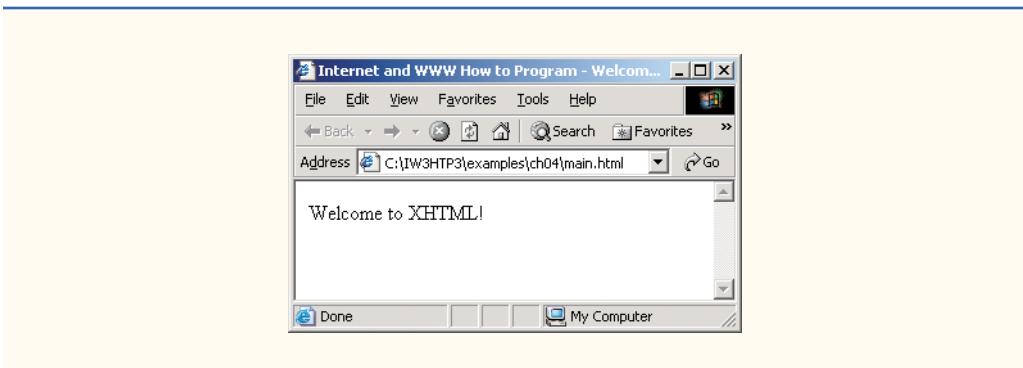


Fig. 4.1 First XHTML example. (Part 2 of 2.)

The key line in the program is line 14, which tells the browser to display “Welcome to XHTML!” Now let us consider each line of the program.

Lines 1–3 are required in XHTML documents to conform with proper XHTML syntax. For now, copy and paste these lines into each XHTML document you create. The meaning of these lines is discussed in detail in Chapter 20, Extensible Markup Language (XML).

Lines 5–6 are **XHTML comments**. XHTML document creators insert comments to improve markup readability and describe the content of a document. Comments also help other people read and understand an XHTML document’s markup and content. Comments do not cause the browser to perform any action when the user loads the XHTML document into the Web browser to view the document. XHTML comments always start with `<!--` and end with `-->`. Each of our XHTML examples includes comments that specify the figure number and file name, and provide a brief description of the example’s purpose. Subsequent examples include comments in the markup, especially to highlight new features.



Good Programming Practice 4.2

Place comments throughout your markup. Comments help other programmers understand the markup, assist in debugging and list useful information that you do not want the browser to render. Comments also help you understand your own markup when you revisit a document to modify or update it in the future.

XHTML markup contains text that represents the content of a document and **elements** that specify a document’s structure. Some important elements of an XHTML document are the **html** element, the **head** element and the **body** element. The **html** element encloses the **head section** (represented by the **head** element) and the **body section** (represented by the **body** element). The head section contains information about the XHTML document, such as its **title**. The head section also can contain special document formatting instructions called **style sheets** and client-side programs called **scripts** for creating dynamic Web pages. (We introduce style sheets in Chapter 6 and scripting with JavaScript in Chapter 7.) The body section contains the page’s content that the browser displays when the user visits the Web page.

XHTML documents delimit an element with **start** and **end** tags. A start tag consists of the element name in angle brackets (e.g., `<html>`). An end tag consists of the element name preceded by a / in angle brackets (e.g., `</html>`). In this example, lines 8 and 16 define the start and end of the **html** element. Note that the end tag in line 16 has the same name

as the start tag, but is preceded by a / inside the angle brackets. Many start tags have **attributes** that provide additional information about an element. Browsers can use this additional information to determine how to process the element. Each attribute has a **name** and a **value** separated by an equals sign (=). Line 8 specifies a required attribute (`xmlns`) and value (`http://www.w3.org/1999/xhtml`) for the `html` element in an XHTML document. For now, simply copy and paste the `html` element start tag in line 8 into your XHTML documents. We discuss the details of the `html` element's `xmlns` attribute in Chapter 20, Extensible Markup Language (XML).



Common Programming Error 4.1

Not enclosing attribute values in either single or double quotes is a syntax error. However, some Web browsers may still render the element correctly.



Common Programming Error 4.2

Using uppercase letters in an XHTML element or attribute name is a syntax error. However, some Web browsers may still render the element correctly.

An XHTML document divides the `html` element into two sections—head and body. Lines 9–11 define the Web page's head section with a `head` element. Line 10 specifies a `title` element. This is called a **nested element** because it is enclosed in the `head` element's start and end tags. The `head` element is also a nested element because it is enclosed in the `html` element's start and end tags. The `title` element describes the Web page. Titles usually appear in the **title bar** at the top of the browser window and also as the text identifying a page when users add the page to their list of **Favorites** or **Bookmarks** that enables them to return to their favorite sites. Search engines (i.e., sites that allow users to search the Web) also use the `title` for cataloging purposes.



Good Programming Practice 4.3

Indenting nested elements emphasizes a document's structure and promotes readability.



Common Programming Error 4.3

XHTML does not permit tags to overlap—a nested element's end tag must appear in the document before the enclosing element's end tag. For example, the nested XHTML tags `<head><title>hello</head></title>` cause a syntax error, because the enclosing `head` element's ending `</head>` tag appears before the nested `title` element's ending `</title>` tag.



Good Programming Practice 4.4

Use a consistent title-naming convention for all pages on a site. For example, if a site is named “Bailey’s Web Site,” then the `title` of the links page might be “Bailey’s Web Site—Links.” This practice can help users better understand the Web site’s structure.

Line 13 opens the document's body element. The body section of an XHTML document specifies the document's content, which may include text and tags.

Some tags, such as the **paragraph tags** (`<p>` and `</p>`) in line 14, mark up text for display in a browser. All the text placed between the `<p>` and `</p>` tags forms one paragraph. When the browser renders a paragraph, a blank line usually precedes and follows paragraph text.

This document ends with two end tags (lines 15–16). These tags close the body and `html` elements, respectively. The `</html>` tag in an XHTML document informs the browser that the XHTML markup is complete.

To view this example in Internet Explorer, perform the following steps:

1. Copy the Chapter 4 examples onto your machine from the CD that accompanies this book (or download the examples from www.deitel.com).
2. Launch Internet Explorer and select **Open...** from the **File** Menu. This displays the **Open** dialog.
3. Click the **Open** dialog's **Browse...** button to display the **Microsoft Internet Explorer** file dialog.
4. Navigate to the directory containing the Chapter 4 examples and select the file `main.html`, then click **Open**.
5. Click **OK** to have Internet Explorer render the document. Other examples are opened in a similar manner.

At this point your browser window should appear similar to the sample screen capture shown in Fig. 4.1. (Note that we resized the browser window to save space in the book.)

4.4 W3C XHTML Validation Service

Programming Web-based applications can be complex, and XHTML documents must be written correctly to ensure that browsers process them properly. To promote correctly written documents, the World Wide Web Consortium (W3C) provides a **validation service** (validator.w3.org) for checking a document's syntax. Documents can be validated either from a URL that specifies the location of the file or by uploading a file to the site validator.w3.org/file-upload.html. Uploading a file copies the file from the user's computer to another computer on the Internet. Figure 4.2 shows `main.html` (Fig. 4.1) being uploaded for validation. The W3C's Web page indicates that the service name is **MarkUp Validation Service**, and the validation service is able to validate the syntax of XHTML documents. All the XHTML examples in this book have been validated successfully using validator.w3.org.

By clicking **Browse...**, users can select files on their own computers for upload. After selecting a file, clicking the **Validate this file** button uploads and validates the file. Figure 4.3 shows the results of validating `main.html`. This document does not contain any syntax errors. If a document does contain syntax errors, the validation service displays error messages describing the errors. In Exercise 4.13, we ask the reader to create an invalid XHTML document (i.e., one that contains syntax errors) and check its syntax using the W3C validation service. This will enable the reader to see the types of error messages generated by the validator.



Error-Prevention Tip 4.1

Most current browsers attempt to render XHTML documents even if they are invalid. This often leads to unexpected and possibly undesirable results. Use a validation service, such as the W3C MarkUp Validation Service, to confirm that an XHTML document is syntactically correct.

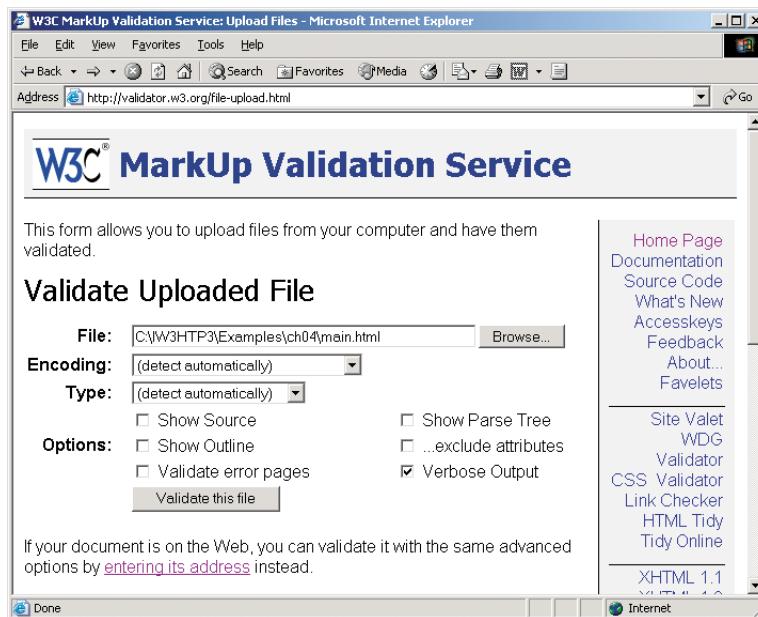


Fig. 4.2 Validating an XHTML document. (Courtesy of World Wide Web Consortium (W3C).)

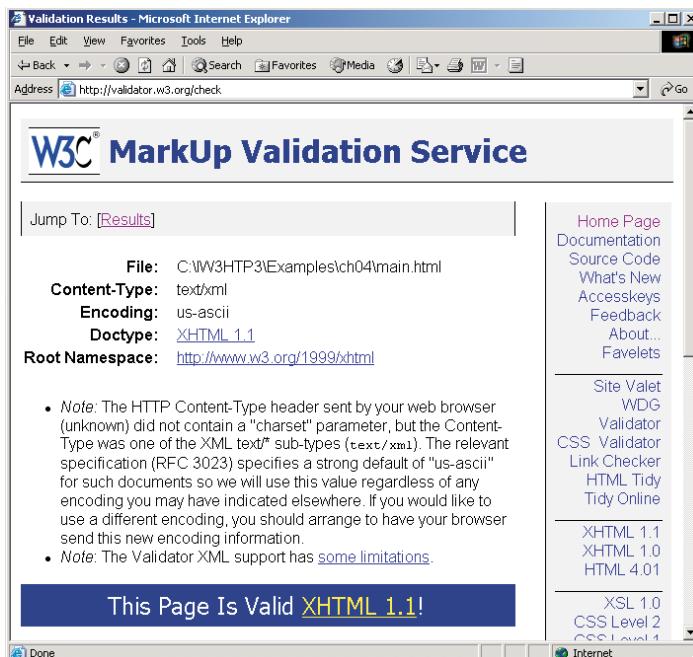


Fig. 4.3 XHTML validation results. (Courtesy of World Wide Web Consortium (W3C).)

4.5 Headers

Some text in an XHTML document may be more important than other text. For example, the text in this section is considered more important than a footnote. XHTML provides six **headers**, called **header elements**, for specifying the relative importance of information. Figure 4.4 demonstrates these elements (h1 through h6). Header element h1 (line 15) is considered the most significant header and is typically rendered in a larger font than the other five headers (lines 16–20). Each successive header element (i.e., h2, h3, etc.) is typically rendered in a progressively smaller font.

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
3   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
4
5 <!-- Fig. 4.4: header.html -->
6 <!-- XHTML headers -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>Internet and WWW How to Program - Headers</title>
11   </head>
12
13   <body>
14
15    <h1>Level 1 Header</h1>
16    <h2>Level 2 header</h2>
17    <h3>Level 3 header</h3>
18    <h4>Level 4 header</h4>
19    <h5>Level 5 header</h5>
20    <h6>Level 6 header</h6>
21
22   </body>
23 </html>
```

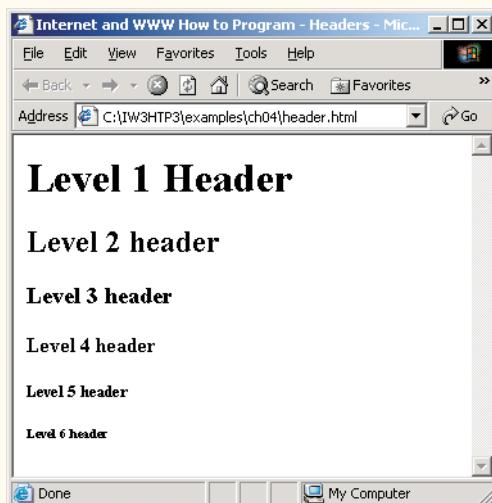


Fig. 4.4 Header elements h1 through h6.



Portability Tip 4.1

The text size used to display each header element can vary significantly between browsers. In Chapter 6, we discuss how to control the text size and other text properties.



Look-and-Feel Observation 4.1

Placing a header at the top of every XHTML page helps viewers understand the purpose of each page.



Look-and-Feel Observation 4.2

Use larger headers to emphasize more important sections of a Web page.

4.6 Linking

One of the most important XHTML features is the **hyperlink**, which references (or **links** to) other resources, such as XHTML documents and images. In XHTML, both text and images can act as hyperlinks. Web browsers typically underline text hyperlinks and color their text blue by default, so that users can distinguish hyperlinks from plain text. In Fig. 4.5, we create text hyperlinks to four different Web sites.

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
3   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
4
5 <!-- Fig. 4.5: links.html          -->
6 <!-- Introduction to hyperlinks -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>Internet and WWW How to Program - Links</title>
11   </head>
12
13   <body>
14
15     <h1>Here are my favorite sites</h1>
16
17     <p><strong>Click a name to go to that page.</strong></p>
18
19     <!-- Create four text hyperlinks -->
20     <p><a href = "http://www.deitel.com">Deitel</a></p>
21
22     <p><a href = "http://www.prenhall.com">Prentice Hall</a></p>
23
24     <p><a href = "http://www.yahoo.com">Yahoo!</a></p>
25
26     <p><a href = "http://www.usatoday.com">USA Today</a></p>
27
28   </body>
29 </html>
```

Fig. 4.5 Linking to other Web pages. (Part 1 of 2.)

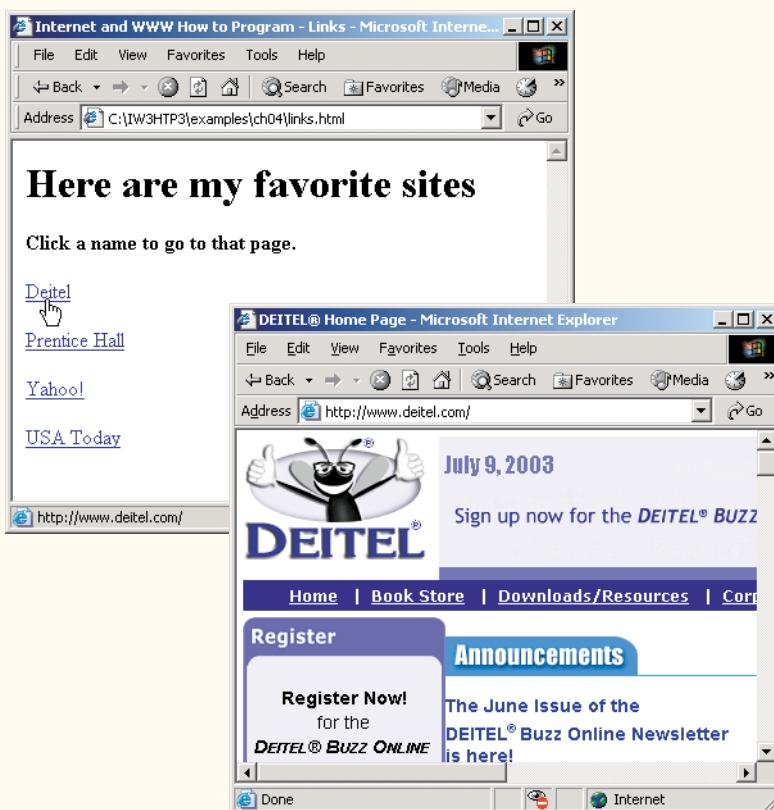


Fig. 4.5 Linking to other Web pages. (Part 2 of 2.)

Line 17 introduces the **strong** element. Browsers typically display such text in a bold font.

Links are created using the **a** (anchor) element. Line 20 defines a hyperlink that links the text **Deitel** to the URL assigned to attribute **href**, which specifies the location of a linked resource, such as a Web page, a file or an e-mail address. This particular anchor element links to a Web page located at <http://www.deitel.com>. When a URL does not indicate a specific document on the Web site, the Web server returns a default Web page. This page is often called `index.html`; however, most Web servers can be configured to use any file as the default Web page for the site. (Open <http://www.deitel.com> in one browser window and <http://www.deitel.com/index.html> in a second browser window to confirm that they are identical.) If the Web server cannot locate a requested document, it returns an error indication to the Web browser, and the browser displays a Web page containing an error message to the user.

Anchors can link to e-mail addresses using a **mailto:** URL. When someone clicks this type of anchored link, most browsers launch the default e-mail program (e.g., Outlook Express) to enable the user to write an e-mail message to the linked address. Figure 4.6 demonstrates this type of anchor. Lines 17–19 contain an e-mail link. The form of an e-mail

anchor is `...`. In this case, we link to the e-mail address `deitel@deitel.com`.

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
3   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
4
5 <!-- Fig. 4.6: contact.html -->
6 <!-- Adding email hyperlinks -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>Internet and WWW How to Program - Contact Page</title>
11   </head>
12
13   <body>
14
15     <p>
16       My e-mail address is
17       <a href = "mailto:deitel@deitel.com">
18         deitel@deitel.com
19       </a>
20       . Click the address and your browser will
21       open an e-mail message and address it to me.
22     </p>
23   </body>
24 </html>
```

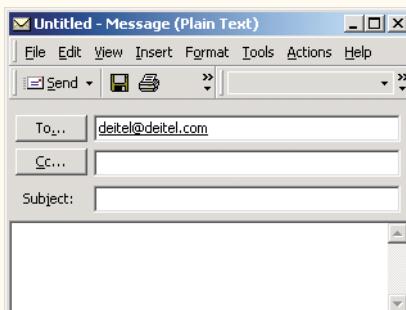
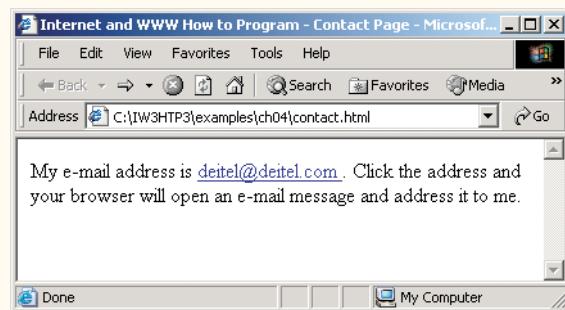


Fig. 4.6 Linking to an e-mail address.

4.7 Images

The examples discussed so far demonstrate how to mark up documents that contain only text. However, most Web pages contain both text and images. In fact, images are an equal, if not essential, part of Web-page design. The three most popular image formats used by Web developers are Graphics Interchange Format (GIF), Joint Photographic Experts Group (JPEG) and Portable Network Graphics (PNG) images. Users can create images using specialized pieces of software, such as Adobe Photoshop Elements 2.0 (discussed in Chapter 3), Macromedia Fireworks (www.macromedia.com) and Jasc Paint Shop Pro (www.jasc.com). Images may also be acquired from various Web sites, such as the Yahoo! Picture Gallery (gallery.yahoo.com). Figure 4.7 demonstrates how to incorporate images into Web pages.

Lines 16–17 use an **img** element to insert an image in the document. The image file's location is specified with the **img** element's **src** attribute. In this case, the image is located in the same directory as this XHTML document, so only the image's file name is required. Optional attributes **width** and **height** specify the image's width and height, respectively. The document author can scale an image by increasing or decreasing the values of the image **width** and **height** attributes. If these attributes are omitted, the browser uses the image's actual width and height. Images are measured in pixels ("picture elements"), which represent dots of color on the screen. The image in Fig. 4.7 is 183 pixels wide and 238 pixels high.



Good Programming Practice 4.5

Always include the **width** and the **height** of an image inside the **** tag. When the browser loads the XHTML file, it will know immediately from these attributes how much screen space to provide for the image and will lay out the page properly, even before it downloads the image.

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
3   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
4
5 <!-- Fig. 4.7: picture.html -->
6 <!-- Adding images with XHTML -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>Internet and WWW How to Program - Welcome</title>
11   </head>
12
13   <body>
14
15     <p>
16       <img src = "xmlhttp.jpg" height = "238" width = "183"
17         alt = "XML How to Program book cover" />
18       <img src = "jhttp.jpg" height = "238" width = "183"
19         alt = "Java How to Program book cover" />
20     </p>
21   </body>
22 </html>
```

Fig. 4.7 Images in XHTML files. (Part 1 of 2.)

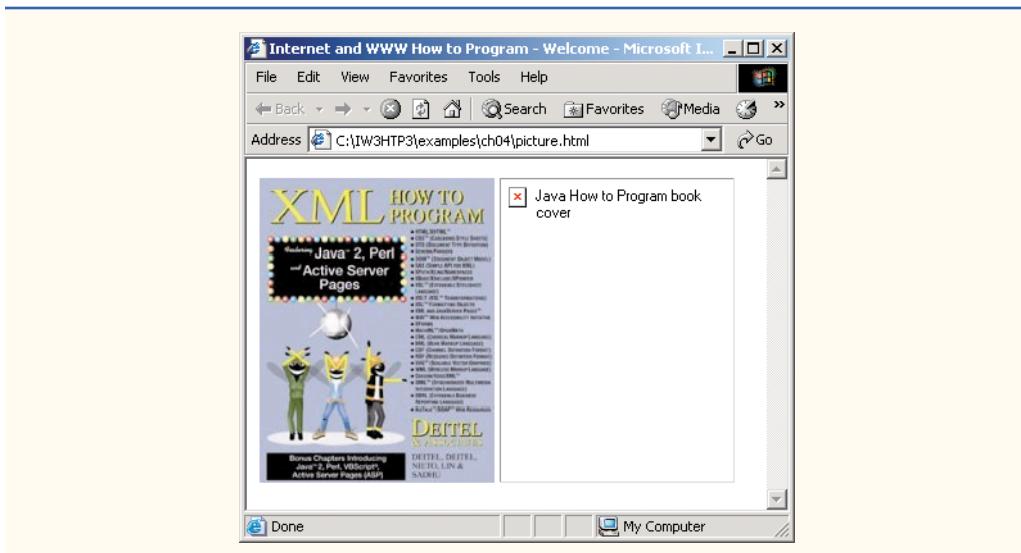


Fig. 4.7 Images in XHTML files. (Part 2 of 2.)



Performance Tip 4.1

Including the `width` and `height` attributes in an `` tag can result in the browser loading and rendering pages faster.



Common Programming Error 4.4

Entering new dimensions for an image that change its inherent width-to-height ratio distorts the appearance of the image. For example, if your image is 200 pixels wide and 100 pixels high, you should ensure that any new dimensions have a 2:1 width-to-height ratio.

Every `img` element in an XHTML document has an `alt` attribute. If a browser cannot render an image, the browser displays the `alt` attribute's value. A browser may not be able to render an image for several reasons. It may not support images—as is the case with a **text-based browser** (i.e., a browser that can display only text)—or the client may have disabled image viewing to reduce download time. Figure 4.7 shows Internet Explorer 6 rendering the `alt` attribute's value when a document references a nonexistent image file (`jhttp.jpg`).

The `alt` attribute is important for creating **accessible** Web pages for users with disabilities, especially those with vision impairments who use text-based browsers. Specialized software called a **speech synthesizer** often is used by people with disabilities. This software application “speaks” the `alt` attribute's value so that the user knows what the browser is displaying. We discuss accessibility issues in detail in Chapter 29.

Some XHTML elements (called **empty elements**) contain only attributes and do not mark up text (i.e., text is not placed between the start and end tags). Empty elements (e.g., `img`) must be terminated, either by using the **forward slash character** (/) inside the closing right angle bracket (>) of the start tag or by explicitly including the end tag. When using the forward slash character, we add a space before the forward slash to improve readability (as shown at the ends of lines 17 and 19). Rather than using the forward slash character, lines 18–19 could be written with a closing `` tag as follows:

```
<img src = "jhtp.jpg" height = "238" width = "183"
     alt = "Java How to Program book cover"></img>
```

By using images as hyperlinks, Web developers can create graphical Web pages that link to other resources. In Fig. 4.8, we create six different image hyperlinks.

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
3   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
4
5 <!-- Fig. 4.8: nav.html           -->
6 <!-- Using images as link anchors -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>Internet and WWW How to Program - Navigation Bar
11    </title>
12   </head>
13
14   <body>
15
16     <p>
17       <a href = "links.html">
18         <img src = "buttons/links.jpg" width = "65"
19           height = "50" alt = "Links Page" />
20       </a><br />
21
22       <a href = "list.html">
23         <img src = "buttons/list.jpg" width = "65"
24           height = "50" alt = "List Example Page" />
25       </a><br />
26
27       <a href = "contact.html">
28         <img src = "buttons/contact.jpg" width = "65"
29           height = "50" alt = "Contact Page" />
30       </a><br />
31
32       <a href = "header.html">
33         <img src = "buttons/header.jpg" width = "65"
34           height = "50" alt = "Header Page" />
35       </a><br />
36
37       <a href = "table1.html">
38         <img src = "buttons/table.jpg" width = "65"
39           height = "50" alt = "Table Page" />
40       </a><br />
41
42       <a href = "form.html">
43         <img src = "buttons/form.jpg" width = "65"
44           height = "50" alt = "Feedback Form" />
45       </a><br />
46     </p>
```

Fig. 4.8 Images as link anchors. (Part 1 of 2.)

```
47
48      </body>
49  </html>
```

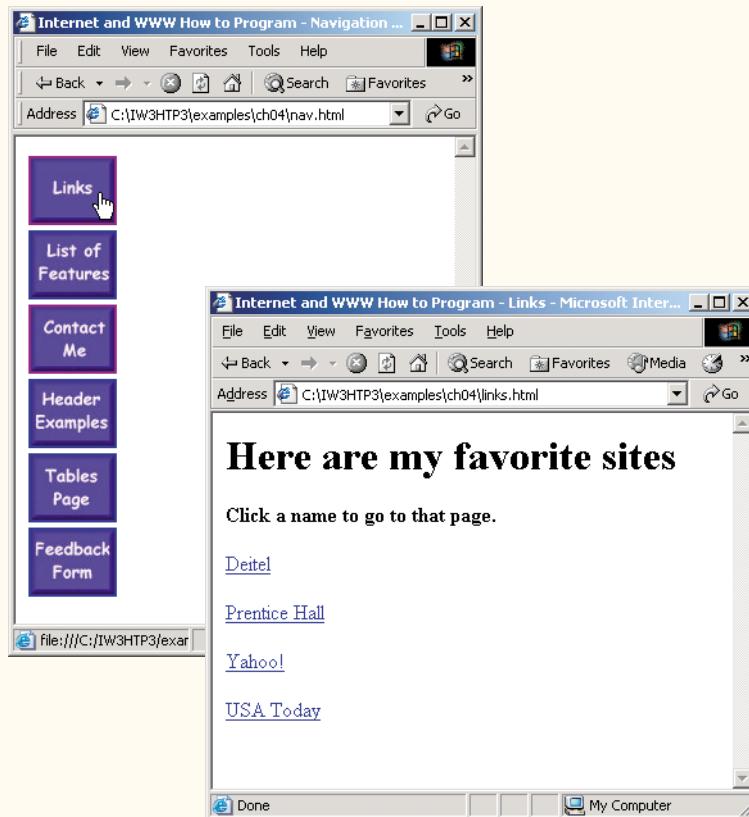


Fig. 4.8 Images as link anchors. (Part 2 of 2.)

Lines 17–20 create an **image hyperlink** by nesting an `img` element in an anchor (`a`) element. The value of the `img` element's `src` attribute value specifies that this image (`links.jpg`) resides in a directory named `buttons`. The `buttons` directory and the XHTML document are in the same directory. Images from other Web documents also can be referenced (after obtaining permission from the document's owner) by setting the `src` attribute to the name and location of the image. Clicking an image hyperlink takes a user to the Web page specified by the surrounding anchor element's `href` attribute.

In line 20, we introduce the **`br`** element, which most browsers render as a **line break**. Any markup or text following a `br` element is rendered on the next line. Like the `img` element, `br` is an example of an empty element terminated with a forward slash. We add a space before the forward slash to enhance readability. [Note: The last two image hyperlinks in Fig. 4.8 link to XHTML documents (i.e., `table1.html` and `form.html`) presented as examples in Chapter 5 and included in the Chapter 5 examples directory. Clicking these links now will result in errors.]

4.8 Special Characters and More Line Breaks

When marking up text, certain characters or symbols (e.g., <) may be difficult to embed directly into an XHTML document. Some keyboards do not provide these symbols, or the presence of these symbols may cause syntax errors. For example, the markup

```
<p>if x < 10 then increment x by 1</p>
```

results in a syntax error because it uses the less-than character (<), which is reserved for start tags and end tags such as <p> and </p>. XHTML provides **character entity references** (in the form &code;) for representing special characters. We could correct the previous line by writing

```
<p>if x &lt; 10 then increment x by 1</p>
```

which uses the character entity reference < for the less-than symbol.

Figure 4.9 demonstrates how to use special characters in an XHTML document. For a list of special characters, see Appendix A, XHTML Special Characters.

```
1  <?xml version = "1.0"?>
2  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
3      "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
4
5  <!-- Fig. 4.9: contact2.html      -->
6  <!-- Inserting special characters -->
7
8  <html xmlns = "http://www.w3.org/1999/xhtml">
9      <head>
10         <title>Internet and WWW How to Program - Contact Page
11         </title>
12     </head>
13
14     <body>
15
16         <!-- special characters are entered -->
17         <!-- using the form &code;      -->
18         <p>
19             Click
20             <a href = "mailto:deitel@deitel.com">here</a>
21             to open an e-mail message addressed to
22             deitel@deitel.com.
23         </p>
24
25         <hr /> <!-- inserts a horizontal rule -->
26
27         <p>All information on this site is <strong>&copy;.</strong>
28             Deitel <strong>&amp;lt;</strong> Associates, Inc. 2004.</p>
29
30         <!-- to strike through text use <del> tags    -->
31         <!-- to subscript text use <sub> tags        -->
32         <!-- to superscript text use <sup> tags       -->
33         <!-- these tags are nested inside other tags -->
```

Fig. 4.9 Special characters in XHTML. (Part 1 of 2.)

```
34      <p><del>You may download  $3.14 \times 10^{22}$ </sup></del>
35          characters worth of information from this site.</del>
36          Only <sub>one</sub> download per hour is permitted.</p>
37
38      <p>Note: <strong>&lt; &frac14;</strong> of the information
39          presented here is updated daily.</p>
40
41      </body>
42  </html>
```

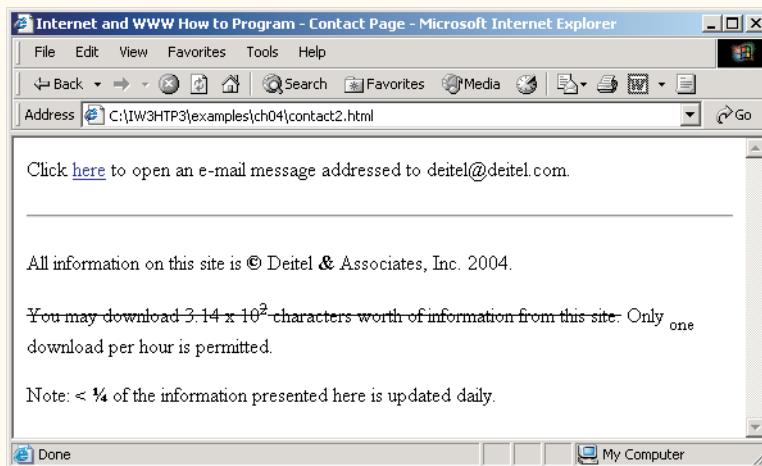


Fig. 4.9 Special characters in XHTML. (Part 2 of 2.)

Lines 27–28 contain other special characters, which can be expressed as either character entity references (i.e., word abbreviations such as `amp` for ampersand and `copy` for copyright) or **numeric character references**—decimal or **hexadecimal** (hex) values representing special characters. For example, the & character is represented in decimal and hexadecimal notation as `&` and `&`, respectively. Hexadecimal numbers are base 16 numbers—digits in a hexadecimal number have values from 0 to 15 (a total of 16 different values). The letters A–F represent the hexadecimal digits corresponding to decimal values 10–15. Thus in hexadecimal notation we can have numbers like 876 consisting solely of decimal-like digits, numbers like DA19F consisting of digits and letters and numbers like DCB consisting solely of letters. We discuss hexadecimal numbers in detail in Appendix E, Number Systems.

In lines 34–36, we introduce three new elements. Most browsers render the `del` element as strike-through text. With this format users can easily indicate document revisions. To **superscript** text (i.e., raise text on a line with a decreased font size) or **subscript** text (i.e., lower text on a line with a decreased font size), use the `sup` or `sub` element, respectively. We also use character entity reference `<` for a less-than sign and `¼` for the fraction 1/4 (line 38).

In addition to special characters, this document introduces a **horizontal rule**, indicated by the `<hr />` tag in line 25. Most browsers render a horizontal rule as a horizontal line. The `<hr />` tag also inserts a line break above and below the horizontal line.

4.9 Unordered Lists

Up to this point, we have presented basic XHTML elements and attributes for linking to resources, creating headers, using special characters and incorporating images. In this section, we discuss how to organize information on a Web page using lists. In Chapter 5, we introduce another feature for organizing information, called a table. Figure 4.10 displays text in an **unordered list** (i.e., a list that does not order its items by letter or number). The unordered list element `ul` creates a list in which each item begins with a bullet symbol (called a **disc**). Each entry in an unordered list (element `ul` in line 20) is an `li` (**list item**) element (lines 23, 25, 27 and 29). Most Web browsers render these elements with a line break and a bullet symbol indented from the beginning of the new line.

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
3   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
4
5 <!-- Fig. 4.10: links2.html -->
6 <!-- Unordered list containing hyperlinks -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>Internet and WWW How to Program - Links</title>
11   </head>
12
13  <body>
14
15    <h1>Here are my favorite sites</h1>
16
17    <p><strong>Click on a name to go to that page.</strong></p>
18
19    <!-- create an unordered list -->
20    <ul>
21
22      <!-- add four list items -->
23      <li><a href = "http://www.deitel.com">Deitel</a></li>
24
25      <li><a href = "http://www.w3.org">W3C</a></li>
26
27      <li><a href = "http://www.yahoo.com">Yahoo!</a></li>
28
29      <li><a href = "http://www.cnn.com">CNN</a></li>
30
31    </ul>
32  </body>
33 </html>
```

Fig. 4.10 Unordered lists in XHTML. (Part 1 of 2.)

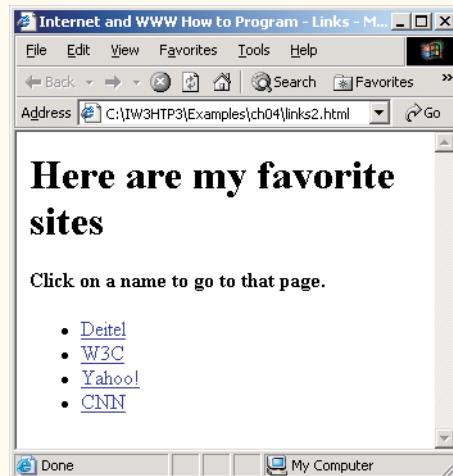


Fig. 4.10 Unordered lists in XHTML. (Part 2 of 2.)

4.10 Nested and Ordered Lists

Lists may be nested to represent hierarchical relationships, as in an outline format. Figure 4.11 demonstrates nested lists and **ordered lists**. The ordered list element **ol** creates a list in which each item begins with a number.

A Web browser indents each nested list to indicate a hierarchical relationship. The first ordered list begins at line 33. Items in an ordered list are enumerated one, two, three and so on. Nested ordered lists are enumerated in the same manner. The items in the outermost unordered list (line 18) are preceded by discs. List items nested inside the unordered list of line 18 are preceded by circles. Although not demonstrated in this example, subsequent nested list items are preceded by squares.

```
1 <?xml version = "1.0"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
3   "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
4
5 <!-- Fig. 4.11: list.html           -->
6 <!-- Advanced Lists: nested and ordered -->
7
8 <html xmlns = "http://www.w3.org/1999/xhtml">
9   <head>
10    <title>Internet and WWW How to Program - Lists</title>
11   </head>
12
13   <body>
```

Fig. 4.11 Nested and ordered lists in XHTML. (Part 1 of 3.)

```
14
15      <h1>The Best Features of the Internet</h1>
16
17      <!-- create an unordered list -->
18      <ul>
19          <li>You can meet new people from countries around
20              the world.</li>
21          <li>
22              You have access to new media as it becomes public:
23
24              <!-- this starts a nested list, which uses a -->
25              <!-- modified bullet. The list ends when you -->
26              <!-- close the <ul> tag. -->
27          <ul>
28              <li>New games</li>
29              <li>
30                  New applications
31
32                  <!-- nested ordered list -->
33                  <ol>
34                      <li>For business</li>
35                      <li>For pleasure</li>
36                  </ol>
37              </li>
38
39          <li>Around the clock news</li>
40          <li>Search engines</li>
41          <li>Shopping</li>
42          <li>
43              Programming
44
45          <!-- another nested ordered list -->
46          <ol>
47              <li>XML</li>
48              <li>Java</li>
49              <li>XHTML</li>
50              <li>Scripts</li>
51              <li>New languages</li>
52          </ol>
53
54      </li>
55
56      </ul> <!-- ends the nested list of line 27 -->
57  </li>
58
59  <li>Links</li>
60  <li>Keeping in touch with old friends</li>
61  <li>It is the technology of the future!</li>
62
63  </ul>  <!-- ends the unordered list of line 18 -->
64
65  </body>
66  </html>
```

Fig. 4.11 Nested and ordered lists in XHTML. (Part 2 of 3.)

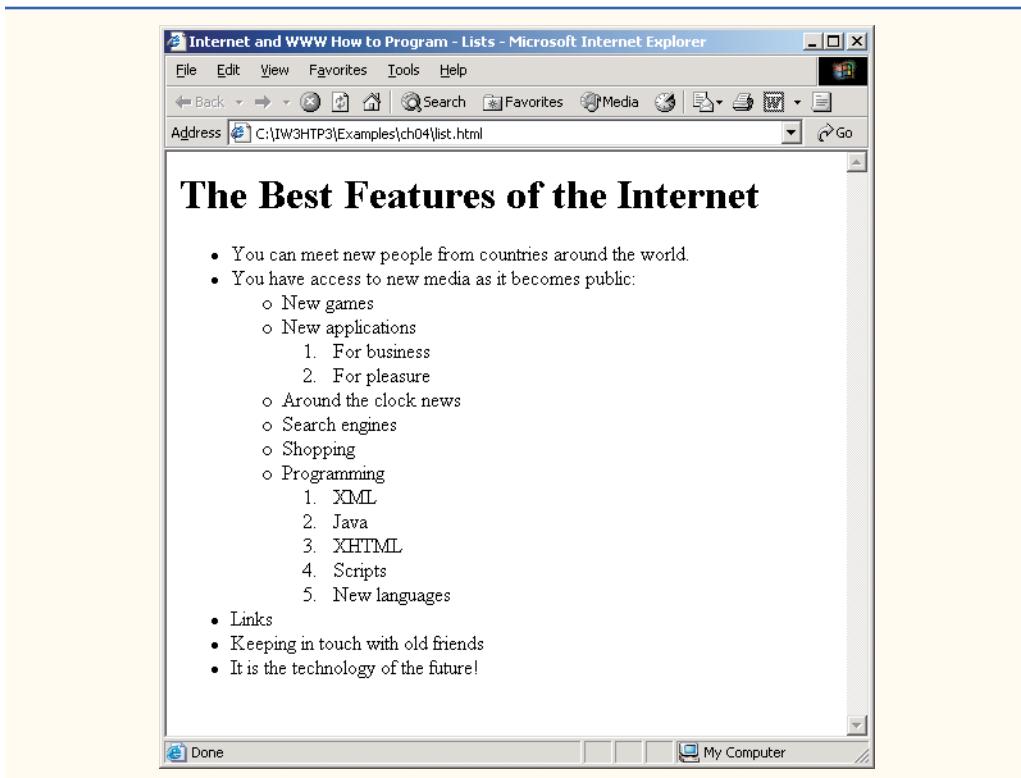


Fig. 4.11 Nested and ordered lists in XHTML. (Part 3 of 3.)

4.11 Web Resources

www.w3.org/TR/xhtml11

The *XHTML 1.1 Recommendation* contains XHTML 1.1 general information, compatibility issues, document type definition information, definitions, terminology and much more.

www.xhtml.org

XHTML.org provides XHTML development news and links to other XHTML resources, including books and articles.

www.w3schools.com/xhtml/default.asp

The *XHTML School* provides XHTML quizzes and references. This page also contains links to XHTML syntax, validation and document type definitions.

validator.w3.org

This is the W3C XHTML validation service site.

hotwired.lycos.com/webmonkey/00/50/index2a.html

This site provides an article about XHTML. Key sections of the article overview XHTML and discuss tags, attributes and anchors.

wdvl.com/Authoring/Languages/XML/XHTML

The *Web Developers Virtual Library* provides an introduction to XHTML. This site also contains articles, examples and links to other technologies.

www.w3.org/TR/2001/REC-xhtml11-20010531

The XHTML 1.1 DTD documentation site provides technical specifications of XHTML 1.1 syntax.

SUMMARY

- XHTML (Extensible HyperText Markup Language) is a markup language for creating Web pages.
- A key issue when using XHTML is the separation of the presentation of a document (i.e., the document's appearance when rendered by a browser) from the structure of the information in the document.
- In XHTML, text is marked up with elements delimited by tags that are names contained in pairs of angle brackets. Some elements may contain attributes that provide additional information about the element.
- A machine that runs a specialized piece of software called a Web server stores XHTML documents.
- XHTML documents that are syntactically correct are guaranteed to render properly. XHTML documents that contain syntax errors may not display properly.
- Validation services (e.g., validator.w3.org) ensure that an XHTML document is syntactically correct.
- Every XHTML document contains a start `<html>` tag and an end `</html>` tag.
- Comments in XHTML always begin with `<!--` and end with `-->`. The browser ignores all text inside a comment.
- Every XHTML document contains a `head` element, which generally contains information, such as a title, and a `body` element, which contains the page content. Information in the `head` element generally is not rendered in the display window but may be made available to the user through other means.
- The `title` element names a Web page. The title usually appears in the colored bar (called the title bar) at the top of the browser window and also appears as the text identifying a page when users add your page to their list of **Favorites** or **Bookmarks**.
- The `body` of an XHTML document is the area in which the document's content is placed. The content may include text and tags.
- All text placed between the `<p>` and `</p>` tags forms one paragraph.
- XHTML provides six headers (`h1` through `h6`) for specifying the relative importance of information. Header element `h1` is considered the most significant header and is rendered in a larger font than the other five headers. Each successive header element (i.e., `h2`, `h3`, etc.) is rendered in a progressively smaller font.
- Web browsers typically underline text hyperlinks and color them blue by default.
- The `strong` element typically causes the browser to render text in a bold font.
- Users can insert links with the `a` (anchor) element. The most important attribute for the `a` element is `href`, which specifies the resource (e.g., page, file, e-mail address) being linked.
- Anchors can link to an e-mail address using a `mailto:` URL. When someone clicks this type of anchored link, most browsers launch the default e-mail program (e.g., Outlook Express) to initiate an e-mail message addressed to the linked address.
- The `img` element's `src` attribute specifies an image's location. Optional attributes `width` and `height` specify the image width and height, respectively. Images are measured in pixels, which represent dots of color on the screen. Every `img` element in a valid XHTML document must have an `alt` attribute, which contains text that is displayed if the client cannot render the image.
- The `alt` attribute makes Web pages more accessible to users with disabilities, especially those with vision impairments.
- Some XHTML elements are empty elements that contain only attributes and do not mark up text. Empty elements (e.g., `img`) must be terminated, either by using the forward slash character (/) or by explicitly writing an end tag.

- The **br** element causes most browsers to render a line break. Any markup or text following a **br** element is rendered on the next line.
- XHTML provides special characters or entity references (in the form &code;) for representing characters that cannot be rendered otherwise.
- Most browsers render a horizontal rule, indicated by the **<hr />** tag, as a horizontal line. The **hr** element also inserts a line break above and below the horizontal line.
- The unordered list element **ul** creates a list in which each item in the list begins with a bullet symbol (called a disc). Each entry in an unordered list is a **li** (list item) element. Most Web browsers render these elements with a line break and a bullet symbol at the beginning of the line.
- The ordered list element **ol** creates a list in which each item begins with a number.
- Lists may be nested to represent hierarchical data relationships.

TERMINOLOGY

<!----> (XHTML comment)	li element (list item)
a element (<a>...)	linked document
alt attribute (img)	mailto: URL
&# (special character)	markup language
anchor	nested list
angle brackets (< >)	numeric character reference
attribute	ol element (ordered list)
body element	p element (paragraph)
br element (line break)	special character
character entity reference	src attribute (img)
comment in XHTML	strong element
© (© special character)	sub element
disc	subscript
element	superscript
e-mail anchor	syntax
empty tag	tag
Extensible HyperText Markup Language (XHTML)	text editor
head element	text-based browser
header	title element
h1 through h6 (header elements)	ul element (unordered list)
height attribute (img)	valid document
hexadecimal code	Web page
hr element (horizontal rule)	width attribute (img)
href attribute (a)	World Wide Web (WWW)
.htm (XHTML file-name extension)	XHTML (Extensible HyperText Markup Language)
<html> tag	XHTML comment
.html (XHTML file-name extension)	XHTML markup
hyperlink	XHTML tag
img element	XML declaration
level of nesting	xm1ns attribute

SELF-REVIEW EXERCISES

- 4.1** State whether each of the following is *true* or *false*. If *false*, explain why.
- An ordered list cannot be nested inside an unordered list.

- b) XHTML is an acronym for XML HTML.
 - c) Element `br` represents a line break.
 - d) Hyperlinks are denoted by `link` elements.
- 4.2** Fill in the blanks in each of the following:
- a) The _____ element inserts a horizontal rule.
 - b) A superscript is marked up using element _____ and a subscript is marked up using element _____.
 - c) The least important header element is _____ and the most important header element is _____.
 - d) Element _____ marks up an unordered list.
 - e) Element _____ marks up a paragraph.

ANSWERS TO SELF-REVIEW EXERCISES

- 4.1** a) False. An ordered list can be nested inside an unordered list and vice versa. b) False. XHTML is an acronym for Extensible HyperText Markup Language. c) True. d) False. Hyperlinks are denoted by `a` elements.
- 4.2** a) `hr`. b) `sup`, `sub`. c) `h6`, `h1`. d) `ul`. e) `p`.

EXERCISES

- 4.3** Use XHTML to create a document that contains the following text:

Internet and World Wide Web How to Program: Third Edition
Welcome to the world of Internet programming. We have provided topical coverage for many Internet-related topics.

Use `h1` for the title (the first line of text), `p` for text (the second and third lines of text) and `sub` for each word that begins with a capital letter. Insert a horizontal rule between the `h1` element and the `p` element. Open your new document in a Web browser to view the marked up document.

- 4.4** Why is the following markup invalid?

```
<p>Here is some text...
<hr />
<p>And some more text...</p>
```

- 4.5** Why is the following markup invalid?

```
<p>Here is some text...<br>
And some more text...</p>
```

- 4.6** An image named `deitel.gif` is 200 pixels wide and 150 pixels high. Write a separate XHTML statement using the `width` and `height` attributes of the `img` element to perform each of the following transformations:

- a) Increase the size of the image by 100%.
 - b) Increase the size of the image by 50%.
 - c) Change the width-to-height ratio to 2:1, keeping the `width` attained in part (a).
- 4.7** Create a link to each of the following:
- a) The file `index.html`, located in the `files` directory.
 - b) The file `index.html`, located in the `text` subdirectory of the `files` directory.
 - c) The file `index.html`, located in the `other` directory in your parent directory.
[Hint: ... signifies parent directory.]

- d) The President's e-mail address (`president@whitehouse.gov`).
 - e) The file named `README` in the `pub` directory of `ftp.cdrom.com`. [Hint: Use `ftp://`.]
- 4.8** Create an XHTML document that marks up your resume.
- 4.9** Create an XHTML document containing three ordered lists: ice cream, soft serve and frozen yogurt. Each ordered list should contain a nested, unordered list of your favorite flavors. Provide a minimum of three flavors in each unordered list.
- 4.10** Create an XHTML document that uses an image as an e-mail link. Use attribute `alt` to provide a description of the image and link.
- 4.11** Create an XHTML document that contains links to your favorite Web sites. Your page should contain the header "My Favorite Web Sites."
- 4.12** Create an XHTML document that contains an unordered list with links to all the examples presented in this chapter. [Hint: Place all the chapter examples in one directory.]
- 4.13** Modify the XHTML document (`picture.html`) in Fig. 4.7 by removing all end tags from inside the document's body. Attempt to validate this document using the W3C validation service. What happens? Next remove the `alt` attributes from the `` tags and revalidate your document. What happens?
- 4.14** Identify each of the following as either an element or an attribute:
- a) `html`
 - b) `width`
 - c) `href`
 - d) `br`
 - e) `h3`
 - f) `a`
 - g) `src`
- 4.15** State which of the following statements are *true* and which are *false*. If *false*, explain why.
- a) A valid XHTML document can contain uppercase letters in element names.
 - b) Tags need not be closed in a valid XHTML document.
 - c) XHTML documents can have the file extension `.htm`.
 - d) Valid XHTML documents can contain tags that overlap.
 - e) `&less;` is the character entity reference for the less-than (<) character.
 - f) In a valid XHTML document, `` can be nested inside either `` or `` tags.
- 4.16** Fill in the blanks in each of the following:
- a) XHTML comments begin with `<!--` and end with _____.
 - b) In XHTML, attribute values must be enclosed in _____.
 - c) _____ is the character entity reference for an ampersand.
 - d) Element _____ can be used to bold text.