

Supplement VI.A: HTML and XHTML Tutorial

For Introduction to Java Programming

By Y. Daniel Liang

This supplement introduces HTML and XHTML. XHTML is a new generation of HTML. Since XHTML is based on HTML and many Web pages were developed and are still being developed using HTML, this tutorial introduces both HTML and XHTML. Since XHTML is almost the same as HTML, this tutorial first introduces HTML, and then XHTML.

- Getting started with HTML
- HTML document structures
- HTML text appearance tags
- HTML paragraph style tags
- HTML color and font tags
- HTML list tags
- HTML table tags
- HTML hyperlink tags
- HTML image tags
- HTML image maps
- HTML meta tags
- HTML frames
- HTML forms
- XHTML

1 Introduction

<Side Remark: HTML>

HTML (HyperText Markup Language) is a markup language used to design Web pages for creating and sharing multimedia-enabled, integrated electronic documents over the Internet. HTML allows documents on the Internet to be hyperlinked and presented using fonts and image and line justification appropriate for the systems on which they are displayed. The World Wide Web is a network of static and dynamic documents, including texts, sound, and images. The Internet has been around for more than 30 years, but has only recently become popular. The Web is the major reason for its popularity. HTML documents are displayed by a program called a *Web browser*. When a document is coded in HTML, a Web browser interprets the HTML to identify the elements of the document and render it. The browser has control over the document's look-and-feel. There are no absolute, unifying HTML standards. Different vendors have rushed to introduce their own features, interpretable by their proprietary browsers.

<Side Remark: XHTML>

<Side Remark: W3C>

XHTML (eXtensible HyperText Markup Language) is a new generation of HTML, proposed by the W3C (World Wide Web Consortium). It replaces HTML to provide a unified standard for creating Web pages. Since XHTML is based on HTML, XHTML is almost identical to HTML. However, XHTML is stricter and cleaner than HTML. XHTML makes sure that a document is well-formed so that it can be rendered on any type of XHTML-compatible browsers. All the current versions of browsers support XHTML.

Since many Web pages have been developed using HTML, and HTML are still popular among Web developers, this tutorial introduces both HTML and XHTML. Since XHTML is based on HTML and HTML is almost identical to HTML, this tutorial first introduces HTML, and then XHTML.

2 Getting Started with HTML

Let us begin with an example that demonstrates the structure and syntax of an HTML document. Listing 1 is an HTML document that displays a message and a list of Web browsers: Netscape, Internet Explorer, and Mosaic.

Listing 1 HTMLDemo.html (First HTML Example)

*****PD: Please add line numbers (including space lines) in the following code*****

*****Layout: Please layout exactly. Don't skip the space. This is true for all source code in the book. Thanks, A.**

<Side Remark line 1: html tag>

<Side Remark line 2: head tag>

<Side Remark line 3: title tag>

<Side Remark line 5: body tag>

<Side Remark line 6: italic tag>

<Side Remark line 14: closing body tag>

<Side Remark line 15: closing html tag>

```
<html>
  <head>
    <title>My First Web Page</title>
  </head>
  <body>
    <i>Welcome to</i> <b>HTML</b>. Here is a list of popular Web browsers.
    <ul>
      <li>Internet Explorer
      <li>Netscape
      <li>Mosaic
    </ul>
    <hr size = "3">
    Created by <a href = "www.cs.armstrong.edu/liang">Y. Daniel Liang</a>.
  </body>
</html>
```

<Side Remark: .html or .htm>

You can create an HTML document using any text editor such as NotePad or WordPad and save it in a file with .html or .htm filename extension. Assume that you have created a file named **HTMLDemo.html** for this document. You may use any Web browser to view the document. To view it on Internet Explorer, start up your browser and choose Open from the File menu. A popup window opens to accept the filename. Type the full name of the file (including path), or click the Browser button to locate the file. Click OK to load and display the HTML file. Your document should be displayed as shown in Figure 1.



Figure 1

The XHTML page in Listing 1 is rendered by a Web browser.

<Side Remark: attributes>

What makes *Welcome to* appear in italic in Figure 1? What makes the document appear in a desired style? HTML is a document-layout and hyperlink-specification language; that is, it tells the Web browser how to display the contents of the document, including text, images, and other media, using instructions called *tags*. The browser interprets the tags and decides how to display or otherwise treat the subsequent contents of the HTML document. Tags are enclosed in brackets; `<html>`, `<i>`, ``, and `</html>` are tags that appear in the preceding HTML example. The first word in a tag, called the *tag name*, describes tag functions. Tags may have additional attributes, sometimes with a value after an equals sign, which further define their action. For example, in Listing 1, the attribute `size` in the tag `<hr>` defines the width of the bar as 3 inches. Attribute values should be enclosed in double quotes.

<Side Remark: start tag>

<Side Remark: end tag>

Tags have a *start tag* and a corresponding *end tag*. A tag has a specific effect on the region between the start tag and the end tag. For example, text advises the browser to display the word "text" in bold. and are the start and end tags for displaying boldface text. An end tag is always the start tag's name preceded by a forward slash (/). Some end tags may be omitted in HTML, but they cannot be omitted in XHTML. For example, <hr>, a tag to draw a line, has no mandatory end tag in HTML, but the end tag </hr> is required for XHTML.

A tag can be embedded inside another tag; for example, all tags are embedded within <html> and </html>. However, tags cannot overlap in XHTML; it would be wrong, for instance, to use bold and <i>italic</i>; the correct use should be <i>bold and italic</i>.

NOTE:

<Side Remark: lowercase tags>

HTML tags are not case-sensitive in HTML, but are case-sensitive in XHTML. All tags in XHTML are lowercase. To prepare for transition from HTML to XHTML, this tutorial uses lowercase tags.

The following types of tags are introduced in the upcoming sections:

- **Structure tags:** Define the structure of documents.
- **Text appearance tags:** Define the appearance of text.
- **Paragraph tags:** Define headings, paragraphs, and line breaks.
- **Font tags:** Specify font sizes and colors.
- **List tags:** Define ordered or unordered lists and definition lists.
- **Table tags:** Define tables.
- **Link tags:** Specify navigation links to other documents.
- **Image tags:** Specify where to get images and how to display them.
- **Frame tags:** Define frames.

- **Form tags:** Define forms.

3 Structure Tags

<Side Remark: head>

<Side Remark: title>

<Side Remark: body>

An HTML document begins with the `<html>` tag, which declares that the document is written with HTML. Every document has two parts, a *head* and a *body*, defined, respectively, by a `<head>` tag and a `<body>` tag. The head part contains the document title (using the `<title>` tag) and other parameters the browser may use when rendering the document; the body part contains the actual contents of the document. An HTML document may have the following structure:

```
<html>
  <head>
    <title>My First Web Page</title>
  </head>
  <body bgcolor = "white" text = "black">
    <!-- document body-->
  </body>
</html>
```

Here the special starting tag `<!--` and ending tag `-->` are used to enclose comments in the HTML documents. The comments are not displayed. You can use the `bgcolor` and `text` attributes to specify the background color and text color in the `<body>` tag.

NOTE: Your documents may be displayed properly even if you don't use `<html>`, `<head>`, `<title>`, and `<body>` tags. However, use of these tags is strongly recommended because they communicate certain information to the browser about the properties of a document; the information they provide helps in using the document effectively.

4 Text Appearance Tags

<Side Remark: content-based tags>

<Side Remark: physical tags>

HTML provides tags to advise about the appearance of text. At present, some text tags have the same effect. For example, ``, `<cite>`, and `<i>` will all display the text in italic. However, a future version of HTML may make these tags distinct. Text tag names are fairly descriptive. Text tags can be classified into two categories: *content-based tags* and *physical tags*.

4.1 Content-Based Tags

Content-based tags inform the browser to display the text based on semantic meaning, such as citation, program code, and emphasis. Here is a summary of the content-based tags:

- **<cite>**: Indicates that the enclosed text is a bibliographic citation, displayed in italic.
- **<code>**: Indicates that the enclosed text is a programming code, displayed in monospace font.
- ****: Indicates that the enclosed text should be displayed with emphasis, displayed in italic.
- ****: Indicates that the enclosed text should be strongly emphasized, displayed in bold.
- **<var>**: Indicates that the enclosed text is a computer variable, displayed in italic.
- **<address>**: Indicates that the enclosed text is an address, displayed in italic.

Table 1 lists the content-based tags and provides examples of their use.

Table 1 Using Content-Based Tags

| Tag | Example | Display |
|---|---|-----------------------------|
| <u><cite> . . . </cite></u> | <u><cite>bibliographic</cite></u> | <i>bibliographic</i> |
| <u><code> . . . </code></u> | <u><code>source code</code></u> | <code>source code</code> |
| <u> . . . </u> | <u>emphasis</u> | <i>emphasis</i> |
| <u> . . . </u> | <u>strongly emphasized</u> | strongly emphasized |
| <u><var> . . . </var></u> | <u><var>programming variable</var></u> | <i>programming variable</i> |
| <u><address> . . . </address></u> | <u><address>Computer Dept</address></u> | <i>Computer Dept</i> |

4.2 Physical Tags

Physical tags explicitly ask the browser to display text in bold, italic, or other ways. Following are six commonly used physical tags:

- **<i>** (italic)

- (bold)
- <u> (underline)
- <tt> (monospace)
- <strike> (strike-through text)
- <blink> (blink) (doesn't work with Internet Explorer)

Table 2 lists the physical tags and provides examples of their use.

Table 2 Using Physical Tags

| Tag | Example | Display |
|---|--|--------------------------|
| <u><i> . . . </i></u> | <u><i>italic</i></u> | <i>italic</i> |
| <u> . . . </u> | <u>bold</u> | bold |
| <u><u> . . . </u></u> | <u><u>underline</u></u> | <u>underline</u> |
| <u><tt> . . . </tt></u> | <u><tt>monospace</tt></u> | <u>monospace</u> |
| <u><strike> . . . </strike></u> | <u><strike>strike</strike></u> | <u>strike</u> |

5 Paragraph-Style Tags

There are many tags in HTML for dealing with paragraph styles. There are six heading tags (<h1>, <h2>, <h3>, <h4>, <h5>, <h6>) for different sizes of headings, a line-break tag (
), a paragraph start tag (<p>), a pre-format tag (<pre>), and a block-quote tag (<blockquote>).

<Side Remark: header tags>

The six heading tags indicate the highest (<h1>) and lowest (<h6>) precedence a heading may have in the document. Heading tags may be used with an align attribute to place the heading toward *left*, *center*, or *right*. The default alignment is left. For example, <h3 align = right>Heading</h3> tells the browser to right-align the heading.

<Side Remark: line-break tag>

The line-break tag
 tells the browser to start displaying from the next line. This tag does not have a closing tag in HTML. In XHTML, you should write
.

<Side Remark: paragraph start tag>

The paragraph-start tag `<p>` signals the start of a paragraph.

<Side Remark: pre-format tag>

The `<pre>` tag and its required end tag (`</pre>`) define the enclosed segment to be displayed in monospaced font by the browser.

<Side Remark: block quote tag>

The `<blockquote>` tag is used to contain text quoted from another source. The quote will be indented from both left and right.

Listing 2 shows the XHTML source code that illustrates the use of paragraph tags. The file is stored in `example2.html` and is rendered as shown in Figure 2.

Listing 2 ParagraphTagDemo.html (Paragraph Tag Demo)

*****PD: Please add line numbers (including space lines) in the following code*****

*****Layout: Please layout exactly. Don't skip the space. This is true for all source code in the book. Thanks, A.**

<Side Remark line 1: html tag>

<Side Remark line 2: head tag>

<Side Remark line 9: title tag>

<Side Remark line 12: body tag>

```
<html>
  <head>
    <title>Paragraph Tag Demo</title>
  </head>

  <body>
    <h1 align=right>h1: Heading 1</h1>
    <h3 align=center>h3: Heading 3</h3>
    <h6 align=left>h6: Heading 6</h6>
    <p> This document demonstrates paragraph tags </p>
    <pre>preformat tag</pre>
    <blockquote>
      block quote tag
      <br />and line break
    </blockquote>
  </body>
</html>
```

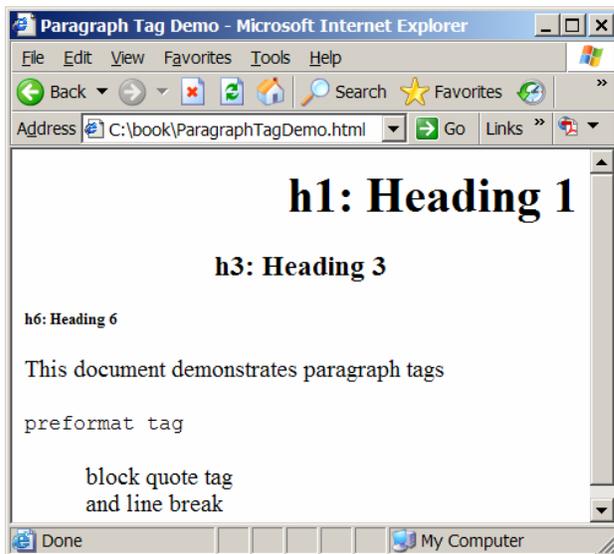


Figure 2

Paragraph tags specify heading styles, paragraph format, block quote, line break, and so on.

6 Specifying Fonts and Colors

<Side Remark: <basefont>>

<Side Remark: >

You can use <basefont> and to specify fonts and colors. The <basefont> tag is typically placed in the head of an HTML document. However, it may appear anywhere in the document, and it may appear many times, each time with new attributes. The <basefont> tag does not have a closing tag. The tag allows you to specify the font and color of the enclosed text.

NOTE:

<Side Remark: *using style sheets*>

Developing large Web sites where fonts and color information had to be added to every single page is a long, expensive, and painful process. The W3C recommends using Cascading Style Sheets to achieve the same effect.

<Side Remark: face>

<Side Remark: size>

<Side Remark: color>

You may use the face, size, and color attributes to specify font name, size, and color. Popular font names are Times Roman, Verdana, Helvetica, Courier, and Courier New. Many browsers use a relative model for sizing fonts, ranging from 1 to 7; the default base font size is 3. Each successive size is 20 percent larger than its predecessor in the range.

<Side Remark: color names>

The color attribute sets the color for the enclosed text between `` and ``. The value of the attribute is a six-digit hex number preceded by a pound sign (#). The first two digits are the red component, the next two digits are the green component, and the last two digits are the blue component. The digits are from 00 to FF. Alternatively, you may set the color by using standard names. The supported color names in HTML 4.0 are aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow.

Listing 3 gives an example that demonstrates the `<basefont>` and `` tags. The example is rendered as shown in Figure 3.

Listing 3 FontColorDemo.html (Font and Color Demo)

*****PD: Please add line numbers (including space lines) in the following code*****

*****Layout: Please layout exactly. Don't skip the space. This is true for all source code in the book. Thanks, A.**

<Side Remark line 1: html tag>

<Side Remark line 2: head tag>

<Side Remark line 3: title tag>

<Side Remark line 5: basefont>

<Side Remark line 6: body tag>

<Side Remark line 8: font>

```
<html>
  <head>
    <title>Demonstrating Fonts, Size and Color</title>
  </head>
  <basefont size = "5" face = "Verdana" color = "green">
  <body>
    basefont basefont basefont<br />
    <font size = "4" face = "Helvetica" color = "blue">
      blue Helvetica blue Helvetica</font><br />
    <font size = "3" face = "Courier New" color = "#FF0000">
      Courier New Courier New</font>
  </body>
</html>
```



Figure 3

Font and color tags specify fonts and colors in HTML pages.

7 List Tags

HTML allows you to define three kinds of lists: *ordered lists*, *unordered lists*, and *definition lists*. You can also build nested lists. Listing 1 contains an unordered list of three Web browsers.

7.1 Ordered Lists

<Side Remark: tag>

<Side Remark: type attribute>

Ordered lists label the items they contain. An ordered list is used when the sequence of the listed items is important. For example, chapters are listed in order. An ordered list starts with the tag and ends with , and items are placed in between. Each item begins with an tag. The browser automatically numbers list items, starting from numeric 1. Instead of using the default numeric numbers for labeling, you may associate the tag with a type attribute. The value of the type determines the style of the label.

- Type value A for uppercase letter labels A, B, C, . . .
- Type value a for lowercase letter labels a, b, c, . . .
- Type value I for capital roman numerals I, II, III, . . .
- Type value i for lowercase roman numerals i, ii, iii, . . .
- Type value 1 for Arabic numerals 1, 2, 3, . . .

7.2 Unordered Lists

<Side Remark: tag>

When the sequence of the listed items is not important, use an unordered list. For example, a list of Web browsers can be given in any order. An unordered list starts with the tag and ends with . Inside, you use tags for items. By default, the browser uses bullets to mark each item. You may use disc, circle, or square as type values to indicate the use of markers other than bullets.

7.3 Definition Lists

<Side Remark: <dl> tag>

<Side Remark: <dt> tag>

<Side Remark: <dd> tag>

A definition list is used to define terms. The list is enclosed between `<dl>` and `</dl>` tags. Inside the tags are the terms and their definitions. The term and definition have the leading tags `<dt>` and `<dd>`, respectively. Browsers typically render the term name at the left margin and the definition below it and indented.

Listing 4 gives an example that illustrates the use of tags for ordered lists, unordered lists, definition lists, and nested lists. The page is rendered as shown in Figure 4. An order list is created in lines 8-12. An unordered list is created in lines 14-23. Inside this list, another list is embedded in lines 16-20. A definition list is created in lines 25-28.

Listing 4 HTMLListDemo.html (List Tag Demo)

*****PD: Please add line numbers (including space lines) in the following code*****

*****Layout: Please layout exactly. Don't skip the space. This is true for all source code in the book. Thanks, A.**

<Side Remark line 8: ordered list>

<Side Remark line 14: unordered list>

<Side Remark line 20: definition list>

```
<html>
  <head>
    <title>Demonstrating List Tags</title>
  </head>
  <body bgcolor = "white">
    <h3>List Tags</h3>
    An ordered List
    <ol type = "A">
      <li>Chapter 1: Introduction to Java</li>
      <li>Chapter 2: Java Building Elements</li>
      <li>Chapter 3: Control Structures</li>
    </ol>
    An unordered List
    <ul type = "square">
      <li>Apples
        <ul type = "circle">
          <li>Golden Delicious</li>
          <li>Fuji apple</li>
          <li>Macintosh</li>
        </ul></li>
      <li>Oranges</li>
      <li>Peaches</li>
    </ul>
    Definition List
    <dl>
      <dt>What is Java?</dt>
      <dd>An Internet programming language.</dd>
    </dl>
```

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

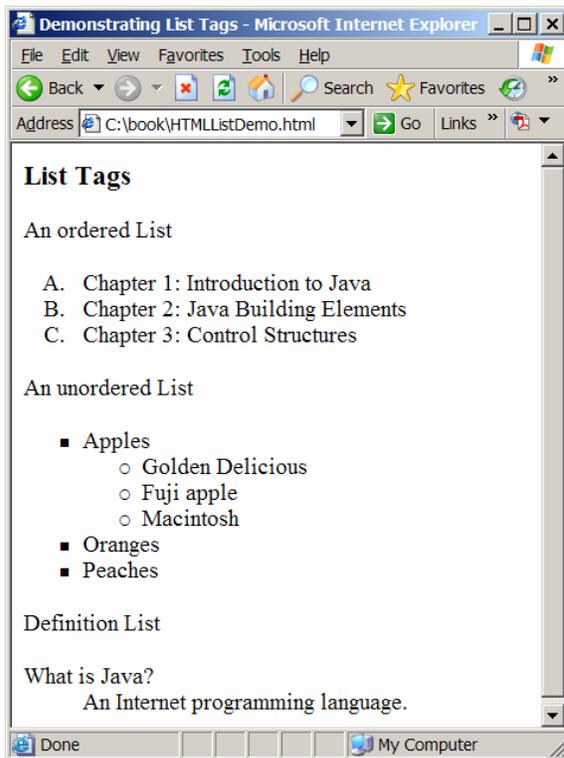


Figure 4

HTML list tags can display ordered lists, unordered lists, and definition lists.

8 Table Tags

<Side Remark: <table> tag>

<Side Remark: <tr> tag>

<Side Remark: <td> tag>

Tables are collections of numbers and words arranged in rows and columns of cells. In HTML, table elements, including data items, row and column headers, and captions, are enclosed between <table> and </table> tags. Several table tags may be used to specify the layout of a table. Each row in the table is wrapped by <tr> and </tr>. Inside the row, data or words in a cell are enclosed by <td> and </td>. You may use <caption>...</caption> to display a caption for the table, and <th>...</th> to display column headers.

Table tags may be used with attributes to obtain special effects. Here are some useful attributes:

<Side Remark: border attribute>

- **border** can appear in the <table> tag to specify that all cells are surrounded with a border. The value for

this attribute is an integer indicating the thickness of the border. 0 indicates no borders.

<Side Remark: align attribute>

- **align** can appear in the `<caption>`, `<tr>`, `<th>`, or `<td>` tag. If it appears in `<caption>`, it specifies whether the caption appears above or below the table, using the values `top` or `bottom`. The default is `align=top`. If it appears in `<tr>`, `<th>`, or `<td>`, `align` specifies whether the text is aligned to the left, the right, or centered inside the table cell(s).

<Side Remark: valign attribute>

- **valign** can appear in `<tr>`, `<th>`, or `<td>`. The values of the attribute are `top`, `middle`, and `bottom` to specify whether the text is aligned to the top, the bottom, or centered inside the table cell(s).

<Side Remark: colspan attribute>

- **colspan** can appear in any table cell to specify how many columns of the table the cell should span. The default value is 1.

<Side Remark: rowspan attribute>

- **rowspan** can appear in any column to specify how many rows of the table the cell should span. The default value is 1.

Listing 5 gives an example of a simple table with a border, centered (line 6), a caption (line 7), column headers (lines 8-12), and four rows (lines 13-32). The table is rendered as shown in Figure 5.

Listing 5 HTMLTableDemo.html (Table Tag Demo)

*****PD: Please add line numbers (including space lines) in the following code*****

*****Layout: Please layout exactly. Don't skip the space. This is true for all source code in the book. Thanks, A.**

<Side Remark line 6: <table>>

<Side Remark line 7: <caption>>

<Side Remark line 8: <tr>>

<Side Remark line 9: <th>>

<Side Remark line 14: <td>>

```
<html>
  <head>
    <title>Demonstrating Table Tags</title>
  </head>
  <body bgcolor = "white">
    <table border = "1" align = "center">
      <caption>Student Table</caption>
```

```

<tr> <!-- column headers -->
  <th>SSN</th>
  <th>Name</th>
  <th>Major</th>
</tr>
<tr align = "center"> <!-- cell data -->
  <td>111-22-1111</td>
  <td>John Baker</td>
  <td>Computer Science</td>
</tr>
<tr > <!-- cell data -->
  <td>111-22-1112</td>
  <td>Peter Yao</td>
  <td>Mathematical Science</td>
</tr>
<tr > <!-- cell data -->
  <td>111-22-1113</td>
  <td>George Smith</td>
  <td>Engineering</td>
</tr>
<tr > <!-- cell data -->
  <td>111-22-1114</td>
  <td>Cindy Wing</td>
  <td></td>
</tr>
</table>
</body>
</html>

```

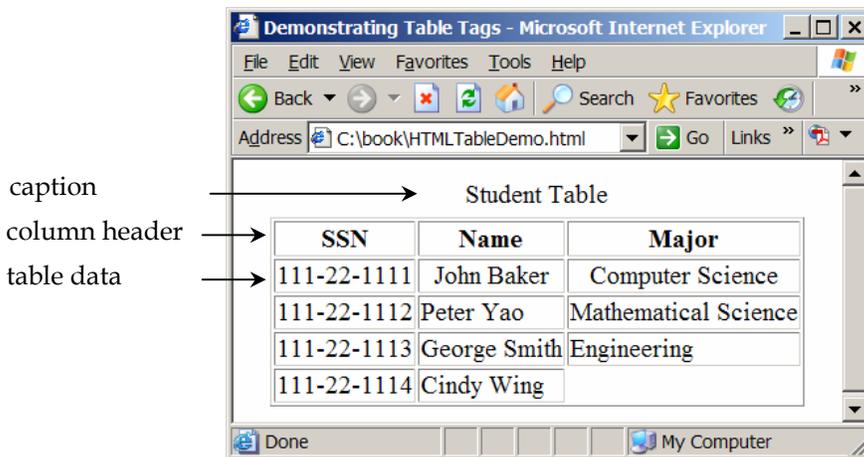


Figure 5

Table tags are useful for displaying tables in HTML pages.

9 Hyperlink Tags

The true power of HTML lies in its capability to join collections of documents together into a full electronic library of information, and to link documents with other documents over the Internet. This is called *hypertext*

linking, which is the key feature that makes the Web appealing and popular. By adding hypertext links, called *anchors*, to your HTML document, you create a highly intuitive information flow and guide users directly to the information they want. You can link documents on different computers or on the same computer, and can jump within the same document using anchor tags.

9.1 Linking Documents on Different Computers

<Side Remark: <a> tag>

<Side Remark: href attribute>

Every document on the Web has a unique address, known as its *Uniform Resource Locator* (URL). To navigate from a source document to a target document, you need to reference the target's URL inside the anchor tags <a> and using attribute href. The following example displays a list of database vendors:

```
<ul>
<li><a href = "http://www.oracle.com">Oracle</a></li>
<li><a href = "http://www.sybase.com">Sybase</a></li>
<li><a href = "http://www.microsoft.com">Microsoft</a></li>
</ul>
```

In this example, clicking on Oracle will display the Oracle homepage. The URL of Oracle's home page Internet address is **http://www.oracle.com**. The general format of a URL is:

method://servername:port/pathname/fullfilename

method is the name of the operation that is performed to interpret this URL. The most common methods are http, ftp, and file.

- **http**: Accesses a page over the network using the HTTP protocol. For example, **http://www.microsoft.com** links to Microsoft's homepage. **http://** can be omitted.
- **ftp**: Downloads a file using anonymous FTP service from a server; for example, **ftp://hostname/directory/fullfilename**.
- **file**: Reads a file from the local disk. For example, **file://home/liang/liang.html** displays the file **liangy.html** from the directory /home/liang on the local machine.

servername is a computer's unique Internet name or Internet Protocol (IP) numerical address on the network. For example, **www.sun.com** is the hostname of Sun Microsystem's Web server.

If a server name is not specified, it is assumed that the file is on the same server.

port is the TCP port number that the Web server is running on. Most Web servers use port number 80 by default.

pathname is optional and indicates the directory under which the file is located.

fullfilename is optional and indicates the target filename. Web servers generally use **index.html** on UNIX and **default.htm** on Windows for a default filename. For example, `Oracle` is equivalent to `Oracle`.

9.2 Linking Documents on the Same Computer

To link documents on the same computer, use the file method rather than the http method in the target URL. There are two types of links: *absolute links* and *relative links*. When linking to a document on a different machine, you must use an absolute link to identify the target document. An absolute link uses a URL to indicate the complete path to the target file.

When you are linking to a document on the same computer, it is better to use a relative link. A relative URL omits method and server name and directories. For instance, assume that the source document is under directory ~liangy/teaching on the server www.cs.armstrong.edu. The URL

`file:// www.cs.armstrong.edu/liang/teaching/teaching.html`

is equivalent to

`file://teaching.html`

Here, file:// can be omitted. An obvious advantage of using a relative URL is that you can move the entire set of documents to another directory or even another server and never have to change a single link.

9.3 Linking Within the Same Document

<Side Remark: name attribute>

HTML offers navigation within the same document. This is helpful for direct browsing of interesting segments of the document. Let us use an example in Listing 6 to demonstrate how to jump within the same document. The example shows a document with three sections, as shown in Figure 6. When the user clicks *Section 1: Introduction* on the list, the browser jumps to *Section 1* of the document. The name attribute

within the `<a>` tag labels the section. The label is used as a link to the section. This feature is also known as *using bookmarks*.

When you test this example, make the window small so that you can see the effects of jumping to each reference through the link tags.

Listing 6 HTMLHyperlinkDemo.html (Hyperlink Demo)

*****PD: Please add line numbers (including space lines) in the following code*****

*****Layout: Please layout exactly. Don't skip the space. This is true for all source code in the book. Thanks, A.**

<Side Remark line 7: `<a>` tag>

<Side Remark line 12: `name` attribute>

```
<html>
  <head>
    <title>Demonstrating Link Tags</title>
  </head>
  <body>
    <ol>
      <li><a href = "#introduction">Section 1: Introduction</a>
      <li><a href = "#methodology">Section 2: Methodology</a>
      <li><a href = "#summary">Section 3: Summary</a>
    </ol>

    <h3><a name = "introduction">Section 1: Introduction</a></h3>
    an introductory paragraph

    <h3><a name = "methodology">Section 2: Methodology</a></h3>
    a paragraph on methodology

    <h3><a name = "summary">Section 3: Summary</a></h3>
    a summary paragraph
  </body>
</html>
```

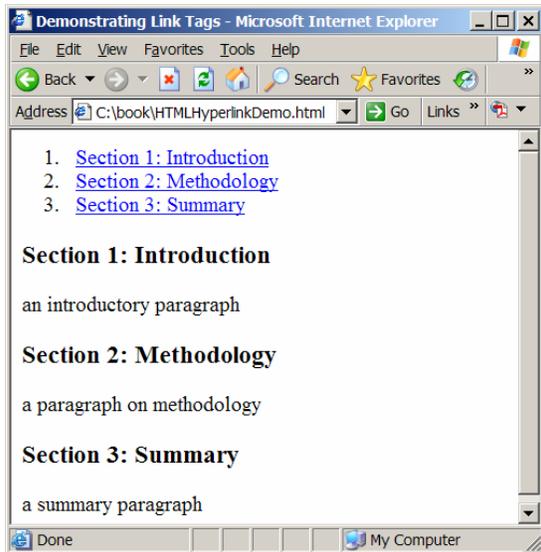


Figure 6

Hyperlink tags link documents.

10 Horizontal Bar Tags

<Side Remark: <hr> tag>

The horizontal bar tag (<hr>) is used to display a rule. It is useful in separating sections of your document with horizontal rules. The attributes size, width, and align can be associated to achieve the desired effect. The rule can be thickened using the size attribute with values in pixels. The width attribute specifies the length of the bar with values in either absolute number of pixels or extension across a certain percentage of the page. The align attribute specifies whether the bar is left, center, or right aligned.

Listing 7 gives an example that illustrates the use of the size, width, and align attributes in horizontal bar tags. The example is rendered as shown in Figure 7.

Listing 7 HTMLBarDemo.html (Bar Tag Demo)

*****PD: Please add line numbers (including space lines) in the following code*****
*****Layout: Please layout exactly. Don't skip the space. This is true for all source code in the book. Thanks, A.**

<Side Remark line 7: <hr> tag>

```
<html>
  <head>
    <title>Demonstrating Horizontal Rules</title>
  </head>
  <body bgcolor = "white">
    <h3>Horizontal Rules</h3>
    <hr size = "3" width = "80%" align = "left">
    <hr size= "2" width = "20%" align = "right" noshade>
```

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



Figure 7

Horizontal bar tags are often used to separate contents in documents.

11 Image Tags

One of the most compelling features of the Web is its ability to embed graphics in a document. Graphics can be used for icons, pictures, illustrations, drawings, and so on. They bring a live dimension to your documents. You may use an image as a visual map of hyperlinks. This section introduces the use of *image tags*.

<Side Remark: >

<Side Remark: alt>

<Side Remark: width>

<Side Remark: height>

The tag lets you embed an image into the document. You can use the src attribute to specify the source of the image, use the alt attribute to specify an alternative text message to be displayed in case the client's browser cannot display the image, and use the width and height attributes to adjust image to desired sizes.

Listing 8 gives an example that creates a document with image tags. The example is rendered as shown in Figure 8.

Listing 8 HTMLImageDemo.html (Image Tag Demo)

*****PD: Please add line numbers (including space lines) in the following code*****

*****Layout: Please layout exactly. Don't skip the space. This is true for all source code in the book. Thanks, A.**

<Side Remark line 7: tag>

<Side Remark line 9: width and height>

```
<html>
```

```
<head>
```

```

<title>Demonstrating Image Tags</title>
</head>
<body bgcolor = "white">
  <h3>Image Tags</h3>
  <img src = "image/us.gif" align = "middle" alt = "US Flag">
  <img src = "image/us.gif" align = "middle" alt = "US Flag"
  width = "120" height = "60">
  <img src = "image/us.gif" align = "middle" alt = "US Flag"
  width = "60" height = "30"><p>
</body>
</html>

```



Figure 8

Image tags display images in HTML pages.

<Side Remark: align>>

<Side Remark: image position>

You can use the align attribute to tell the browser where to place the image with possible values left, center, right, top, middle, bottom, and texttop. Listing 9 gives an example that demonstrates image positions. The example is rendered as shown in Figure 9.

Listing 9 HTMLImagePositionDemo.html (Position Images)

*****PD: Please add line numbers (including space lines) in the following code*****

*****Layout: Please layout exactly. Don't skip the space. This is true for all source code in the book. Thanks, A.**

<Side Remark line 7: top>

<Side Remark line 10: bottom>

<Side Remark line 13: left>

<Side Remark line 17: right>

```

<html>
  <head>
    <title>Demonstrating Image Tags</title>
  </head>
  <body bgcolor = "white">
    <h3>Image Tags</h3>
    An image <img src = "image/us.gif" align = "top" alt = "US Flag"
    width = "60" height = "30"> aligned top.<p>

```

```

An image <img src = "image/us.gif" align = "bottom" alt = "US Flag"
width = "60" height = "30"> aligned bottom.<p>

<img src = "image/us.gif" align = "left" alt = "US Flag"
width = "60" height = "30"> Image can float to the left using the left
value for the align attribute.<p>

<img src = "image/us.gif" align = "right" alt = "US Flag"
width = "60" height = "30"> Image can float to the right using the left
value for the align attribute.<p>
</body>
</html>

```

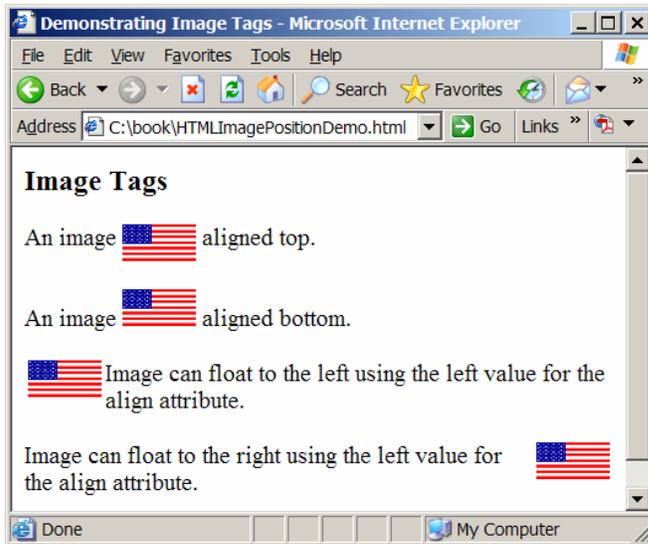


Figure 9

Images can be positioned using the align attribute.

11 Image Maps

You may designate certain areas of an image for users to click on as links. These areas may be a circle, a rectangle, or a polygon. Each area is defined using the `<area>` tag. The `<area>` tag uses the `shape` attribute to specify a shape with values `circle`, `rectangle`, and `polygon`. All the areas are embedded inside a `<map>` tag. The `<map>` uses an `id` attribute to identify the map. Listing 10 gives an example that demonstrates image maps. The image is the US map. Three clickable areas are Colorado, Georgia, and Texas, as shown in Figure 10.

Listing 10 HTMLImageMapDemo.html (Image Map Demo)

```

***PD: Please add line numbers (including space lines) in
the following code***
***Layout: Please layout exactly. Don't skip the space. This
is true for all source code in the book. Thanks, A.
<Side Remark line 8: usemap>

```

<Side Remark line 10: map>
 <Side Remark line 11: rectangle area>
 <Side Remark line 13: polygon area>
 <Side Remark line 15: circle area>

```

<html>
  <head>
    <title>Demonstrating Image Maps</title>
  </head>
  <body bgcolor = "white">
    <h3>Image Maps</h3>

    <img src = "image/us-map.gif" usemap = "#states">

    <map id = "states">
      <area shape = "rectangle" coords = "125, 101, 175, 152"
        href = "http://www.colorado.gov">
      <area shape = "polygon" href = "http://www.georgia.gov"
        coords = "315, 169, 338, 168, 360, 193, 335, 209, 328, 209">
      <area href = "http://www.texas.gov" shape = "circle"
        coords = "199, 216, 40">
    </map>
  </body>
</html>

```

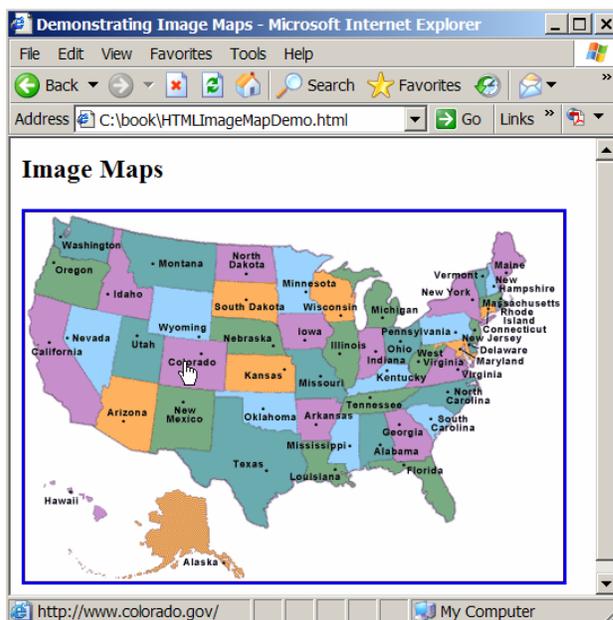


Figure 10

You can click areas on an image.

The `<map>` tag in line 10 is defined with the id `states`. The `` tag (line 8) defines the image that uses the map identified by `#states`.

Three areas are defined in the map. The `<area>` tag in lines 11-12 defines an area in a rectangle. The rectangle is

specified by the coordinates of its upper-left corner (125, 101) and lower-right corner (175, 152).

The `<area>` tag in lines 13-14 defines an area in a polygon. The polygon is specified by the coordinates of all corners (315, 169), (338, 168), (360, 193), (335, 209), and (328, 209). Connecting all the corners form a polygon.

The `<area>` tag in lines 15-16 defines an area in a circle. The circle is specified by the coordinate of its center (199, 216) and the radius 40.

To locate an area in the image, you need to know the coordinates of the area. You can pinpoint the coordinates in the image by turning an image into an image map using the `ismap` attribute with a hyperlink on the image, as shown in Listing 11. If you move the mouse over the image, the coordinates will be displayed on the status bar.

Listing 11 HTMLMapCoordinatesDemo.html (Locating Coordinates)

*****PD: Please add line numbers (including space lines) in the following code*****

*****Layout: Please layout exactly. Don't skip the space. This is true for all source code in the book. Thanks, A.**

<Side Remark line 7: `<a>`>

<Side Remark line 8: `ismap`>

```
<html>
  <head>
    <title>Demonstrating Image Maps</title>
  </head>
  <body bgcolor = "white">
    <h3>Image Maps</h3>
    <a href = "dummy.html">
      <img src = "image/us-map.gif" ismap>
    </a>
  </body>
</html>
```

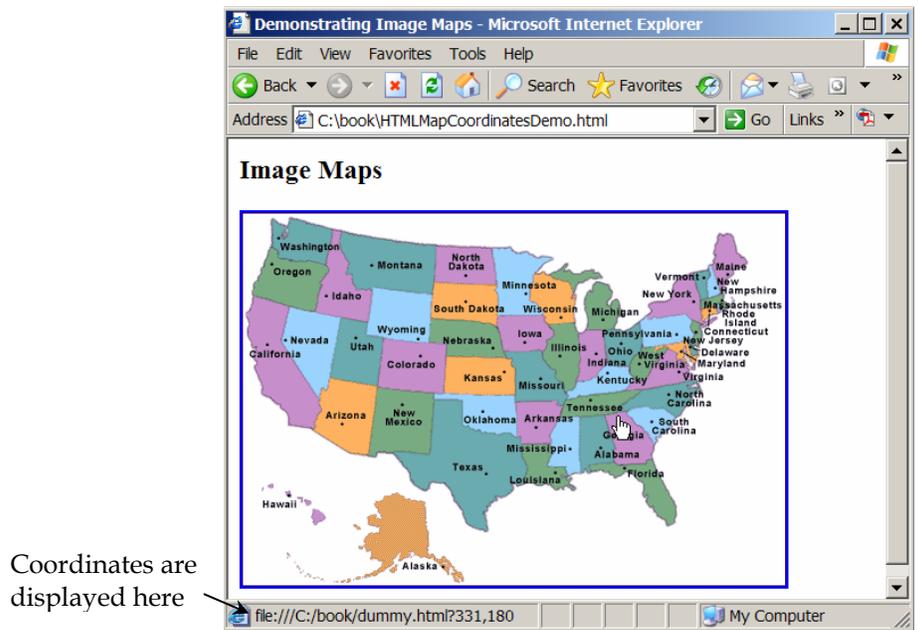


Figure 11

The coordinates of the mouse point are displayed.

12 Special Characters

Some characters like the angle brackets `<` and `>` have special meaning in HTML. To display them, you have to use character entities (in the form of `&code;`) for representing a character. The code may be a name or a number proceeded by the symbol `#`. For example, `<` is the entity name for `<` and `<` is the entity number for `<`. Table 1 lists some common used character entities.

Table 1: Common Character Entities

| Character | Description | Entity Name | Entity Number |
|--------------------|----------------------|---|-------------------------|
| | non-breaking space | <code>&nbsp;</code> | <code>&#160;</code> |
| <code><</code> | less than | <code>&lt;</code> | <code>&#60;</code> |
| <code>></code> | greater than | <code>&gt;</code> | <code>&#62;</code> |
| <code>&</code> | ampersand | <code>&amp;</code> | <code>&#38;</code> |
| <code>"</code> | quotation mark | <code>&quot;</code> | <code>&#34;</code> |
| <code>'</code> | apostrophe | <code>&apos;</code> (does not work in IE) | <code>&#39;</code> |
| ¢ | cent | <code>&cent;</code> | <code>&#162;</code> |
| £ | pound | <code>&pound;</code> | <code>&#163;</code> |
| ¥ | yen | <code>&yen;</code> | <code>&#165;</code> |
| § | section | <code>&sect;</code> | <code>&#167;</code> |
| © | copyright | <code>&copy;</code> | <code>&#169;</code> |
| ® | registered trademark | <code>&reg;</code> | <code>&#174;</code> |

Listing 12 gives an example that displays the special characters in a table.

Listing 12 HTMLCharacterEntityDemo.html (Displaying Special Characters)

*****PD: Please add line numbers (including space lines) in the following code*****

*****Layout: Please layout exactly. Don't skip the space. This is true for all source code in the book. Thanks, A.**

<Side Remark line 14: space character>

<Side Remark line 19: < character>

<Side Remark line 24: > character>

```
<html>
  <head>
    <title>Demonstrating Special Characters</title>
  </head>
  <body bgcolor = "white">
    <table border = "1" align = "center">
      <caption>Common Character Entities</caption>
      <tr <!-- column headers -->
        <th>Character</th>
        <th>Description</th>
        <th>Entity Name</th>
      </tr>
      <tr <!-- cell data -->
        <td>&nbsp;</td>
        <td>non-breaking space</td>
        <td>&nbsp;</td>
      </tr>
      <tr > <!-- cell data -->
        <td>&lt;</td>
        <td>less than</td>
        <td>&lt;</td>
      </tr>
      <tr > <!-- cell data -->
        <td>&gt;</td>
        <td>greater than</td>
        <td>&gt;</td>
      </tr>
      <tr > <!-- cell data -->
        <td>&amp;</td>
        <td>ampersand</td>
        <td>&amp;&amp;</td>
      </tr>
    </table>
  </body>
</html>
```

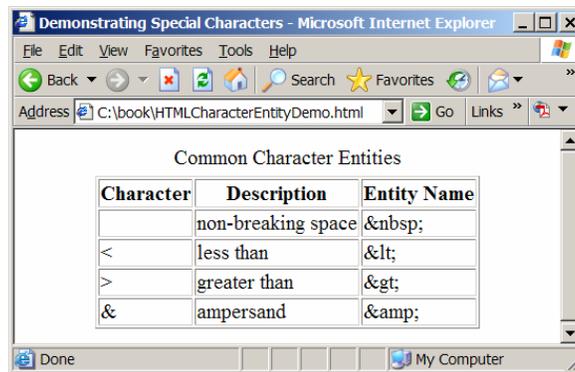


Figure 12

Special characters are represented using character entities.

13 The Meta Tag

The `<meta>` tag can be used to describe the document contents and keywords to search engines. So your documents are more likely to be cataloged by search engines. The `<meta>` can also be used to redirect the user to another URL if your site address has changed.

Listing 13 gives an example that demonstrates the `<meta>` tag. The `<meta>` tags are embedded inside the `<head>` tag. The `<meta>` tag in lines 4-5 redirects the page to a new URL after five seconds. The `<meta>` tag in lines 6-7 gives a description for the page. The `<meta>` tag in lines 8-9 describes the keywords in the document.

Listing 13 exampleA13.html (Displaying Special Characters)

*****PD: Please add line numbers (including space lines) in the following code*****
*****Layout: Please layout exactly. Don't skip the space. This is true for all source code in the book. Thanks, A.**
<Side Remark line 14: space character>
<Side Remark line 19: < character>
<Side Remark line 24: > character>

```

<html>
  <head>
    <title>Demonstrating Meta Tags</title>
    <meta http-equiv = "refresh"
      content = "5; url = http://www.cs.armstrong.edu">
    <meta name = "description"
      content = "Introduction to Java Programming, Sixth Edition">
    <meta name = "keywords"
      content = "Java, Comprehensive, Core, Data Structures">
  </head>
  <body bgcolor="white">
    <p>This page has been moved to
    <a href = "http://www.cs.armstrong.edu">www.cs.armstrong.edu

```

```

        </a>.</p>

        <p>You will be redirected to the new address in five seconds.</p>

        <p>If you see this message for more than 5 seconds,
        please click on the link above!</p>
    </body>
</html>

```

14 Frames

The display area in a browser can be divided into frames. With frames, you can display multiple pages. You use the `<frameset>` tag to define the layout of the frames. Each `frameset` uses the `rows` or `cols` attribute to indicate the amount of screen area each row or column will occupy. Inside the `<frameset>` tag, you use the `<frame>` tag to put an HTML document into a frame.

NOTE:

<Side Remark: CSS replacing frames>

There are three problems on frames: (1) The Web developer must track multiple documents; (2) It is difficult to print all frames; (3) Some browsers do not support frames. The W3C recommends using Cascading Style Sheets (CSS) to achieve the same effect.

Listing 14 gives an example that demonstrates the frames, as shown in Figure 13. A Web page that uses frames consists of the head section and the frameset section without the body section. The `<frameset>` tag in line 5 specifies the layout of the frames. You can use the `cols` or `rows` to specify two columns or two rows. In this case, `cols = "294,*"` specifies tells the browser that there are two vertical frames. The first frame occupies 294 pixels and the second fills the rest. The `<frame>` tag specifies the document to be displayed in the frame. The `<frame>` tag in line 6 displays the document named `left.html` (shown in Listing 15) and the `<frame>` tag in line 7 displays the document whose URL is www.oracle.com.

Listing 14 HTMLFrameDemo.html (Displaying Frames)

```

***PD: Please add line numbers (including space lines) in
the following code***
***Layout: Please layout exactly. Don't skip the space. This
is true for all source code in the book. Thanks, A.
<Side Remark line 5: <frameset>>
<Side Remark line 6: left frame>
<Side Remark line 7: right frame>
    <html>

```

```

<head>
  <title>Demonstrating Frames</title>
</head>
<frameset cols = "294,*">
  <frame name = "left" src = "left.html">
  <frame name = "right" src = "http://www.oracle.com">
</frameset>
</html>

```

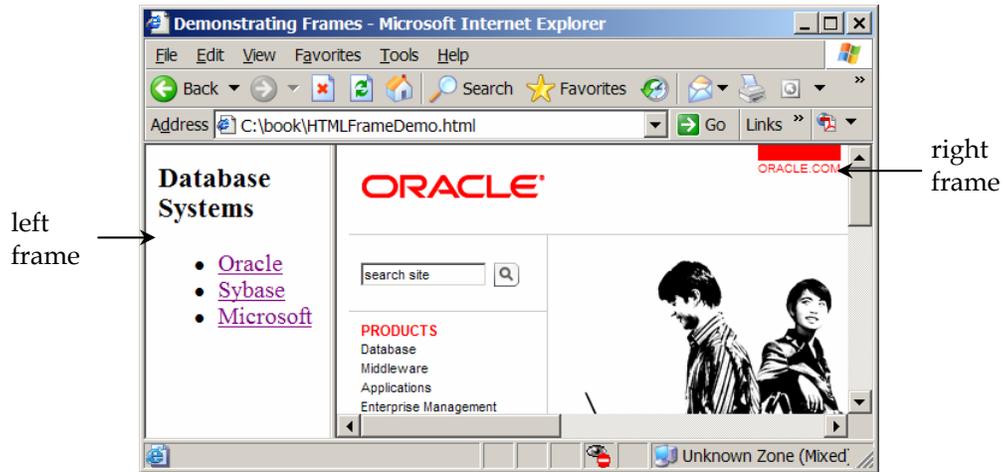


Figure 13

Each frame displays an independent Web page.

Listing 15 left.html (Left Frame)

*****PD: Please add line numbers (including space lines) in the following code*****

*****Layout: Please layout exactly. Don't skip the space. This is true for all source code in the book. Thanks, A.**

<Side Remark line 8: target>

```

<html>
<head>
  <title>Demonstrating Frames</title>
</head>
<body bgcolor = "white">
  <h3>Database Systems</h3>
  <ul>
    <li><a href = "http://www.oracle.com" target = "right">Oracle</a></li>
    <li><a href = "http://www.sybase.com" target = "right">Sybase</a></li>
    <li><a href = "http://www.microsoft.com" target = "right">Microsoft</a></li>
  </ul>
</body>
</html>

```

<Side Remark: referencing frame>

The `<frame>` tag has the optional `name` attribute, which is useful for referencing the frame dynamically. The name attribute for the second frame is `right`. In Listing 15, the hyperlink target is set to the frame named `right`. So, when

you click on Oracle, Sybase, or Microsoft, the hyperlinked content will be displayed in the right frame, as shown in Figure 14.



Figure 14

Clicking on Microsoft on the left frame displays the content to the right frame.

By default, the contents in a frame can be scrolled automatically. To disable it, add `scrolling = "no"` in the `<frame>` tag. By default, the divider line can be moved freely. To disable it, add `noresize` in the `<frame>` tag.

<Side Remark: embedded frames>

The `<frameset>` tag can be embedded to create more complex layout. For example, the following script defines three frames:

```
<html>
  <head>
    <title>Demonstrating Frames</title>
  </head>
  <frameset rows = "100,*">
    <frame name = "top" src = "top.html">
      <frameset cols = "200,*">
        <frame name = "left" src = "left.html">
          <frame name = "right" src = "right.html">
        </frameset>
      </frameset>
    </frameset>
  </html>
```

Often it is more flexible and manageable to define two frames in one `<frameset>` tag. The source of the `<frame>` tag may be a page with multiple frames. Listing 16 gives such an example, as shown in Figure 15. The example defines two

horizontal frames in the `<frameset>`. The first frame in line 6 refers to `top.html` (Listing 17). The second frame in line 7 refers to `exampleA14.html`. Recall that `exampleA14.html` defines two vertical frames.

Listing 16 `HTMLMultiplFrameDemo.html` (Displaying Multiple Frames)

*****PD: Please add line numbers (including space lines) in the following code*****

*****Layout: Please layout exactly. Don't skip the space. This is true for all source code in the book. Thanks, A.**

<Side Remark line 5: <frameset>>

```
<html>
  <head>
    <title>Demonstrating Multiple Frames</title>
  </head>
  <frameset rows = "100,*">
    <frame name = "top" src = "top.html">
    <frame name = "main" src = "HTMLFrameDemo.html">
  </frameset>
</html>
```

Listing 17 `top.html` (Top Frame)

*****PD: Please add line numbers (including space lines) in the following code*****

*****Layout: Please layout exactly. Don't skip the space. This is true for all source code in the book. Thanks, A.**

<Side Remark line 8: frame source>

```
<html>
  <head>
    <title>Demonstrating Multiple Frames</title>
  </head>
  <body bgcolor = "white">
    <table align = "center">
      <tr>
        <td width = "100"><a href = "HTMLFrameDemo.html"
          target = "main">Database</a></td>
        <td width = "130"><a href = "http://www.prenhall.com"
          target = "main">Prentice Hall</a></td>
        <td width = "150"><a href = "http://www.ca.gov"
          target = "main">California</a></td>
      </tr>
    </table>
  </body>
</html>
```

<Side Remark: referencing main>

The name attribute in the `<frame>` tags in Listing 17 references the main frame defined in Listing 16. When you click on Database, Prentice Hall, and California, the hyperlinked content will be displayed in the main frame, as shown in Figures 15 and 16.

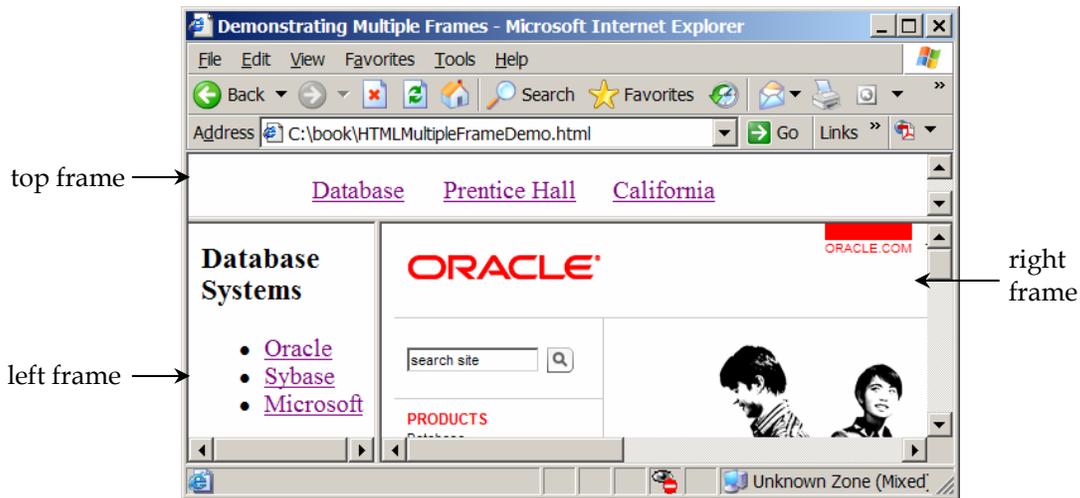


Figure 15

Multiple frames are displayed in the browser.

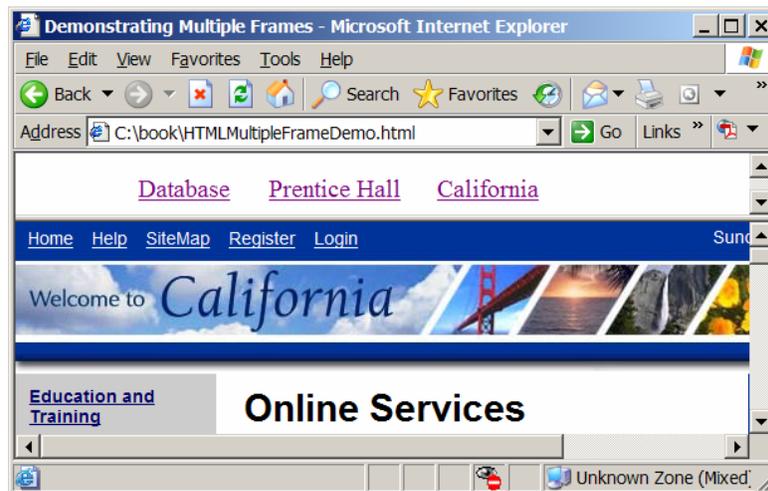


Figure 16

Clicking on California displays the content to the main frame.

15 Forms

HTML forms are for submitting data from a Web browser to a Web server. Once the server receives the data, it runs a program to process the data. Listing 18 demonstrates how to use the forms. The form is displayed as shown in Figure 17.

Listing 18 student_registration_form.html (HTML Form Demo)

*****PD: Please add line numbers in the following code*****
*****Layout: Please layout exactly. Don't skip the space. This is true for all source code in the book. Thanks, AU.**
<Side Remark line 9: form tag>

<Side Remark line 12: label>
 <Side Remark line 13: text field>
 <Side Remark line 21: radio button>
 <Side Remark line 26: combo box>
 <Side Remark line 35: list>
 <Side Remark line 44: check box>
 <Side Remark line 51: text area>
 <Side Remark line 54: submit button>
 <Side Remark line 35: reset button>

```

<!--An HTML Form Demo -->
<html>
  <head>
    <title>Student Registration Form</title>
  </head>
  <body>
    <h3>Student Registration Form</h3>

    <form action = "http://localhost:8080/liangweb/GetParameters"
      method = "get">
      <!-- Name text fields -->
      <p><label>Last Name</label>
        <input type = "text" name = "lastName" size= "20" />
        <label>First Name</label>
        <input type = "text" name = "firstName" size = "20" />
        <label>MI</label>
        <input type = "text" name = "mi" size = "1" /></p>

      <!-- Gender radio buttons -->
      <p><label>Gender:</label>
        <input type = "radio" name = "gender" value = "M" checked /> Male
        <input type = "radio" name = "gender" value = "F" /> Female</p>

      <!-- Major combo box -->
      <p><label>Major</label>
        <select name = "major" size = "1">
          <option value = "CS">Computer Science</option>
          <option value = "Math">Mathematics</option>
          <option>English</option>
          <option>Chinese</option>
        </select>

      <!-- Minor list -->
      <label>Minor</label>
      <select name = "minor" size = "2" multiple>
        <option>Computer Science</option>
        <option>Mathematics</option>
        <option>English</option>
        <option>Chinese</option>
      </select></p>

      <!-- Hobby check boxes -->
      <p><label>Hobby:</label>
        <input type = "checkbox" name = "tennis" /> Tennis
        <input type = "checkbox" name = "golf" /> Golf
  
```

```



```

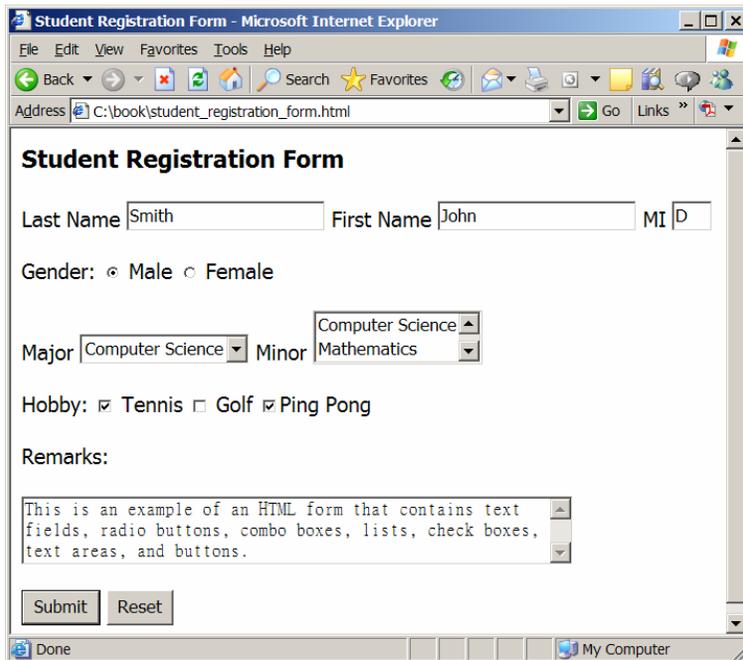


Figure 17

An HTML form may contain text fields, radio buttons, combo boxes, lists, check boxes, text areas, and buttons.

The following HTML tags are used to construct HTML forms:

<Side Remark: <form>>

<Side Remark: action>

<Side Remark: method>

- **<form> ... </form>** defines a form body. The attributes for the **<form>** tag are **action** and **method**. The **action** attribute specifies the server program to be executed on the Web server when the form is submitted. In this case the URL of the program is `http://localhost:8080/liangweb/GetParameters`. The **method** attribute is either **get** or **post**. If the GET method is used, the data in the form are appended to the request string as if they were submitted using a

URL. If the POST method is used, the data in the form are packaged as part of the request file. The server program obtains the data by reading the file.

<Side Remark: <label>>

- <label> ... </label> simply defines a label.

<Side Remark: <input>>

- <input> defines an input field. The attributes for this tag are type, name, value, checked, size, and maxlength. The type attribute specifies the input type. Possible types are text for a one-line text field, radio for a radio button, and checkbox for a check box. The name attribute gives a formal name for the attribute. This name attribute is used by the servlet program to retrieve its associated value. The names of radio buttons in a group must be identical. The value attribute specifies a default value for a text field and text area. The checked attribute indicates whether a radio button or a check box is initially checked. The size attribute specifies the size of a text field, and the maxlength attribute specifies the maximum length of a text field.

<Side Remark: <select>>

- <select> ... </select> defines a combo box or a list. The attributes for this tag are name, size, and multiple. The size attribute specifies the number of rows visible in the list. The multiple attribute specifies that multiple values can be selected from a list. Set size to 1 and do not use a multiple for a combo box.

<Side Remark: <option>>

- <option> ... </option> defines a selection list within a <select> ... </select> tag. This tag may be used with the value attribute to specify a value for the selected option (e.g., <option value = "CS">Computer Science). If no value is specified, the selected option is the value.

<Side Remark: <textarea>>

- <textarea> ... </textarea> defines a text area. The attributes are name, rows, and cols. The rows and cols attributes specify the number of rows and columns in a text area.

16 XHTML

XHTML is almost identical to HTML, but it is a stricter and

cleaner version of HTML. This section discusses the important differences between XHTML and HTML.

<Side Remark: lowercase tags>

Tags and attributes are not case-sensitive in HTML, but are case-sensitive in XHTML. All tags and attributes in XHTML are lowercase.

<Side Remark: no overlapping tags>

Elements must be properly nested in XHTML. A tag can be embedded inside another tag; for example, all tags are embedded within `<html>` and `</html>`. However, tags cannot overlap in XHTML; it would be wrong, for instance, to use `<i>bold and italic</i>`; the correct use should be `<i>bold and italic</i>`.

<Side Remark: closing tags>

In HTML, some tags (e.g., the `<p>`, `
`, and `<hr>` tag) do not require closing tags. But in XHTML, every start tag must have a closing tag, or the start tag must end with `/>`. For example, the following is correct in HTML, but not in XHTML.

```
<p>a new paragraph  
A new line<br>  
<hr size = "3">  
<p>another paragraph
```

In XHTML, you have to rewrite the code as follows:

```
<p>a new paragraph</p>  
A new line<br />  
<hr size = "3" />  
<p>another paragraph</p>
```

<Side Remark: quoted attributes>

In XHTML, attribute names must be lowercase and values must be quoted. Furthermore, each attribute must have a name and a value in XHTML. For example, the following is correct in HTML, but not in XHTML.

```
<hr SIZE = 4>  
<input type = "radio" name = "gender" value = "M" checked /> Male  
<img src = "image/us-map.gif" ismap>
```

In XHTML, you have to rewrite it as:

```
<hr size = "4">  
<input type = "radio" name = "gender" value = "M"  
checked = "checked" /> Male  
<img src = "image/us-map.gif" ismap = "ismap">
```

<Side Remark: validating XHTML>

Every XHTML document must specify a DOCTYPE so that the document can be properly validated against a Document Type Definition (DTD). There are three types: *strict*, *transitional*, and *frameset*.

The strict DTD includes elements and attributes that have not been deprecated or do not appear in framesets:

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

The transitional DTD includes everything in the strict DTD plus deprecated elements and attributes:

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

The frameset DTD includes everything in the transitional DTD plus frames as well:

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

Listing 19 gives an XHTML document.

Listing 19 XHTMLDemo.html (XHTML Demo)

*****PD: Please add line numbers in the following code*****
*****Layout: Please layout exactly. Don't skip the space. This is true for all source code in the book. Thanks, AU.**

<Side Remark line 1: DOCTYPE>

<Side Remark line 4: <html>>

<Side Remark line 5: <head>>

<Side Remark line 8: <body>>

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

```
<html>
```

```
<head>
```

```
<title>A Simple XHTML Page</title>
```

```
</head>
```

```
<body>
```

```
<i>Welcome to</i> <b>HTML</b>. Here is a list of popular
```

```
<font color = "red">Web browsers.</font>
```

```
<ul>
```

```
<li>Internet Explorer</li>
```

```
<li>Netscape</li>
```

```
<li>Mosaic</li>
```

```
</ul>
```

```
<hr size = "3" />
```

```
Created by <a href = "www.cs.armstrong.edu/liang">
```

```
Y. Daniel Liang</a>.
```

```
</body>
```

```
</html>
```

Every XHTML document must specify a DOCTYPE at the beginning

of the document. In this case, a transitional document type is specified in lines 1-2.

<Side Remark: validating XHTML>

The W3C provides a free service for validating whether a document conforms to the XHTML standard. To use the service, go to validator.w3.org.

A valid transitional XHTML may contain deprecated tags and attributes. A strict XHTML does not allow deprecated tags and attributes. Every paragraph in a strict XHTML must be enclosed in the `<p>` tag.

<Side Remark: deprecated tags>

The `<basefont>`, ``, `<center>`, `<strike>`, and `<u>` tags are deprecated in XHTML and no longer supported in strict XHTML. You should use CSS (Cascading Style Sheet) to achieve the same effect.

<Side Remark: `<applet>` deprecated>

The `<applet>` tag is deprecated and replaced by the `<object>` tag. However, the `appletviewer` utility in Java 5 does not support it. So, you should continue to use the `<applet>` tag in the near future.

<Side Remark: deprecated attributes>

The attributes in many tags are deprecated and not supported in strict XHTML. For example, the `size` attribute in the `<hr>` is deprecated, and the `type` attribute in the `` is also deprecated.

Listing 19 is not a strict XHTML document, because it contains deprecated tags (``) and attributes (`size` in the `<hr>` tag), and also it contains a paragraph in lines 9-10 and in lines 17-18 without the `<p>` tag. Listing 20 revises Listing 19 to make the document strict XHTML.

Listing 20 StrictXHTMLDemo.html (Strict XHTML Demo)

*****PD: Please add line numbers in the following code*****
*****Layout: Please layout exactly. Don't skip the space. This is true for all source code in the book. Thanks, AU.**

<Side Remark line 1: strict XHTML>

<Side Remark line 9: paragraph tag>

<Side Remark line 17: paragraph tag>

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

```
<html>
<head>
<title>A Strict XHTML Page</title>
```

```
</head>
<body>
  <p><i>Welcome to</i> <b>HTML</b>. Here is a list of popular
  Web browsers.</p>
  <ul>
    <li>Internet Explorer</li>
    <li>Netscape</li>
    <li>Mosaic</li>
  </ul>
  <hr />
  <p>Created by <a href = "www.cs.armstrong.edu/liang">
  Y. Daniel Liang</a>.</p>
</body>
</html>
```