

AN HTML TUTORIAL

Java applets are embedded in HTML files. 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, 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 thirty 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. At present there are no absolute, unifying HTML standards. Different vendors have rushed to introduce their own features, interpretable by their proprietary browsers. However, the differences aren't significant. This tutorial introduces some frequently used HTML features that have been adopted by most browsers.

There are many easy-to-use authoring tools for creating Web pages. For example, you can create HTML files using Microsoft Word. The Internet Explorer and Netscape Navigator have simple authoring tools to let you create and edit HTML files. Microsoft FrontPage is a comprehensive and fully loaded tool that enables you to design more sophisticated Web pages. Authoring tools greatly simplify the task of creating Web pages but do not support all features of HTML. Since you will probably end up editing the source text produced by whatever tool you adopt, it is imperative to know the basic concept of HTML. In this tutorial, you will learn how to use HTML to create your own Web pages.

F.1 Getting Started

Let us begin with an example that demonstrates the structure and syntax of an HTML document.

Example F.1 An HTML Example

The following HTML document displays a message and a list of Web browsers: Netscape, Internet Explorer, and Mosaic. You may use any text editor, such as Microsoft NotePad on Windows, to create HTML documents, as long as it can save the file in ACSII text format.

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

Assume that you have created a file named **ExampleF1.html** for this HTML 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 file name. 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 F.1.

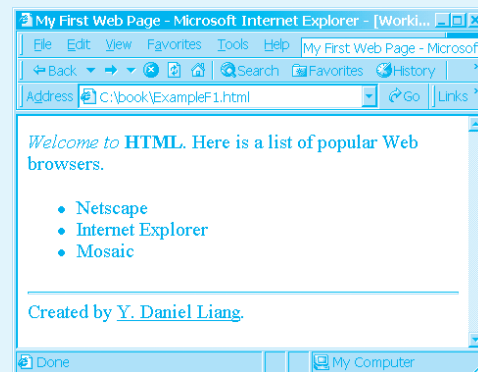


Figure F.1 The HTML page is rendered by a Web browser.



NOTE

HTML file names are case-sensitive on UNIX but not on other operating systems. HTML files end with .html or .htm.

**NOTE**

The same document may be rendered differently depending on the capabilities of the Web browser. Regardless of the differences, the power of HTML is that your document can be viewed on a variety of browsers and on most platforms, and can be formatted to suit any reader. This tutorial uses Internet Explorer 5.0 for illustrations.

What makes *Welcome to* appear in italic in Figure F.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 Example F.1, the attribute `size` in the tag `<hr>` defines the size of the bar as 3 inches.

Most 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 (`/`). A few tags do not have end tags. For example, `<hr>`, a tag to draw a line, has no corresponding end tag.

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

**TIP**

Tags are not case-sensitive, but it is good practice to use case consistently for clarity and readability. This tutorial uses lowercase for 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.

F.2 Structure Tags

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>
<!-- .. document body .. -->
</body>
</html>
```

Here the special starting tag `<!--` and ending tag `-->` are used to enclose comments in the HTML document. The comments are not displayed.



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.

F.3 Text Appearance 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*.

F.3.1 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 F.1 lists the content-based tags and provides examples of their use.

TABLE F.1 Using Content-Based Tags

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

F.4 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)

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

F.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

TABLE F.2 Using Physical Tags

<i>Tag</i>	<i>Example</i>	<i>Display</i>
<code><i> . . . </i></code>	<code><i>italic</i></code>	<i>italic</i>
<code> . . . </code>	<code>bold</code>	bold
<code><u> . . . </u></code>	<code><u>underline</u></code>	underline
<code><tt> . . . </tt></code>	<code><tt>monospace</tt></code>	monospace
<code><strike> . . . </strike></code>	<code><strike>strike </strike></code>	strike
<code><blink> . . . </blink></code>	<code><blink>blink </blink></code>	blink (causes it to blink)

line-break tag (`
`), a paragraph start tag (`<p>`), a preformat tag (`<pre>`), and a block-quote tag (`<blockquote>`).

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.

The line-break tag `
` tells the browser to start displaying from the next line. This tag has an optional end tag `
`.

The paragraph-start tag `<p>` signals the start of a paragraph. This tag has an optional end tag `</p>`.

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

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

Example F.2 HTML Source Code Using Structure Tags

The following HTML source code illustrates the use of paragraph tags. The text the code creates is displayed in Figure F.2.

```
<html>
<head>
<title>Demonstrating Paragraph Tags</title>
</head>
<body>
<!-- Example F.2 -->
<h1 align=right>h1: Heading 1</h1>
<h3 align=center>h3: Heading 3</h3>
<h6 align=left>h6: Heading 6</h6>
<p>
```



F.6 Font, Size, and Color Tags

With HTML you can specify font size and colors using font tags. There are two font tags: `<basefont>` and ``.

The `<basefont>` tag is typically placed in the head of an HTML document, where it sets the base font size for the entire document. However, it may appear anywhere in the document, and it may appear many times, each time with a new size attribute. 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.

The `` tag allows you to specify the size and color of the enclosed text, using size and color attributes. The size attribute is the same as the one for the `<basefont>` tag. 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 like red, yellow, blue, or orange.

Example F.3 Testing Font Tags

The following HTML source code illustrates the use of the `<basefont>` and `` tags. The text it creates is displayed in Figure F.3.

```
<html>
<head>
<title>Demonstrating Fonts, Size and Color</title>
</head>
<basefont size=6>
<body bgcolor=white>
<!-- Example F.3 -->
basefont<br>
<font size=7 color=blue>blue7</font><br>
<font size=3 color=#FF0000>red3</font><br>
</body>
</html>
```

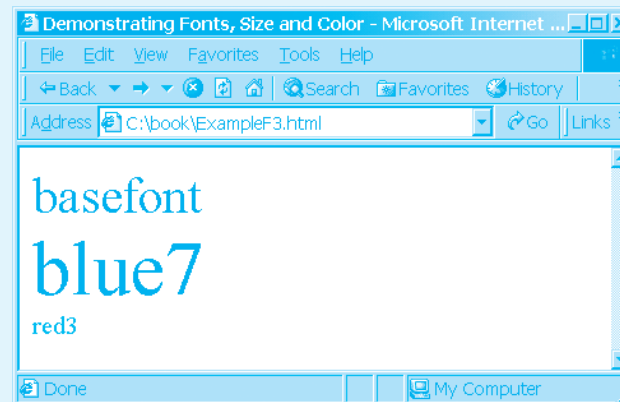


Figure F.3 Font and color tags specify fonts and colors in HTML pages.

F.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. Example F.1 contains an unordered list of three Web browsers.

F.7.1. Ordered Lists

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, . . .

F.7.2 Unordered Lists

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.

F.7.3 Definition Lists

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.

Example F.4 Using Various List Tags

This example illustrates the use of tags for ordered lists, unordered lists, definition lists, and nested lists. The output of the following code is displayed in Figure F.4.

```
<html>
<head>
<title>Demonstrating List Tags</title>
</head>
<body bgcolor=white>
<!-- Example F.4 List Tags -->
<center><b>List Tags</b></center>
An ordered List
<ol type=A></OL>
  <li>Chapter 1: Introduction to Java
  <li>Chapter 2: Java Building Elements
  <li>Chapter 3: Control Structures
</ol>
An unordered List
<ul type=square>
  <li>Apples
  <li>Oranges
  <li>Peaches
</ul>
Definition List
<dl>
  <dt>What is Java?
  <dd>An Internet programming language.
</dl>
</body>
</html>
```

continues

Example F.4 Continued

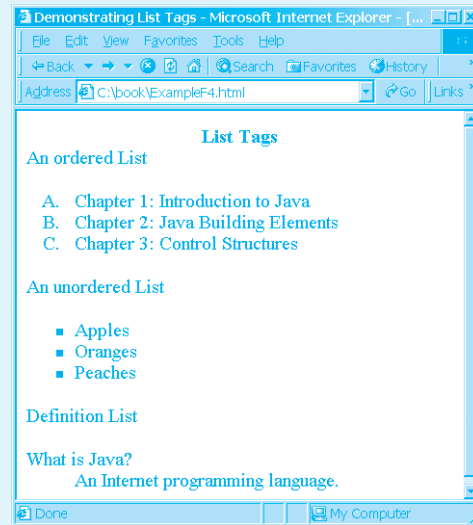


Figure F.4 HTML list tags can display ordered lists, unordered lists, and definition lists.

F.8 Table Tags

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>` to display a caption for the table, and `<th>` to display column headers.

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

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

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

Example F.5 Illustration of Table Tags

This example creates an HTML table. The output of the code is displayed in Figure F.5.

```
<html>
<head>
<title>Demonstrating Table Tags</title>
</head>
<body bgcolor=white>
<!-- Example F.5 Table Tags -->
<center>Table Tags</center>
<br>
<table border=2>
<caption>This is a Table</caption>
<tr>
<th>Table heading</th>
<td>Table data</td>
</tr>
<tr>
<th valign=bottom>Second row
<td>Embedded Table
<table border=3>
<tr>
<th>Table heading</th>
<td align=right>Table data</td>
</tr>
</table>
</td>
</tr>
</table>
</body>
</html>
```

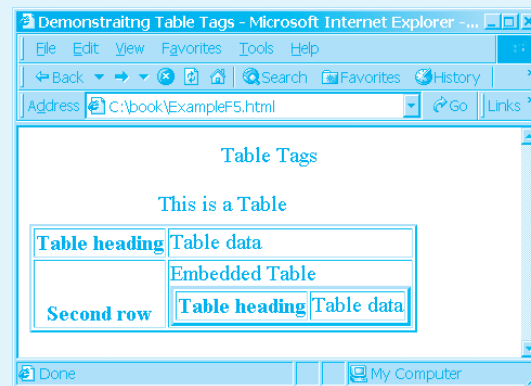


Figure F.5 Table tags are useful for displaying tables in HTML pages.

F.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.

F.9.1 Linking Documents on Different Computers

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><a href="http://www.sybase.com">Sybase</a>
<li><a href="http://www.microsoft.com">Microsoft</a>
</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 **liang.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 Microsystems'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 80 by default.

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

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

F.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 the directory `liang/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.

F.9.3 Jumping Within the Same Document

HTML offers navigation within the same document. This is helpful for direct browsing of interesting segments of the document.

Example F.6 Navigation Within the Same Document

This example shows a document with three sections. The output of the following code is shown in Figure F.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.

```
<html>
<head>
<title>Demonstrating Link Tags</title>
</head>
<body>
```

continues

Example F.6 Continued

```
<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"><b>Section 1</b>: Introduction</a></h3>
an introductory paragraph

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

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

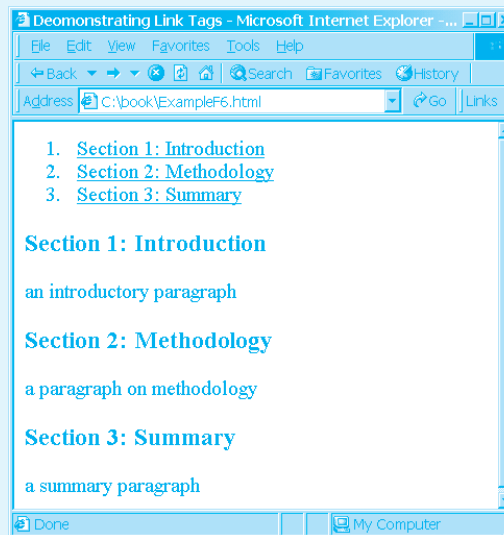


Figure F.6 Hyperlink tags link documents.

F.10 Embedding Graphics

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 *horizontal bar tags* and *image tags*.

F.10.1 Horizontal Bar Tags

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-aligned, centered, or right-aligned.

Example F.7 Illustration of Horizontal Bar Tags

This example illustrates the use of the `size`, `width`, and `align` attributes in horizontal bar tags. The output of the following code is shown in Figure F.7.

```
<html>
<head>
<title>Demonstrating Horizontal Rules</title>
</head>
<body bgcolor=white>
<!.. Example F.7 Horizontal Rule ..>
<center>Horizontal Rules</center>
<hr size=3 width=80% align=left>
<hr size=2 width=20% align=right noshade>
<hr>
</body>
</html>
```

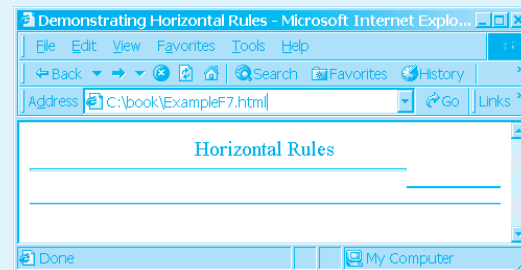


Figure F.7 Horizontal bar tags are often used to separate contents in documents.

F.10.2 Image Tags

The image tag, ``, lets you reference and insert images into the current text. The syntax for the tag is:

```
<img src=URL alt=text align = [top | middle | bottom | texttop ]>
```

Most browsers support GIF and JPEG image formats. Format is an encoding scheme to store images. The attribute `src` specifies the source of the image. The attribute `alt` specifies an alternative text message to be displayed in case the client's browser cannot display the image. The attribute `alt` is optional; if it is omitted, no message is displayed. The attribute `align` tells the browser where to place the image.

Example F.8 Illustration of Image Tags

This example creates a document with image tags. The output of the code is shown in Figure F.8.

```
<html>
<head>
<title>Demonstrating Image Tags</title>
</head>
<body bgcolor=white>
<!-- Example F.8 Image Tags -->
<center>Image Tags</center>

</body>
</html>
```

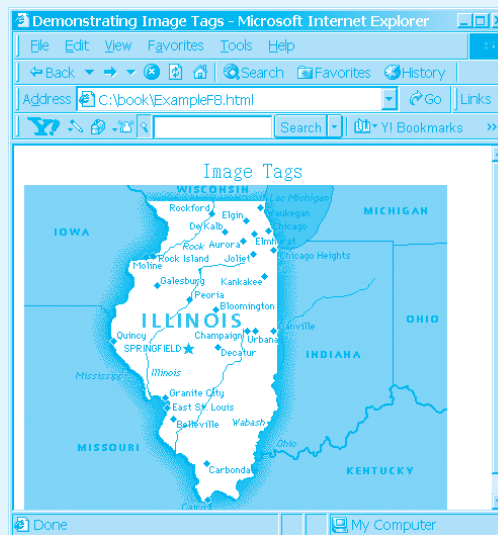


Figure F.8 Image tags display images in HTML pages.

F.11 More on HTML

This tutorial is not intended to be a complete reference manual on HTML. It does not mention many interesting features, such as *forms* and *frames*. You will find dozens of books on HTML in your local bookstore. *Special Edition Using HTML* by Mark Brown and John Jung, published by QUE, is a comprehensive reference; it covers all the new HTML features supported by Netscape Navigator and Microsoft Internet Explorer. *HTML Quick Reference* by Robert Mullen, also published by QUE, contains all the essential information you need to build Web pages with HTML in 100 pages. Please refer to these and other books for more information. You can also get information online at the following Web sites:

www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html

www.w3.org/pub/WWW/MarkUp/

www.mcli.dist.maricopa.edu/tut/lessons.html

www.netscape.com/assist/net_sites/frames.html