2 Jump Start: Browsing and Publishing

Leaving Georgia's beautiful azaleas and red clay behind, it was a thrill to get into our second state. We had already given away a substantial amount of our gear to reduce our pack weights—most of it to a hiker who was out soul searching as to whether or not to go through with his upcoming marriage. I had gotten rid of a couple of cooking pans, silverware, a book, a trail seat pad, and some food. I was to soon regret off-loading the food. We totally ran out of food just before Rainbow Springs, North Carolina, our next resupply stop. In fact, we had been out of food for a while and were feeling so desperate that we had already licked out the inside of a couple of plastic bags that had contained powdered gatorade. This hardly provided sufficient calories to keep hiking. We were sitting exhausted at roadside a mere five miles from our resupply point when trail angel "Ron" appeared and offered us each an apple. We quickly ate them. We exchanged looks that said "If you are not going to eat the core of that apple, give it to me and I will. I don't want to have to pick it out of the dirt and dust it off if you discard it." We both ate everything except the stem. Ron gave us each another apple (same result) and two sodas apiece. We thanked him heartily. We pushed on having learned several valuable lessons during this stretch.

North Carolina
193 miles

Chapter 2

2.1 INTRODUCTION

This chapter provides the background necessary to start using the World Wide Web and to create Web pages. We will introduce the following topics:

Objectives

- Web browsers
- Web surfing
- *HyperText Markup Language* (HTML)
- Web page installation
- Web page setup
- HTML formatting and hyperlink creation

2.2 Browser Bare Bones

We begin with browser essentials to get you up and surfing the Web. Even if you have already been browsing the Web, please read on; a few pointers may be useful.

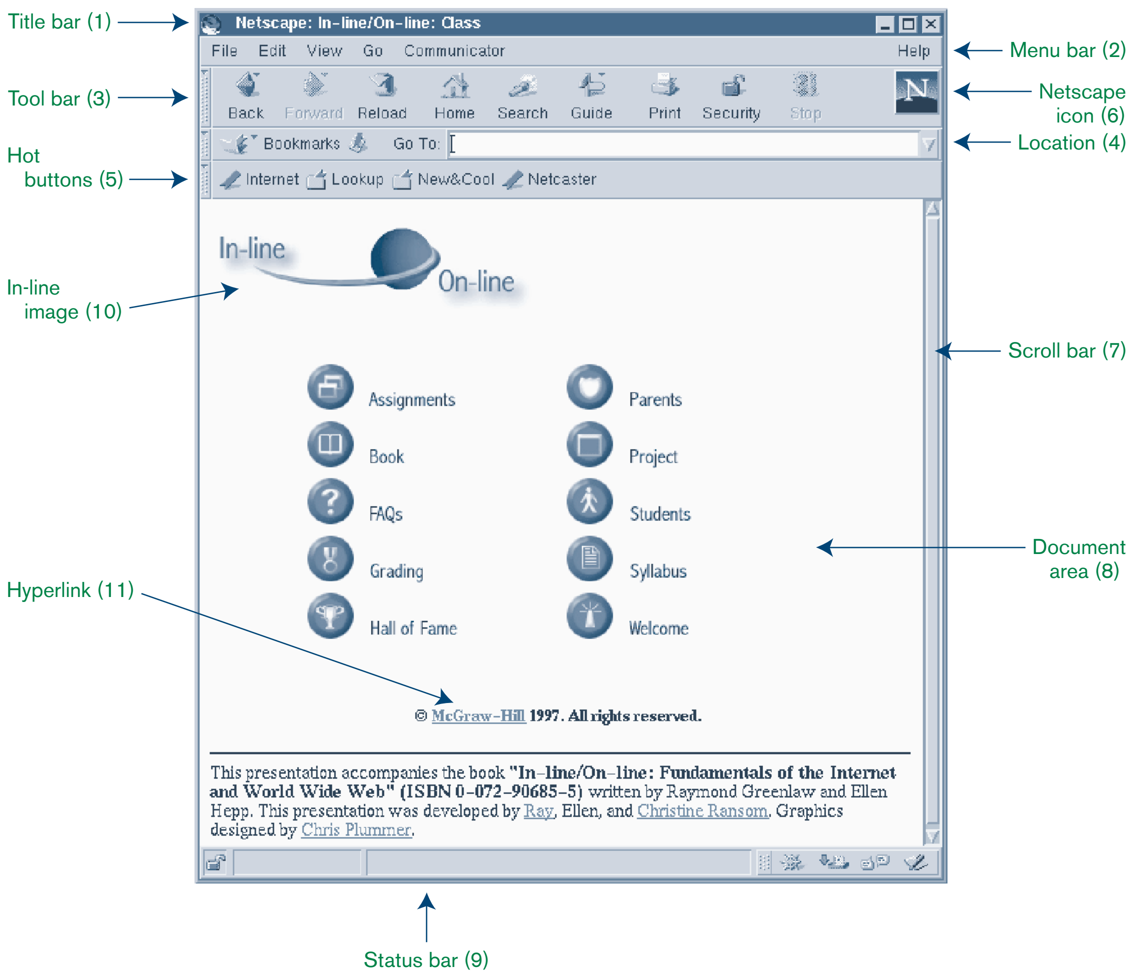
Very few agreed-upon precise definitions have been given for new terms involving the Internet, such as *Web browser*. A Web browser is one of many software applications that function as the interface between a user and the Internet. The browser not only sends messages to Web servers to retrieve your page requests, but also *parses* and *renders* the HTML code once it arrives. That is, the browser interprets the code and displays the results on the screen. Many browsers have built-in mail clients and/or newsreaders. Additionally, auxiliary programs such as *helper applications* and *plug-ins* can be configured into the browser. (We will describe these features in more detail later.) It is safe to assume that browsers will continue to grow in complexity and functionality in the foreseeable future.

Popular browsers include Netscape Navigator, Microsoft's Internet Explorer, Mosaic, and Lynx. The first few are *graphical-based Web browsers*, whereas Lynx is a *text-only browser*. This makes Lynx (see Section 4.7) very fast, since graphics often cause Web pages to load more slowly. For our presentation, we will describe the Netscape Navigator, although most browsers have similar features (the exception being Lynx).

2.2.1 Browser Window Terminology

Figure 2.1 illustrates a sample browser window. The different components of the window are numbered. We will provide a name and a short explanation of each part and will then discuss the parts in more detail.

1. *Title bar*—The location where the document's title is displayed.
2. *Menu bar*—The place showing the headings of the main pull-down command menus.
3. *Toolbar*—The area providing access to a number of single-mouse-click commands.
4. *Location*—The area where the *Uniform Resource Locator* (URL) (discussed in Section 2.3.2) of the document is displayed. The item in the location field usually begins with `http://`, although you may also see `file://`, `https://`, `ftp://`, `gopher://`, `javascript:`, `mailto:`, `news:`, or `telnet://`.
5. *Hot buttons*—Single-click buttons that provide a number of convenient features.
6. *Netscape icon*—An image that shows movement to indicate when a document is being downloaded from the Internet.
7. *Scroll bar*—Arrows that allow the user to display a different part of a "large" document.
8. *Document area*—The part of the window that is used for displaying the currently loaded document.
9. *Status bar*—A field used to convey helpful (and current) information to the user, such as a URL or a programmer-specified message.
10. *In-line image*—An image appearing within a document.
11. *Hyperlink*—A highlighted (usually underlined) part of a document that, when selected, causes the browser to retrieve and display a (new) document.



© McGraw-Hill 1997. All rights reserved.

This presentation accompanies the book "In-line/On-line: Fundamentals of the Internet and World Wide Web" (ISBN 0-072-90685-5) written by Raymond Greenlaw and Ellen Hepp. This presentation was developed by Ray, Ellen, and Christine Ransom. Graphics designed by Chris Plummer.

FIGURE 2.1
Sample screen illustrating the terminology associated with a browser window.

2.2.2 Menu Bar

Useful Item

Looking at the Netscape[1] window in Figure 2.1, you will notice the items **File**, **Edit**, **View**, **Go**, **Communicator**, and **Help** in the menu bar. (You will see different items, such as **Bookmarks**, **Options**, **Directory**, and **Window**, in earlier versions of Netscape.) As is customary when using

1 It is common to refer to the Netscape Navigator by the name of the company, Netscape, that developed it. Netscape's terminology is very confusing. With version 4.x of their browser, it is still the Navigator, part of a suite called *Communicator*, but many people now call the browser Communicator.

menus, menu items that are not available to the current configuration of the browser are shown in a lighter color gray ("grayed out").

- The **File** menu item will allow you to launch a new browser, utilize the Netscape mailer, open a new URL, open a local file, save a file, print a screen, or exit the browser. The **Send Link** option provides a convenient way to email the URL of the page you are currently visiting to someone else. If you are surfing the Web from someone else's account, this is a good way to send yourself a URL so you can bookmark it later.
- The **Edit** button provides basic text editing capabilities.
- The **View** menu item is especially useful because it allows you to view the HTML source code of the document being displayed. This is a great way to learn how someone else achieved a certain layout on their Web page. Other functions allow you to reload a page, load images, refresh a page, obtain document information, view *frame* source code, and view frame information.
- The **Go** menu item displays a list of Web pages that you have visited and allows you to select any one to return to. It also provides **Back**, **Forward**, **Home**, and **Stop** loading options.
- In Netscape 4.x, the **Communicator** menu contains the items **Collabra Discussions** (Netscape's *collaborative computing* software), **Page Composer** (Netscape's *HTML editor*), **Message Center**, **Bookmarks**, and **History**, among others.
- The **Bookmarks** menu item lets you add a bookmark or directly select a (previously saved) bookmark. A *bookmark* is simply a saved Web location (i.e., URL). URLs are often cumbersome to type. For Web pages you visit often or just want to remember, the bookmark mechanism is a handy tool.
- The **Help** menu item provides Netscape help and information.
- (Netscape 3.x) The **Options** menu item has a number of features that allow you to customize your browser. For example, you can toggle the images setting (see page 49), specify *cache* size, and allow or disallow *cookies* to be written.
- (Netscape 3.x) The **Directory** menu item provides a vertical list of the hot buttons.

- (Netscape 3.x) The **Window** menu item provides access to news, mail, your address book, bookmarks, and a history mechanism.

2.2.3 Toolbar

The toolbar is located under the title bar and contains buttons for **Back**, **Forward**, **Reload**, **Home**, **Search**, **Guide**, **Print**, **Security**, and **Stop**. (You will see different items, such as **Images**, **Open**, and **Find**, in Netscape 3.x.)

- **Back** A browser normally saves copies of the pages you have viewed in a *cache*. Think of a cache as local computer disk space from which the browser can quickly retrieve a document. For example, suppose you load a Web page with lots of graphics from a Web site located across the country. If this document is cached locally, then the next time you request it, the browser can load the copy stored in cache. The document can thus be loaded much faster. How much disk space should be allocated to cache? This is an option that the user can set. What if a document is updated but you keep retrieving the old cached version? You can try to reload the document and, if necessary, clear the cache. Many other interesting issues involving caches are dealt with in introductory computer science courses.

  Suppose you just visited Web pages A, B, C, D, and E, in that order. At this point, hitting the **Back** button would take you from E to D. Clicking on **Back** again takes you to C, and so on. The **Back** button allows you to access the most recently visited page without typing in its URL. The page is usually loaded quickly, since it is available from cache.

- **Forward** The **Forward** button allows you to page forward in much the same way that the **Back** button operates.

- **Reload** The most current version of a Web page can be loaded by clicking on the **Reload** button. This is particularly useful if you have just modified the source code for a page and want to view and check the changes.

- **Home** The user can specify what Web page to load when the browser is first activated. This page is often called the *homepage*. The **Home** button will load the homepage that has been designated. The default page is often set to the Web page of the company that developed the browser.

- **Search** Clicking on the **Search** button brings up one of the many useful search tools that Netscape "knows" about. Once the search tool is loaded, you can use it to search the Internet.

- **Guide** The **Guide** button leads you to a mini information center (provided by Netscape) from which you can locate all kinds of useful items. The information displayed is updated frequently.

- **Print** You can obtain a hardcopy of the currently displayed Web page by clicking the **Print** button.

- **Security** The **Security** button allows you to examine and specify security options.

- **Stop** The **Stop** button is used to stop the transfer of a Web page. This can be handy if you realize that you have selected the wrong link, or if the page you selected is going to take too much time to load. This button also allows you to stop endlessly looping animated GIFs.

Hot Topic

- **Images** (Netscape 3.x) The **Images** button lets you "toggle" the state of image downloading; that is, if the browser is currently downloading images, selecting the **Images** button tells the browser not to download images. If the browser is not downloading images, then selecting the **Images** button tells the browser to download images. Since images require a lot of storage and therefore a lot of time to download, this button comes in handy when you do not need or wish to download images.

Hot Topic

- **Open** (Netscape 3.x) The **Open** button provides you with a dialog box in which to type a URL. When you hit the return, the browser requests and then renders the Web page you specified.

- **Find** (Netscape 3.x) The **Find** button initiates a search within the current Web page for a word or phrase that you specify. If the pattern of the word or phrase occurs in the document, the browser scrolls the page to the first occurrence of that pattern and then highlights it. If the pattern is not found, the browser will usually ask if you want to search in the reverse direction. (This function is now located in the Menu bar under the **Edit** entry in Netscape 4.x.)

2.2.4 Hot Buttons

Beneath the location area are the hot buttons (also called *directory buttons*) that Netscape provides. Other browsers provide their own versions of these buttons. These buttons include:

Go On-Line

- **Internet** This button has the same effect as the **Guide** button of the tool bar.
- **Lookup** This button contains two options: **People** and **Yellow Pages**.
  - **People** Various search programs that are available to locate an individual.
  - **Yellow Pages** Various search programs that are available to locate a business.
- **New&Cool** This button contains two options: **What's New** and **What's Cool**.
  - **What's New** A list of new, interesting Web pages.
  - **What's Cool** A selected list of "cool" Web pages.
- **Netcaster** This button takes you to information about Netscape's Netcaster product, which allows you to open a "channel" to receive a continuous flow of information to your computer.

The following buttons appeared in earlier versions of Netscape.

- **Destinations** This allowed access to a list of "cool" hyperlinks.
- **Net Search** This provided a quick way to access a variety of search programs.
- **Software** This button accessed information about Netscape software that was currently available for downloading.

2.2.5 Hyperlinks

Let us elaborate on the important concept of a *hyperlink*. Hyperlinks are clickable text and/or images that generally cause the downloading and rendering of a new HTML document. Hyperlinks are often displayed in a different text color than the remainder of the document, and they are usually underlined to make them stand out. An image serving as a link may have a border around it that is the same color as other hyperlinks on the page. In either case, moving the mouse over a hyperlink (termed *mousing over* a hyperlink) will cause the mouse cursor to change appearance, perhaps from an arrow to a hand.

Useful Item

The location (URL) of the link being moused over will be displayed in the status line. It is very helpful if you understand URLs. An experienced user, upon seeing a URL, will know where the document is stored and approximately how long it should take to download (assuming an educated guess can be made as to the size of the document).

EXERCISES 2.2

Browser Bare Bones

1. Print the source code for a Web page related to the Appalachian Trail. Print the browser screen of the same Web page and label the different components of the page on the hardcopy.
2. Explain the types of facts that are available when viewing document information. For example, for an HTML document, can you tell how many bytes long it is or when it was created? What about the size of an image?
3. Is there a difference between typing in a URL in the location area versus selecting the **Open Page** option of the **File** button?
4. Experiment with the history mechanism of the browser. Write a paragraph explaining how it works.
5. What options are available for customizing your browser? Describe them in a few paragraphs.
6. Skim through the on-line help for your browser. Summarize in two paragraphs what type of information is available.
7. Using the hot buttons, try to track down Grandma Gatewood or Earl Schaffer on the Web. Describe how you proceeded and whether or not you were successful.

2.3 Coast-to-Coast Surfing

We are now ready to start using the browser to discover information on the World Wide Web. As is customary, we have been shortening the phrase World Wide Web to “Web.” Other common short forms are WWW, W3, and W^3.

The Web provides a means of accessing an enormous collection of information, including text, graphics, audio, video, movies, and so on. One of the most exciting aspects of the Web is that information can be accessed in a nonlinear and experimental fashion. Unlike reading a book by flipping to the next page in sequential order, you can "jump" from topic to topic via hyperlinks. This nonlinear approach to information gathering, or browsing, is sometimes referred to as "surfing the Web." As a reader, you have the option to select what to explore next. Different readers will proceed through the same Web presentations in totally different ways, depending on their backgrounds, needs, and personalities.

2.3.1 Web Terminology

Go On-Line

Web surfing is a great way to become familiar with the Web. To begin our discussion of Web surfing, we first introduce and review some common Web terminology[2]:

Hot Topic

- **Page** or **Web page** A file that can be read over the World Wide Web.
- **Pages** or **Web pages** The global collection of documents associated with and accessible via the World Wide Web.
- **Hyperlink** A string of clickable text or a clickable graphic that points to another Web page or document. When the hyperlink is selected, another Web page is requested, retrieved, and rendered by the browser.
- **Hypertext** Web pages that have hyperlinks to other pages. More generally, any text having nonlinear links to other text.
- **Browser** A software tool used to view Web pages, read email, and read newsgroups, among other things. Browsers are also called *Web clients*.
- **Multimedia** Information in the form of graphics, audio, video, or movies. A multimedia document contains a media element other than just plaintext.
- **Hypermedia** Media with links and navigational tools.
- **Uniform Resource Locator** A string of characters that specify the address of a Web page.
- **Surfer** A person who spends time exploring the World Wide Web.

2 In a number of sections in this book, we introduce special terminology. The terms are presented in a logical order, rather than an alphabetical order. The terms may be found in alphabetical order in the glossary and index.

- **Web presentation** A collection of associated and hyperlinked Web pages. Usually, there is an underlying theme to the pages. For example, a Web presentation for a company may describe facts about the company, its employees, its products, and the method for ordering the products on-line.
- **Webmaster** A person who maintains, creates, and manages a Web presentation, often for a business, organization, or university. This person usually "signs" Web pages, so that questions and comments can be sent to them.
- **Web manager** Synonym for Webmaster.
- **Web site** An entity on the Internet that publishes Web pages. A Web site typically has a computer serving Web pages, whereas a Web presentation is the actual Web pages themselves. For example, `www.lsu.edu` is the name of a Web site, whereas

  `www.lsu.edu/~holmes/index.html`

  is the name of a Web presentation.
- **Web server** A computer that satisfies user requests for Web pages.
- **Mirror site** A site that contains a duplicate copy of a Web presentation from another site. If a Web presentation is extremely popular, other sites may be used to mirror the original presentation; that is, they contain the same information as the original site. This allows the load on the Web server and the network to be distributed. If one server is down, a mirror site can be tried. If several mirror sites exist, it is a good idea to try the one closest to you first.

2.3.2 Uniform Resource Locator (URL)

In Section 2.2, we mentioned that the address of the Web page being displayed is shown under the toolbar in the location area of the browser window. This Web page address is a URL (pronounced "you-are-ell" or sometimes "earl"). Typing a URL in the location area and hitting the return key will cause the browser to attempt to retrieve that page. If the browser is successful in finding the page, the browser will display it. This high-level explanation does not, however, convey any of the details of what is happening. To go from a URL to having the Web page displayed, the browser needs to be able to answer such questions as:

Useful Item

1. How can the page be accessed?
2. Where can the page be found?
3. What is the file name corresponding to the page?

The URL is designed to incorporate enough information to answer these questions. Quite naturally, then, the URL has three parts. We can view the format of a URL as follows:

```
how://where/what
```

At this point, it is helpful to consider a sample URL to illustrate the three parts:

```
http://pubpages.uminn.edu/index.html
```

Let us break this example down into its components.

1. `http`—Defines the *protocol* or *scheme* by which to access the page. In this case, the protocol is *HyperText Transfer Protocol*. This protocol is the set of rules by which an HTML document is transferred over the Web (see further comments about `index.html`).
2. `pubpages.uminn.edu`—Identifies the domain name of the computer where the page resides. The computer is a Web server capable of satisfying page requests. Just as a waiter serves food, a Web server "serves" Web pages. The name `pubpages.uminn.edu` tells the browser on which computer to find the Web page. In this case, the computer is located at the University of Minnesota.
3. `index.html`—Provides the local name (usually a file name) uniquely identifying the specific page. If no name is specified, the Web server where the page is located may supply a default file. On many systems, the default file is named `index.html` or `index.htm`.

This example demonstrates that the URL consists of a protocol, a Web server's domain name, and a file name.

Like a social security number (SSN), which uniquely identifies a person, URLs uniquely identify Web pages. An SSN is an identifier; it indicates where someone lived regionally when their SSN was issued, and the year of issue. For example, 001 through 003 are for New Hampshire. The middle two digits are an indirect code for the year of issue. However, based on a person's social security number you cannot tell whether they currently live in Alaska or Rhode Island; you also cannot tell what type

TABLE 2.1

Protocols that May Occur in URLs.

Protocol Name	Use	Example
`ftp`	File transfer	`ftp://ftp.bio.umaine.edu`
`gopher`	Gopher	`gopher://gopher.tc.umn.edu/11/Libraries`
`http`	Hypertext	`http://www.chem.uab.edu/~pauling/argon.html`
`https`	Hypertext secure	`https://www.bankvault.com/`
`mailto`	Sending email	`mailto:kim-lee@mycompany.com`
`news`	Requesting news	`news:soc.penpals`
`telnet`	Remote login	`telnet://www.amnesty.org/`

of job they have. In comparison, the URL provides all the information a browser needs to locate and access a Web page anywhere in the world. The URL format is somewhat flexible so that the system can be adapted when necessary.

Entering a URL in the Location field of the browser will bring up the designated Web page, barring any problems. For example, if the Web page has moved to another machine or has been removed, or if you type an invalid URL, or if the server you are trying to access is unavailable, an error message will be displayed. Another way to retrieve a Web page is to mouse over and click on a hyperlink in the Web page that is currently being displayed. Recall that a hyperlink is a string of text or a graphic that points to other pages.

In the URL example presented earlier, the protocol to access the page was `http`. This is used for transferring an HTML document. Much of the power of browsers is that they are *multiprotocol*. That is, they can retrieve and render information from a variety of servers and sources. Table 2.1 provides a summary of other common protocols.

Useful Item

EXERCISES 2.3

Coast-to-Coast Surfing

8. Surf the Web and locate three Web pages that contain glossaries of computer jargon. List three terms you were previously unaware of and their definitions.

9. Compare and contrast an email address and a URL.
10. Give legal URLs for seven different top-level domains. Give legal URLs for three backpacking-related Web sites.
11. When a Web page is requested, a number of different error messages are possible. List their numerical codes and describe what each one means.
12. Can you locate any information about the `file` protocol? Describe your findings.

2.4 HyperText Markup Language: Introduction

Here we describe some basic HTML *tags* to get you started publishing a Web page, and we introduce the most useful *attributes* of each tag. Since HTML is not completely standardized yet, and since there is room for differences in interpreting and implementing a standard, it is possible that not all versions of all browsers will support all attributes.

A Web page is created when an ordinary ASCII text file is "marked up" using HTML tags and is then displayed using a browser. The tags are predefined combinations of characters enclosed between < and > characters. These symbols are called "less than" and "greater than," respectively. Sample tags are `<HTML>`, `<CODE>`, and `<TITLE>`. The tags are embedded within the text of a file, and they indicate how the text is to be interpreted and displayed by the browser. The word "markup" is used because copy editors use similar notations for editing printed matter.

How a Web page looks when displayed depends on (at least) three things:

Useful Item

1. The HTML tags used.
2. The specific browser rendering the page.
3. The user's system and monitor.

HTML tags do not define exactly how the Web page is supposed to look; rather, the tags describe how the elements of the page, such as headings, lists, paragraphs, and so on, are to be used. For example, many people think of a heading as being numbered, appearing in boldface, and being displayed in a larger font. The Web browser actually formats the HTML document, and different browsers may display the (same) HTML code differently. This is a point worth repeating. What you see using your browser may be

Useful Item

different than what someone else sees when viewing the same Web page. Since not all monitors support the same set of colors, the quality of the user's monitor affects the appearance of the colors in a document.[3] That is, your cranberry color may be very different than somebody else's.

HTML is not case sensitive. That is, the tag `<HTML>` means the same as `<html>`, which means the same as `<Html>`. However, you should be consistent with your tags, since it will make them easier to locate when you are editing files or debugging your code. Some Web authors prefer to use all capitals or a particular color for their tags, as these make the tags stand out from the remainder of the document. We follow both of these conventions throughout the book.

2.4.1 HTML Tag Syntax

Fortunately, learning HTML tag syntax is easy. The basic form for all HTML tags can be written abstractly as

```
<TAG ATTRI1 = "V1" ATTRI2 = "V2">item to be formatted</TAG>
```

where `ATTRI` means *attribute*. `TAG` means any HTML tag.

Many HTML tags have attributes. In the general form presented here, we have listed two attributes, called `ATTRI1` and `ATTRI2`. The number of attributes varies from tag to tag. Attributes typically have a choice of several values. In the expression we gave, the values are denoted `V1` and `V2`, respectively. Note the equals (=) sign; this is programming syntax for assigning `ATTRI1` the value `V1`. Also notice that we put quotes around `V1` and `V2`. For all HTML attributes, it is safe to quote their values. However, if the value is a number, it is sometimes not necessary to quote it. Nevertheless, we prefer to quote all values. Also observe that we have not left any white space between the item to be formatted and the surrounding tags. This is a good habit to get into, as otherwise the hyperlinks you format will not appear as you might want.

`TAG` has a corresponding *ending tag* (also referred to as a *closing tag*), namely `</TAG>`. The ending tag is the same as the starting tag except for the "/" character. Not every tag has an ending tag, but most do. Also, some ending tags are commonly omitted. We will mention these where appropriate. Ending tags can always be identified by the forward slash preceding

3 Actually, it is more than just the monitor. Even if you come down to the basic 256 colors, Windows PCs and Macs, for example, use inherently different color palettes, with only 216 colors in common. Also, the brightness setting on the two kinds of systems is different.

the tag name. If you keep these basic rules of syntax in mind while learning new HTML tags, you will have an easier time coding properly.

Many new users of HTML begin learning to program by cutting and pasting code from existing Web pages. Although it is often helpful to look at someone else's HTML code, we recommend against copying it, for the following reasons:

Hot Topic

1. Copyright issues must often be considered.
2. A great deal of HTML code is poorly written.
3. A lot of HTML code contains bugs.
4. It is easy to fool yourself into believing you have learned the basic elements of HTML.
5. A significant fraction of HTML documents have inconsistent styles within them, since they have been cut and pasted together many times over.

Occasionally, you may find that someone else did something so well you want to "borrow" it. In such cases, it is a good idea to ask the person for permission to use the code or to credit the person's work, as appropriate.

2.4.2 HTML Document Creation

To produce an HTML document, you need to use a text editor (Appendix B contains a discussion on text editing and file creation) to create an ASCII file with an extension of `.html` or `.htm`.[4] Remember the MIME type file extensions for HTML given in Table 1.3? The file you produce must contain correct HTML code so the browser can render it. Once you have constructed and saved the file, you need to set the file *permissions* accordingly so that other people on the Web can access the document. In Section 2.5, we describe the basics of installing a Web page, and we address such issues as file protections. Here we focus on creating a simple HTML document.

Every HTML document has two parts: a *head* and a *body*.[5] The associated HTML tags for these parts are `<HEAD>` with closing tag `</HEAD>`, and `<BODY>` with closing tag `</BODY>`. Surrounding all the text in the entire file are the beginning and ending HTML tags—`<HTML>` and `</HTML>`.

4 You could create the file using an HTML editor, which we discuss in Section 9.6. In this text we focus on learning to use HTML tags and attributes as a way of introducing the reader to computer programming.

5 There are a few exceptions, such as frameset pages.

These tags let the browser know that the file is indeed an HTML file. If the browser tried to render a Visual Basic program or a Word file as an HTML document, there would be lots of problems. Thus, the opening `<HTML>` tag and the closing `</HTML>` tag give the browser the go-ahead to render the file as an HTML document.

A title tag, `<TITLE>`, is contained within the head of the document to provide a title for the document. Its corresponding ending tag is `</TITLE>`. Do not include any HTML formatting within the title tag. The title should provide a concise description of the page, since the title is prominently displayed in the browser window's Title bar when the page is being viewed. Perhaps even more important is the fact that the title is also used as the default *bookmark description* when a Web page is bookmarked. A title such as "My Homepage" is a poor choice, whereas "Sung Lee's Homepage" is much more descriptive. Finally, an HTML document's title can affect how the document is indexed by some *search engines*.

Hot Topic

Useful Item

To demonstrate the basic elements of an HTML document, we will create a simple one and place it in a file called `cone.html`, standing for "creation one." In our example, "In-line/On-line: Creation Number One" is an appropriate title. As described so far, we have the following HTML code in `cone.html`:

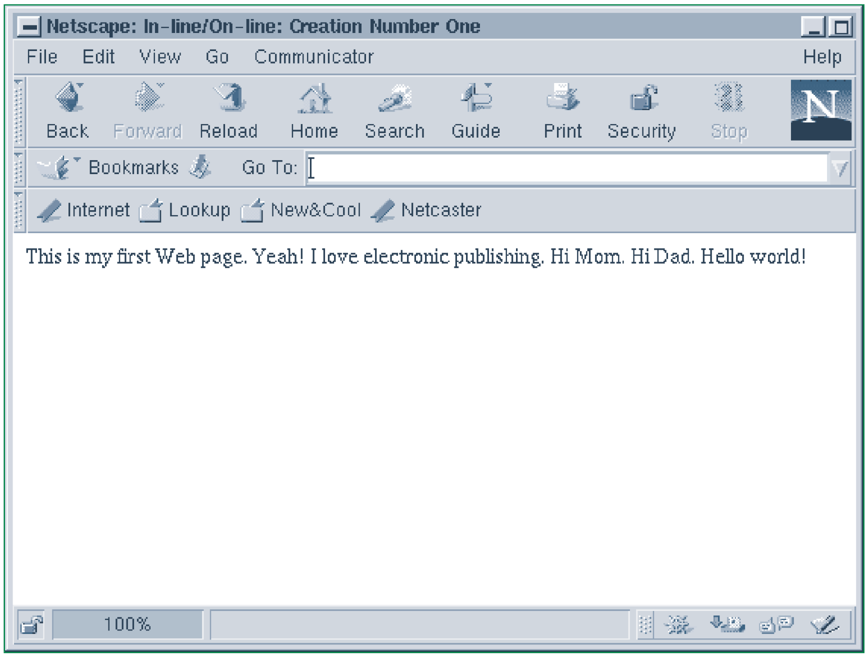
```
<HTML>

<HEAD>
<TITLE>In-line/On-line: Creation Number One</TITLE>
</HEAD>

<BODY>
</BODY>
</HTML>
```

Notice that the spacing (i.e., putting each item on a separate line) makes the code easy to read. Compare this with the following:

```
<HTML> <HEAD> <TITLE>In-line/On-line: Creation Number
   One</TITLE> </HEAD> <BODY> </BODY> </HTML>
```

We suggest you take the time to make your code readable. In the long run, you will save time editing and debugging. Some authors also use indentation to *pretty print* their code. For example:

```
<HTML>
   <HEAD>
      <TITLE>In-line/On-line: Creation Number One</TITLE>
   </HEAD>
   <BODY>
   </BODY>
</HTML>
```


Netscape: In-line/On-line: Creation Number One
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Go To:
Internet Lookup New&Cool Netcaster
100%

This is my first Web page. Yeah! I love electronic publishing. Hi Mom. Hi Dad. Hello world!

FIGURE 2.2 The file `cone.html` as rendered by a browser.

Looking at `cone.html` using your browser will not be very exciting at this point since the document area of the browser window has nothing to display. The only thing you will see is the title, "In-line/On-line: Creation Number One," in the Title bar. To display some words in the document area, add text between the `<BODY>` and `</BODY>` tags, as follows:

```
<BODY>
This is my first Web page.  Yeah!  I love
electronic publishing.  Hi Mom.  Hi Dad.  Hello world!
</BODY>
```

In Figure 2.2, we show the results of viewing `cone.html` with a browser. Notice the text in the document area of the browser window. Closely observe how the text is positioned in the browser window here and in succeeding examples. Browsers incorporate a simple *line breaking algorithm* to wrap the text if it would be too wide for the window's document area. If a window of another width is used, the text will wrap differently.

In the next section we cover how to actually set up Web pages so you can make your Web material available to everyone on the Internet.

EXERCISES 2.4

HyperText Markup Language: Introduction

13. Create a file called `index.html` having the basic set of tags that any HTML document should contain.

14. A field hockey player's coach asks her to design a Web page for the university's team. What are two possible titles for it? Create the code for a simple version of the page using one of your titles.

15. Create an HTML file that contains all the basic tags you have learned so far. Include a title for the page that would be a sensible title for your *personal page*. A personal page, also called a *homepage* by some people, is an individual's top-level HTML document. A personal page typically contains data about the individual, contact information, a table of contents to the Web presentation, and so on.

Useful Item

16. Write two paragraphs explaining to a friend, who has a limited knowledge of computers, how to prepare a basic HTML document.

17. Does your browser hyphenate words when it wraps the end of lines? Is hyphenation a choice that you can toggle on or off?

18. Create a Web page that lists the lengths of the Appalachian Trail, Continental Divide Trail, and the Pacific Crest Trail.

2.5 Web Page Installation

In order to view your pages on the Web, you will need to install them on a *Web server*. A Web server is a program, located on a computer with Internet access, that responds to a browser's request for a URL. That is, a Web server meets the demands of users by supplying or serving them the Web pages requested. Ideally, the server should have an uninterrupted Internet connection, so that the pages it handles are always available.

A Web server is accessible to many of us through work or school, and many ISPs include space on their computer that runs the Web server as part of the basic set of services covered in their monthly fee. The systems administrator responsible for the server can usually fill you in about the site-specific details for publishing your Web pages. In this section, we describe, at a high level, the basic steps necessary for installing your Web presentation. We then take a more detailed look at how Web pages can be set up on a UNIX-based Web server. UNIX is a type of computer operating system. Appendix D contains an introduction to UNIX. Other Web servers, such as those that are Windows based, will require a different installation procedure for Web pages.

Why are we discussing the UNIX platform?

- The basic principles we describe for UNIX can be applied to several other systems.
- The first Web servers developed were built on the UNIX platform.
- A huge number of sites are currently running UNIX-based Web servers. In July 1997, there were about 550,000 UNIX Web servers running *Apache* (the most popular Web server on the Internet).
- UNIX is prevalent in academic settings.
- Many new Web server features appear first on UNIX-based servers.
- The source code is often free for programs developed under UNIX, or at least some version of the software is often available for public use.

We should note that desktop operating systems are increasingly shipped with a simple Web server that can optionally be set up and used. This will eliminate one more distinction between server and client. The last big remaining distinction will be accessibility, that is, whether the server is normally on-line.

2.5.1 Basic Principles

What items are necessary for someone in, say, another country to view your Web pages? Here are a few of the requirements.

1. You need to have Web pages to publish.
2. A Web server where the files can be placed must be available to you, and you need to learn the steps to put the files in place, either to create them in place or (more often) copy them into place after you develop and test them.
3. The permissions on the files need to be set so that any user anywhere can read them. Such file permissions are often referred to as *world readable*.
4. When someone requests your Web page, the server has to deliver it.

The details of exactly how these steps are performed vary from platform to platform. Under normal circumstances, you will only have to go through this entire setup process once. Thus, even though the procedure may be a bit technical, it is worth performing, as the rewards are great.

2.5.2 A Specific Example

In a UNIX environment, setting up a Web page usually involves creating a special directory in your home directory that contains all of your files to be published on the Web. This directory may contain subdirectories as well. Usually, the name of this special directory is fixed. Your systems administrator can tell you what the name of the directory should be on your system.

Suppose the name of the directory is `public_html`. You will need to use the `mkdir` UNIX command, which stands for "make directory," from within your home directory to create the `public_html` directory.

Since this directory must be accessible by others in order to permit them to read your Web pages, you will need to change the permissions on `public_html` to be world readable and world executable. You will also have to change permissions on your home directory so that it is world readable and world executable. This will allow others to access your `public_html` directory.

Be careful not to give anyone extra permissions on your home directory; do not make it world writable. Additionally, you should set permissions on private files sitting in your home directory so that only you can read them. Once the `public_html` directory is in place and the permissions on it and your home directory are set correctly, all your HTML files should be located there, or in subdirectories of the `public_html` directory.

When a Web page request is received for a URL that ends in a directory name, the Web server usually tries to load a default file. Again, the name of this default file can be determined by asking your systems administrator. Many installations use a file called `index.html`, which becomes your top-level Web page. Of course, you need to make this file world readable. If you use the default file name, users who know only your account name can access your Web page. How is this possible? Suppose you know that Sarah Conners' account name is `sarahc` and that her Web pages are served from a UNIX-based Web server called `pubpages.yikes.gov`. To access her pages, you could try the following URL:[6]

```
http://pubpages.yikes.gov/~sarahc
```

6 The ~ and an account name combined are used to refer to the account owner's home directory. For example, `~sarahc` refers to Sarah Conners' home directory.

With a server default top-level page name of `index.html`, this is an abbreviation for the following URL:

```
http://pubpages.yikes.gov/~sarahc/index.html
```

In other words, the server automatically looks in the `public_html` directory and appends the default file name to a URL that does not contain a file name. Notice that the directory `public_html` is not included in the URL for Sarah's personal page even though `public_html` is part of the path for the file `index.html`. By convention the server always completes this portion of the URL and so it does not need to be specified by the user.

Hot Topic

As you create new directories and Web pages, you will need to set the permissions on them so that others can access the directories and files. If you are creating a Web presentation with many HTML files (those with `.html` or `.htm` extensions) and graphics files (for example, those files ending in `.gif` or `.jpeg` extensions), you may be wise to organize your files into subdirectories of the `public_html` directory. For starters, you may want to create `HTML`, `GIF`, and `JPEG` subdirectories.

Summary: UNIX Web Page Setup

We will describe a typical scenario you might go through in setting up your Web page on a UNIX-based Web server. Suppose the directory where Web pages are placed is called `public_html` and the default file the server returns is called `index.html`. The following steps illustrate how to install this page. These steps assume you have read Appendix D or that you are familiar with the UNIX operating system. The commands must be executed in the order shown.[7] We preface each command with the UNIX prompt `%` and then give a brief explanation of what the command accomplishes.

1. `%cd`—Change to your home directory.
2. `%chmod og+x ~`—Set the permissions on your home directory to be world executable.
3. `%mkdir public_html`—Create the directory `public_html`.
4. `%chmod og+x public_html`—Set the permissions on the directory `public_html` to be world executable.
5. `%cd public_html`—Change directories from the current directory to the `public_html` subdirectory.

7 Technically, different sequences would work, but some steps depend on others.

6. `%edit index.html`—Here, `edit` stands for your favorite text editor. The idea is to create a file called `index.html` and include the appropriate HTML code in it.

7. `%chmod og+r index.html`—Set the permissions on `index.html` to be world readable.

Go On-Line

Once this sequence of steps, or a similar one depending on your local site, has been carried out, your `index.html` file should be ready to be viewed on the Web. Enter the URL for your page in the location area of the browser. For example, if your Web server is `www.chem.unlv.edu` and your account name is `shannon`, your URL would be something like

```
www.chem.unlv.edu/~shannon/index.html
```

If the page loads, congratulations, as you have just published your first Web page. If your page does not load, you may get an error indicating that the file protections are not set properly. Review the steps again and use the command `ls -l` to check that the protections on the necessary directories and the file `index.html` are set properly. Remember, as you install new files and subdirectories, you will need to set the permissions on these as you did on `index.html` and `public_html`, respectively.

EXERCISES 2.5

Web Page Installation

19. Create the necessary directories and files to install your Web page. For starting out, create a simple HTML document with a title of your name. Check to see that your page is accessible on the Web. Document any problems you have during the installation process.

20. What is the URL of your Web page? Does your server supply a default file name? If so, what is it? Can you use an abbreviated URL to access your Web page? If so, what is it? (When providing someone else with the URL of your personal page, it is often best to give them the shortest possible URL and thus minimize the chances for typing mistakes.)

Hot Topic

21. Create new folders or directories to hold GIF and JPEG images. Set the protections on them so that images stored within them will be accessible.

2.6 Web Page Setup

Earlier you saw that each HTML document contains a head and a body. You will learn about the `<HEAD>` and `<BODY>` tags in detail here. In addition, we will examine colors, the `<FONT>` tag, the inclusion of hidden comments in an HTML document, and the methods for producing interesting backgrounds. Using the techniques described in the last section, you should be able to implement and test all of the HTML features discussed in this section. Keep in mind that many of our examples are HTML code fragments and not complete documents.

2.6.1 Head Tag

The head tag, `<HEAD>`, has no attributes. However, several tags can be included inside it. The most important of these is the title tag described earlier. A couple of others that you may find useful are described in the following paragraphs.

Basefont Tag

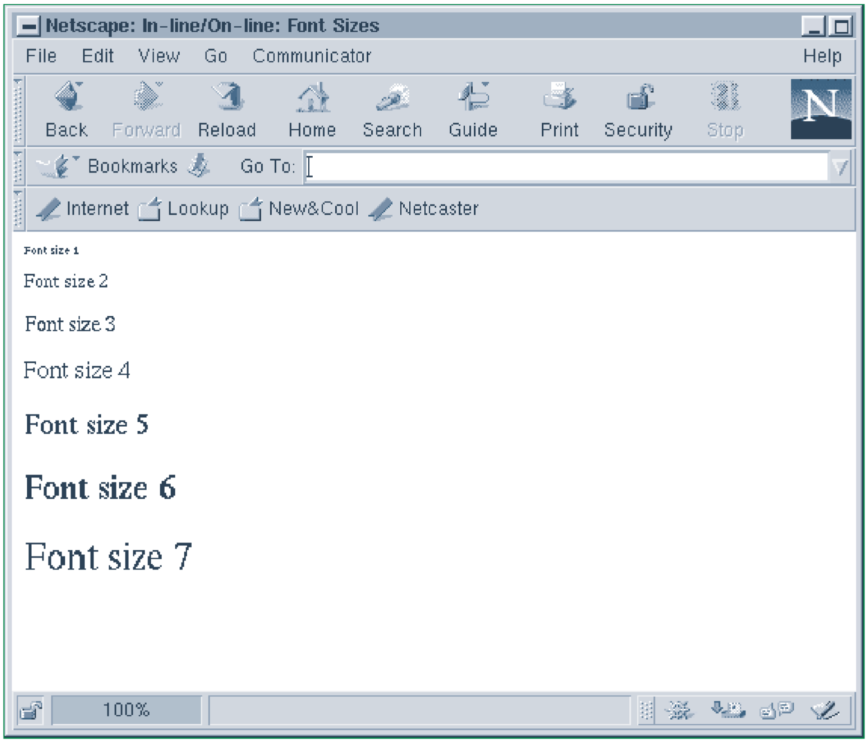
The basefont tag, `<BASEFONT>`, defines the font size to be used in the HTML document and may be included in the head of the document. It is also possible to use the basefont tag in other locations of an HTML document. Most browsers permit a range of font sizes. Seven different sizes are commonly available, with the sizes ranging from 1, which is the smallest, to 7, which is the largest. Figure 2.3 displays the seven different text sizes. Each browser renders an HTML document using a default font size, which is usually 3. To set the font size slightly larger for the overall document, you can use the `SIZE` attribute of the basefont tag, as follows:

```
<HEAD>
<BASEFONT SIZE = "4">
</HEAD>
```

You can also use a setting stated as

```
<BASEFONT SIZE = "+1">
```

Notice the plus sign. This sets the font one size larger than the default. In typing this tag, it is common practice to omit the double quotes. The ending tag `</BASEFONT>` returns the font size to its default value. Note: when `<BASEFONT>` is used in the head of a document, the ending tag is usually omitted.


Netscape: In-line/On-line: Font Sizes
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Go To:
Internet Lookup New&Cool Netcaster
100%

Font size 1

Font size 2

Font size 3

Font size 4

Font size 5

Font size 6

Font size 7

FIGURE 2.3 The seven different text sizes that browsers typically support. A value of 1 indicates the smallest size and a value of 7, the largest size. The default is usually set to 3.

Base Tag

The base tag, `<BASE>`, is useful for setting some *global parameters* of an HTML document and may be included in the head of the document. A global parameter is an attribute that has an effect on the entire document. Before explaining how the base tag is used, you need to understand *absolute* versus *relative* URLs. An absolute URL is complete in that it contains all the components of a URL: the how, where, and what. For example, the following:

```
http://www.hospital.arizona.com/library/books/main.html
```

is an absolute URL. The how is

```
http
```

the where is

```
www.hospital.arizona.com
```

and the what is

```
library/books/main.html
```

By comparison, a relative URL, as the name implies, relates to some base URL and may be used in many different places in an HTML document. The default base URL of an HTML document is the URL of the document itself. However, the base URL can be changed using the `<BASE>` tag and its attribute `HREF`. For example, suppose we have the following code:[8]

```
<HEAD>
<TITLE>Water Sports to Die For</TITLE>
<BASE HREF =
  "http://www.fishing.com/BOATS/outboard.html">
</HEAD>
```

in the HTML document whose URL is

```
http://www.paloalto.gov/entertainment/water.html
```

Then all references to URLs in the file `water.html` would be relative to the base URL,

```
http://www.fishing.com/BOATS/outboard.html
```

instead of the default URL,

```
http://www.paloalto.gov/entertainment/water.html
```

More concretely, suppose a hyperlink in the file `water.html` referenced the URL

```
http://www.fishing.com/BOATS/inboard.html
```

Having set the base URL to

```
http://www.fishing.com/BOATS/outboard.html
```

8 Notice that we have split the `HREF` over two lines. This is not significant. We have only done this so the reference did not extend into the margin of the book. In your files, you may include the expression on a single line. To minimize ambiguity, we normally break a line at a delimiter, such as a comma or an equals sign, or between attributes.

the referenced URL could be specified simply as `inboard.html`, since the prefix of the URL can be determined from the base URL. Why is this useful? If, for example, there are many references to URLs in the document collection found on the `www.fishing.com` server, their names can all be shortened.

Hot Topic

Web addresses change frequently, and as explained here, the base tag can be used to simplify the updating of hyperlinks inside a file. If absolute URLs are hard-coded into HTML documents, then if a collection of documents moves to a new server, it may be necessary to edit all the files in the collection to update the URLs. This can result in a tremendous amount of editing. However, if relative URLs are used, it would probably only be necessary to update the base tag's `HREF` at the beginning of each document. We recommend using relative URLs where possible. An example should make this point clear.

Suppose a student at Winthrop University has the following absolute URL for her personal Web page:

```
http://www.winthrop.edu/~JenniferJones/index.html
```

Figure 2.4 illustrates Jennifer's file structure, viewed as a tree. In her account, she has two top-level directories (or folders) called `private` and `public_html`. In her `public_html` directory, which contains her WWW material, she has four subdirectories and a file called `index.html`. The four subdirectories of `public_html` are `books`, `family`, `gif`, and `jpg`. The directories `gif` and `jpg` contain some of her graphics. The absolute URL for her Web page about her mother is

```
http://www.winthrop.edu/~JenniferJones/family/mom.html
```

Suppose Jennifer's brother Jeff, who attends Tennessee Technical University, wants to include a picture of South Carolina in his HTML document called `sister.html`. Jeff could do this using an absolute URL of

```
http://www.winthrop.edu/~JenniferJones/jpg/sc.jpg
```

He would have to type equally cumbersome URLs to include the picture of Rock Hill and also the picture of Jenny's favorite lake. In the interest of future portability, Jeff decides instead to use a base tag by including the

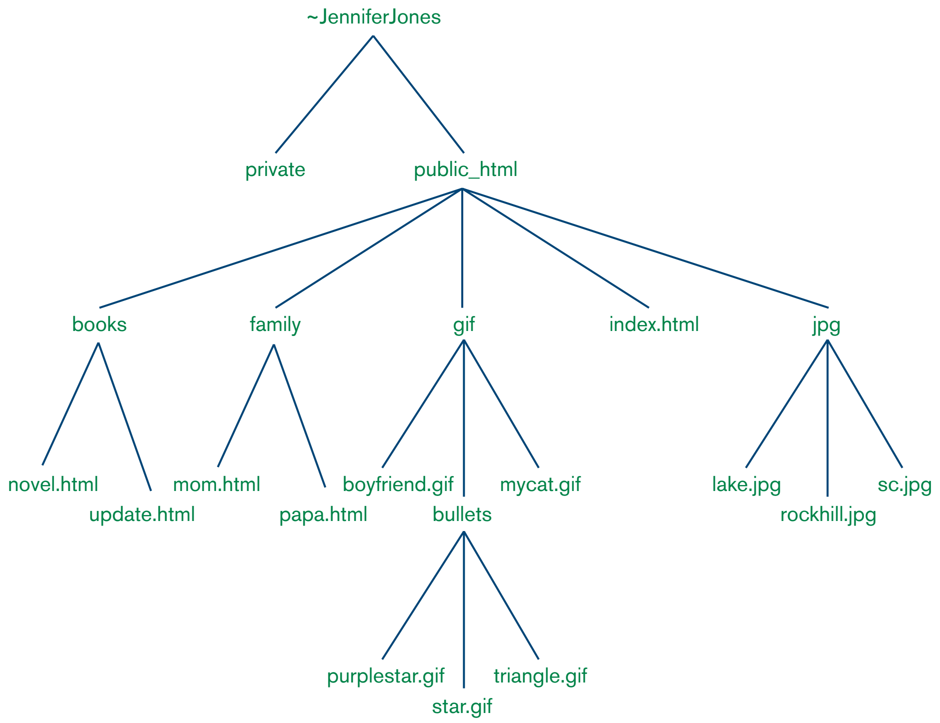


FIGURE 2.4
File structure of Jennifer Jones' computer account.

following code in his file called `sister.html`.

```
<HEAD>
<TITLE>Jenny Jones: My Cool Sister</TITLE>
<BASE HREF = "http://www.winthrop.edu/~JenniferJones ↩
    /jpg/sc.jpg">
</HEAD>
```

(We use the symbol ↩ to denote that the line continues without any spaces.) The reference

```
http://www.winthrop.edu/~JenniferJones/jpg/sc.jpg
```

can now be replaced by just `sc.jpg`. Similarly, a reference to

```
http://www.winthrop.edu/~JenniferJones/jpg/lake.jpg
```

can be specified simply as `lake.jpg`. When Jenny graduates and moves her document collection to a new Web server, Jeff only has to update the

HREF in his base tag, as opposed to changing all references to Jenny's files individually.

Jenny can also refer to her own document collection using relative URLs. Suppose, for the sake of discussion, that www.winthrop.edu is a UNIX-based Web server. Also assume that Jenny has not set the base tag in her papa.html file. If Jenny wants to include a picture of Rock Hill in her papa.html file, she could use an absolute URL of

```
http://www.winthrop.edu/~JenniferJones/jpg/rockhill.jpg
```

or she could use a relative URL of

```
../jpg/rockhill.jpg
```

(Note: the symbol .. is a way of moving up the directory structure one level.) This URL is relative to the default URL for the file papa.html, which is

```
http://www.winthrop.edu/~JenniferJones/family/papa.html
```

In other words, the default URL for a file is just the URL of the file itself.

The expression ../jpg/rockhill.jpg can be best understood if partitioned in the following way:

- ".." says go up one directory, which places us in the directory /public_html.
- /jpg says go into the directory /jpg.
- rockhill.jpg tells us the file name to include.

An expression such as

```
../gif/bullets/triangle.gif
```

in the file update.html could be used to refer to the triangle icon called triangle.gif. To refer to the purple star bullet from within the file index.html, Jenny could use the expression

```
gif/bullets/purplestar.gif
```

We will have more to say about the base tag, and in particular its TARGET attribute, when we discuss the concept of frames.

If possible, try to use all lowercase or all uppercase in your own URLs, since this may help people avoid typing errors. Also, try not to use "underscore" (_) and "dash" (-), and try never to use both of these in the same URL (or email address). These symbols may be hard to distinguish on some monitors.

Meta Tag

The least well understood and second most widely used (and occasionally abused) tag inside the head tag is the meta tag, `<META>`. This tag is used to include additional information about a document and can be used to pass additional information to a browser. There is no ending tag for `<META>`, and a document can have multiple `<META>` tags.

The attributes of the meta tag are `NAME`, `CONTENT`, and `HTTP-EQUIV`. You should include a modest list of keywords, say three to five, as the value of the `CONTENT` attribute. If someone is searching for a particular topic, your page may be returned if one or more of your keywords match their search request. Do not abuse the meta tag by expanding the number of items in the `CONTENT` attribute to ridiculous lengths. People who abuse the meta tag in this fashion are known as *spamdexers*, and there are many classic cases of this behavior.

Useful Item

For example, someone who has a Web page about woodworking might include the following:

```
<HEAD>
<META NAME = "keywords"
      CONTENT = "woodworking, cabinetmaking,
                 handmade furniture">
</HEAD>
```

2.6.2 HTML and Colors

Colors can help or hinder readers of your Web pages. There are two ways of defining colors in HTML documents. One involves straightforward color names, such as blue, cranberry, green, orange, red, and yellow. Many browsers have a list of predefined color names. Such lists are usually easy to find on-line. However, since different browsers have different lists and since the definitions of individual colors may vary from browser to browser, we recommend using the color numbering scheme. Although somewhat more complex, this scheme is better supported across different platforms.

Go On-Line

A little computer numbering terminology is necessary first. A *bit* is either 0 or 1. The term bit stands for *bi*nary digi*t*. Bits are useful for counting in base two. Table 2.2 shows some binary numbers and their

TABLE 2.2

Sample Decimal Numbers and Their Corresponding Binary Values.

Decimal	Binary
0	0
1	1
2	10
4	100
5	101
29	11101
255	11111111

corresponding decimal values. In this text, we use the phrase “decimal number” to represent a base ten number from the set of natural numbers, 0, 1, 2, ... For example, 11101 in binary represents 29 in decimal, as follows:

$$(1 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$$

equals

$$16 + 8 + 4 + 0 + 1 = 29$$

(Remember that any natural number raised to the zero power equals one.)

This explanation of binary numbers is really just a useful warmup. Colors in HTML documents are represented as *hexadecimal* numbers, which are numbers in base sixteen. Hexadecimal can also be considered a shorthand for representing four bits, since only four bits are needed to represent the numbers 0 through 15. Since there are only 10 base ten digits (0–9), we need six additional symbols in the hexa decimal number system: A, B, C, D, E, and F. Table 2.3 provides the values of these and some other hexadecimal numbers, with their corresponding decimal values. For example, 752 in hexadecimal represents the number 1874 in decimal, as follows:

$$(7 \times 16^2) + (5 \times 16^1) + (2 \times 16^0) = 1792 + 80 + 2 = 1874$$

You will be most concerned with hexadecimal numbers having two digits, because colors in HTML documents are represented by three two-digit hexadecimal numbers. Each of the two digits signifies the amount of one of three primary colors. In other words, a color is formed by mixing different amounts of red, green, and blue. The first two digits represent the

TABLE 2.3

Sample Decimal Numbers and Their Corresponding Binary and Hexadecimal Values, with Each Hexadecimal Number Written as Four Binary Digits.

Decimal	Binary	Hexadecimal
0	0000	0
1	0001	1
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F
17	0001 0001	11
35	0010 0011	23
255	1111 1111	FF
1874	0111 0101 0010	752

red component, the next two the green portion, and the last two the amount of blue. This method of representing colors is called the *RGB color model*. We can view this as follows:

$$\underbrace{\text{digit1 digit2}}_{\text{red}}\ \underbrace{\text{digit3 digit4}}_{\text{green}}\ \underbrace{\text{digit5 digit6}}_{\text{blue}}$$

The first two digits, designated digit1 and digit2, represent the red component. For example, if digit1 = 0 and digit2 = 0, there is no red component. However, if digit1 = F and digit2 = F, the maximum possible red component is used. Since FF is the largest two-digit hexadecimal number (it represents 255 in decimal), this is the maximum red we can specify using two hexadecimal digits. The green component is specified by digit3 and digit4, whereas the blue portion is given by digit5 and digit6.

As an example, 000000 means 00 or no red, 00 of green, and 00 of blue. This total absence of color is the color black. So, 000000 represents black.

It is common practice to preface these six-digit combinations by a # sign to denote that they represent a color. You can imagine the possible ambiguity arising from a six-letter color name consisting only of the letters A–F. Did the user want a color name, or a hexadecimal number to be interpreted as a color?

TABLE 2.4

Some Colors and Their Corresponding Hexadecimal Representations.

Color	Hexadecimal Value	Color	Hexadecimal Value
black	`#000000`	orange	`#FFA500`
blue	`#0000FF`	plum	`#DDA0DD`
chocolate	`#D2691E`	purple	`#800080`
crimson	`#DC143C`	red	`#FF0000`
gold	`#FFD700`	salmon	`#FA8072`
green	`#00FF00`	silver	`#C0C0C0`
gray	`#808080`	violet	`#EE82EE`
maroon	`#800000`	white	`#FFFFFF`
navy	`#000080`	yellow	`#FFFF00`

The color #FFFFFF represents bright red, bright green, and bright blue. This complete mix of these three colors yields white. Table 2.4 lists several color names and their corresponding hexadecimal values.

Desktop window systems allow the opportunity to download and install a *freeware* or *shareware* utility with a "color picker" to determine the hexadecimal value of any color you click on.

2.6.3 Body Tag

The body is the second and main part of every HTML document. The text and HTML code that goes between the body beginning and ending tags is rendered and displayed in the document area of the browser's window. The body tag, `<BODY>`, has a number of useful attributes that let you set some global parameters. The most interesting attributes deal with the document's text color and background color, and the properties of hyperlinks.

Text Color

The `TEXT` attribute is used to change the default text color for an entire document. (We will see how to override this setting in the next section on fonts.) Suppose you have a document and would like to use a maroon-colored text. The following HTML code shows how:

```
<BODY TEXT = "#800000">
```

A common mistake for beginners is the use of a text color that clashes with the background color. This makes your document hard to read and less desirable to visit. Make sure to select a background that goes well with your choice of text color. If possible, view the color combination on several different platforms.

Background Color and Tilings

Hot Topic

Including effective colors (legible text and coordinated colors) in your Web pages can really improve appearance and navigability. Not only that, if you have a consistent color scheme running throughout your presentation, visitors will know they are still at your presentation as they select new hyperlinks. Conversely, if someone is surfing your Web pages and they click on a hyperlink taking them to a new page having a completely different color scheme, they will assume that they have left your pages. A carefully chosen color scheme can unite your pages and give them your own "look."

Two attributes to the body tag that let you add color to a Web page background are `BGCOLOR` and `BACKGROUND`. (The default is typically either a gray or white background.) The `BGCOLOR` attribute is used to set the background of an HTML document to a single color. For example, the following HTML code sets the document background area to blue:

```
<BODY BGCOLOR = "#0000FF">
```

You can also use color names:

```
<BODY BGCOLOR = "blue">
```

As we explained earlier, we recommend you stick with hexadecimal numbers, since there are many shades of blue, and the one you see on your screen may not match the one another viewer might see on their system. Choosing a good background color is not easy and may require some experimentation. Just because your favorite color is aqua does not mean that it is the best choice for your Web pages. Select a color that is easy on the eyes and makes the text easy to read. White is usually safe. Black is difficult to use effectively. If you use a black background with white text, many users will have a problem printing your Web page. The black background color will not be printed, and the white text will not show up on white paper.

Hot Topic

Useful Item

As for the `BACKGROUND` attribute, imagine holding up a postage-stamp-sized tile in front of your face. When you see an interesting tiled pattern on a Web page, it is usually created by taking postage-stamp-sized images and repeating them as necessary to fill in the document area in the browser's window. If you widen the document window, the pattern "expands" to fill it. Shrink the window and the pattern "contracts" to fill it.

The concept we are describing is called *tiling*. You may take any image and include it in your HTML document so that it tiles the background. The tiling is performed by the browser using a *tiling algorithm*. Abstractly, the tiler starts in one corner of the screen and horizontally lays tiles, which are copies of the image, until the right edge of the browser window is reached. If necessary, a tile is "cut." The tiler then moves down to the next horizontal row. The entire process is repeated, with the possibility that all tiles in the last row need to be cut.[9] Since computer scientists have figured out very efficient tiling algorithms, you do not notice any delay caused by the tiling procedure.

When using an image to tile a background, choose something that will not interfere with the legibility of your text. Common choices involve patterns of paper, clouds, and water. Marble textures are also popular. If you have an image called `marble.jpg`, the following HTML code would include it as a tiled background for you:

```
<BODY BACKGROUND = "marble.jpg">
```

As we said, be careful to choose a background pattern that makes your text easy to read. For example, a complex psychedelic image may be awesome, but it may not be suitable as a background. If no text color is clearly readable against that background, do not use it.

Some Web authors like to create *splash screen* effects by first loading in a color and then tiling a pattern over it. The following HTML code first loads in a color and then a background pattern:

```
<BODY BGCOLOR = "#008888" BACKGROUND = "dotblue.jpg">
```

Notice that when two attributes are used within the same tag, by convention a single blank space is used between the value of the first attribute and the name of the second attribute.

One reason for including the `BGCOLOR` attribute, even if you plan to use the `BACKGROUND` attribute, is that if someone has the automatic display of images turned off in their browser, the background image will not tile, but they will still get the background color. As a routine, the `BGCOLOR` attribute should be specified before the `BACKGROUND` attribute.[10]

Hyperlink Colors

Three attributes are used for changing the color of a hyperlink, where the color depends on the current *state of the hyperlink*. The three possible

9 There are many presentations on the Web where good background graphics are available for free. This book's on-line Web presentation provides the URLs of several.

10 In theory, the order should not matter, but it seems to on some systems.

states are: unvisited, visited, and currently thinking of visiting. These are defined as follows:

LINK Unvisited hyperlinks. The color value assigned to LINK sets the color for all unvisited hyperlinks in the HTML document.

VLINK Visited hyperlinks. The color value assigned to VLINK sets the color for all visited hyperlinks, that is, hyperlinks the user has already explored.[11]

ALINK A hyperlink the user is thinking of visiting. The *A* stands for *active hyperlink*. The color value assigned to ALINK sets the color of a hyperlink that the user has moused over and depressed the mouse button on. (This option is not supported by all browsers.)

Most browsers provide default colors for the three types of hyperlinks, and text-based browsers usually use underlining or reverse video to make hyperlinks more prominent on screen. However, many HTML documents specify all three attributes in the body tag. We encourage you to think carefully before you select colors that change the browser defaults people are used to seeing. Also, try to select colors that go well with your background and document text. Make sure your hyperlinks stand out; that is, do not set both the background color and LINK to red.

Suppose you have an HTML document that has a white background. The following code specifies unvisited links as red, visited links as gray, and the active link as yellow:

```
<BODY BGCOLOR = "#FFFFFF"
      LINK = "#FF0000"
      VLINK = "#808080"
      ALINK = "#FFFF00">
```

Body Attributes Combined

Conceptually, it is straightforward to use all of the body attributes in combination. To do this effectively often requires some trial and error. When you see particularly effective colors used on someone's Web page, look at the document source and note the colors used. To create a white-colored

11 Your browser keeps a history file of where you have been, in addition to the cache and bookmarks, so it can determine which hyperlinks are to be treated as VLINKs.

document area with red text, along with green unvisited hyperlinks, orange visited hyperlinks, and purple active hyperlinks, use the following HTML code:

```
<BODY BGCOLOR = "#FFFFFF"
      TEXT = "#FF0000"
      LINK = "#00FF00"
      VLINK = "#FFA500"
      ALINK = "#800080">
```

2.6.4 HTML Font Colors

In the previous section, we saw how to set the text color of an entire document to any (single) color. In this section, we examine the font tag, `<FONT>`, which allows us to change the color of any portion of text. Modifying the color of a segment of text is easy using the `COLOR` attribute of the font tag. For example,

```
<FONT COLOR = "#0000FF">
I am going swimming today
</FONT>
```

changes the text "I am going swimming today" to blue. The preceding and succeeding text is unaffected. Only the text between the font beginning and ending tags is changed. When altering the color of small segments of text, check to make sure that the text is still readable when rendered. In general, do not use a large number of text color changes in a single HTML document.

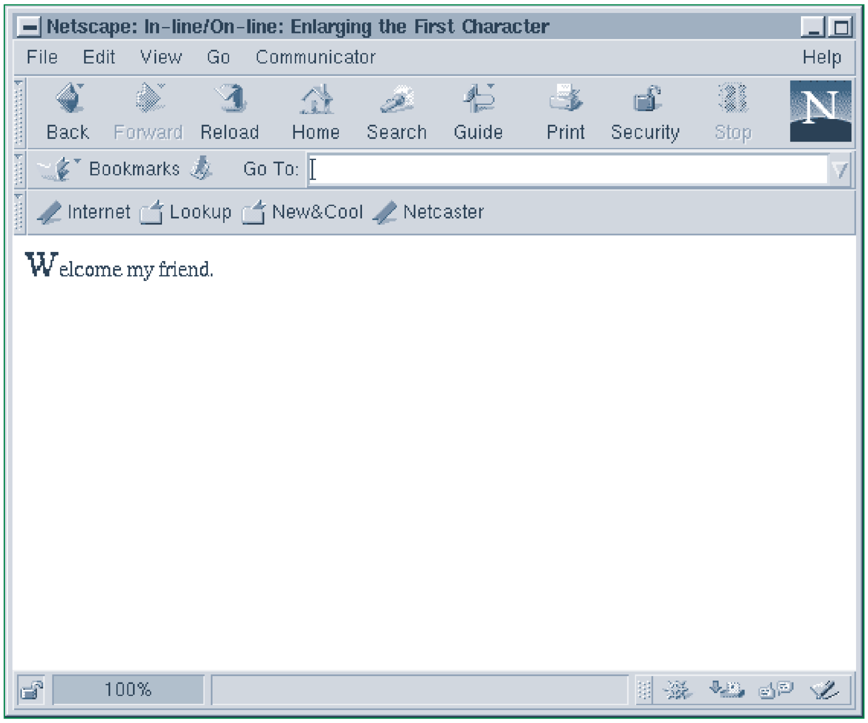
2.6.5 Font Size

As previously stated, the basefont tag is commonly used in the head of a document to alter the font size for the entire document. The `SIZE` attribute of the font tag is typically used to change the font size of an individual part of a document.

One option, for example, makes the first letter of the first paragraph slightly larger than the rest of the text. This can be accomplished with the following code:

```
<FONT SIZE = "+3">W</FONT>elcome my friend.
```

Figure 2.5 illustrates the effect of this sample code. The remarks pertaining to the `SIZE` attribute of the basefont tag are also relevant here. For example, the `SIZE` attribute can have an absolute value of between 1 and 7.


Netscape: In-line/On-line: Enlarging the First Character
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Go To:
Internet Lookup New&Cool Netcaster
100%

Welcome my friend.

FIGURE 2.5 Illustration of enlarging the first character in the beginning paragraph of an HTML document.

2.6.6 Font Face

Most browsers also support a `FACE` attribute for the `<FONT>` tag, allowing you to specify a particular font type. For example,

```
<FONT FACE = "avantgarde">New font type</FONT>
```

specifies that the phrase "New font type" should be displayed in the Avant-garde font. However, not all browsers support the same font families, and various fonts are rendered in different sizes. We therefore recommend using only a limited number of different font types. Otherwise, the spacing and font type you see may be completely different than what someone else sees when viewing your Web page.

2.6.7 HTML Comments

The comment feature provides you with a way to document your HTML files and make notes to yourself. Those who have written program code realize the importance of documenting the process concurrently with program construction. The same can be said for Web page design. Without such notes, it is unlikely you will be able to remember all the important details that went into a Web page's design.

The comment tag is not `<COMMENT>`; it is the set of symbols `<!--` for the beginning tag and `-->` for the ending tag. A comment, properly included, does not change the appearance of your Web page.

To use the comment tag, place the comment text between the pairs of dashes in the tag. For example,

```
<!-- This is a comment. -->
```

The browser will not interpret (i.e., display) the text between the pairs of dashes.

Do not include any embedded HTML code in commented text, since the results are unpredictable. Making a solid line of dashes may also cause unpredictable results in some browsers. Use periods or asterisks instead to get a similar effect. In the exercises, we ask you to experiment with how your browser handles HTML embedded within a comment declaration.

Here is an example of how the comment tag might be used.

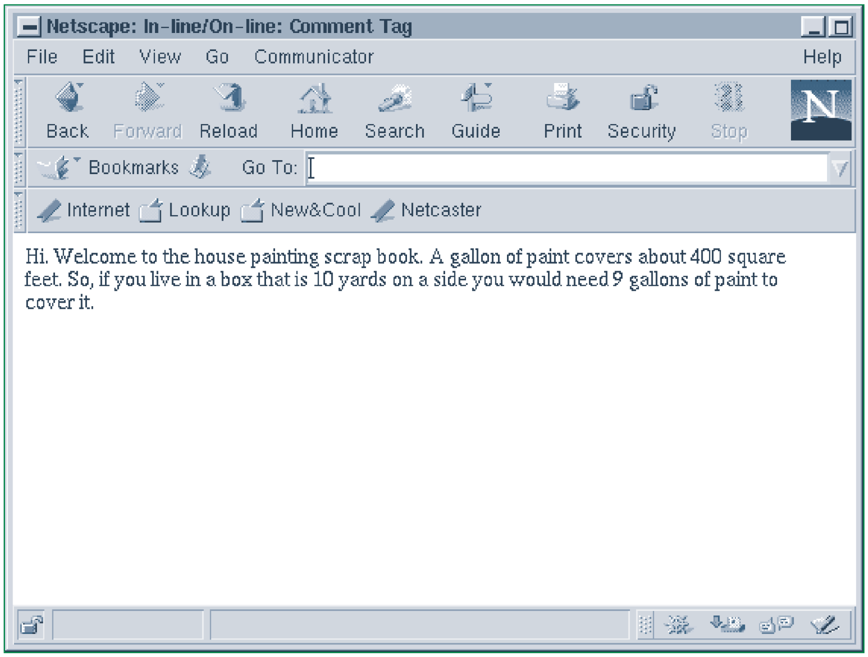
```
<!-- How to paint a house in seven easy steps. -->
<!-- Written in July of 1999.  Gretchen von Gelder -->
<!-- Most of the description is from my file Home-notes. -->

Hi.  Welcome to the house painting scrap book.  A gallon
of paint covers about 400 square feet.  So, if you live
in a box that is 10 yards on a side, you would need
<!-- (4 x (10 x 3) ∧ 2)/400 = 9 -->
9 gallons of paint to cover it.
```

(We have formatted the text inside the comment so that it looks good on the printed book page.) Including a comment like the last one helps you recall the actual calculation that you made. If some user sends you email in January of 2002 asking how you figured on 9 gallons of paint, you can simply consult your comment to review the math. The notes at the top of the document remind you what the HTML code is about, who wrote it, and when. It is generally a good idea to include this information in an HTML file, or any on-line file for that matter. The output resulting from this HTML code is shown in Figure 2.6.

You could also embed a copyright notice at the top of a document. For example,

```
<!-- copyright (c) 2002, Mark Jackson -->
```

With the basic principles and HTML tags described thus far, you should be able to create some interesting Web pages. Remember to comment them so you can easily recall what you did, why, and when.


Netscape: In-line/On-line: Comment Tag
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Go To:
Internet Lookup New&Cool Netcaster

Hi. Welcome to the house painting scrap book. A gallon of paint covers about 400 square feet. So, if you live in a box that is 10 yards on a side you would need 9 gallons of paint to cover it.

FIGURE 2.6 Display of a web page that includes comments. Note that the comments are not rendered by the browser.

EXERCISES 2.6

Web Page Setup

22. Provide the head specification for an HTML document in which the entire document's font size is set to a value two less than the default size.

23. Provide an absolute URL that Jennifer Jones could use to include her South Carolina image in her `novel.html` file. What is a relative URL she could use to do the same thing?

24. Document two widely published cases involving spamdexers. (Hint: Use a search engine such as Yahoo!)

25. What are the decimal equivalents of the following hexadecimal numbers: 16, 22, AD, CB, FG, 79, and FF?

26. What are the probable hexadecimal values for the following colors: black, cranberry, lime, orange, paleblue, royalblue, and white?

27. If you were designing a Web page about the mountains and wanted to create a "wilderness feel" for the page, what colors might you decide to use in your presentation? What are their corresponding hexadecimal values?

28. Andrew "Thumbs" Michaels is typing in a color specification, intending to use a color name, not a hexadecimal number. He "accidentally" hits the Shift key while bumping the 3 key, so the code starts out with a # sign. He then keys in his desired color name, making at most four typos. Are there any color names he could have botched that would result in a valid hexadecimal number? As an example, if he meant to key in "beige" but made three typos `i` $\mapsto$ `a`, `g` $\mapsto$ `a`, and accidentally appended an `a`, the result in capitals would be `BEAAEA`, a valid hexadecimal number.

29. Report the URLs for three different Web pages that have cross references providing hexadecimal values for color names.

30. Can you locate a Web page that allows you to enter a color name and then have it return the corresponding hexadecimal representation? How about vice versa?

31. Are color names case sensitive? How about letters in hexadecimal color representations? (Notice that we have always used lowercase letters for color names and capital letters in hexadecimal numbers.)

32. Write a body tag specification to produce a salmon-colored background. Do this once using the color name "salmon" and once using the hexadecimal representation for salmon.

33. Suppose you did not have (and could not find) a table to look up hexadecimal values for a given color. Describe how you could obtain an approximate value just using HTML code and your browser.

34. What colors do the following hexadecimal patterns probably represent: `#00FF00`, `#FF0000`, `#FFFF00`, `#FA5723`, and `#00BC51`?

35. Write a `<BODY>` tag specification to generate a background that is tiled using the pattern

    ```
    http://www.herewego.com/backgrounds/droplets.gif
    ```

    If you use an absolute URL, then every time your page loads, this background pattern would need to be loaded from another server. This is considered bad practice and is certainly time consuming.[12] Suppose instead that you copy the image to your disk area. Now write a specification using a relative URL to generate the same background.

12 We know the Jennifer Jones example contained some code like this. For illustration purposes of URLs and since we did not expect to get caught by readers too often, we decided to leave it.

36. Write HTML code for a body tag in which you have gold unvisited hyperlinks, silver visited hyperlinks, and bronze active hyperlinks. Do you think this is a good combination to use? Why or why not?

37. Design a single-screen Web page about your favorite animal, using the tags described so far. That is, only use tags presented thus far in the book and do not omit any tags. Use a sensible set of colors. For example, if you love turtles, use a green background with a variation on yellow as a text color, or whatever else makes sense to you. Carefully think about your design and style when you code the page. Make sure to comment the code, explaining why you chose the colors you did.

38. Design a single-screen Web page about your favorite trail mix, using the tags described so far. (See previous exercise for further details.)

39. Describe three scenarios indicating when it would be a good idea to include a comment in an HTML file.

40. How does your browser handle embedded HTML tags within a comment? Does it try to render them?

2.7 HTML Formatting and Hyperlink Creation

At this point you should be creating and viewing simple HTML documents. This section introduces you to four more HTML tags. They are `<P>` for paragraph, `<Hi>` for heading `i`, `<A>` for anchor, and `<IMG>` for image. By using these tags, you can make your pages more readable, interesting, and polished. The most interesting of these tags is the anchor tag; it allows you to create hyperlinks to other Web pages.

Hot Topic

HTML tags describe the desired structure of a Web page, rather than exactly how it should look. For example, HTML tags identify emphasized text, headings, lists, and so on, not items like 11-point font and 0.75-inch margin. Because various browsers will render the same HTML code differently, it is important not to try to force a very specific layout on a Web page. Doing so may cause the page to look fabulous using one browser and awful using another. Keep this point in mind as you increase your repertoire of HTML tags.

2.7.1 Paragraph Tag

The paragraph tag is used to break the text into paragraphs. Most browsers place an empty vertical space between paragraphs so they stand apart from each other. To designate a block of text as a paragraph, enclose it within the paragraph beginning and ending tags: `<P>` and `</P>`. The ending tag is considered optional since (most) browsers assume that the current paragraph ends when the browser encounters the next `<P>`. Nevertheless, we recommend treating `<P>` as a paired tag, even though it is not mandatory. The following HTML code illustrates the use of the paragraph tag:

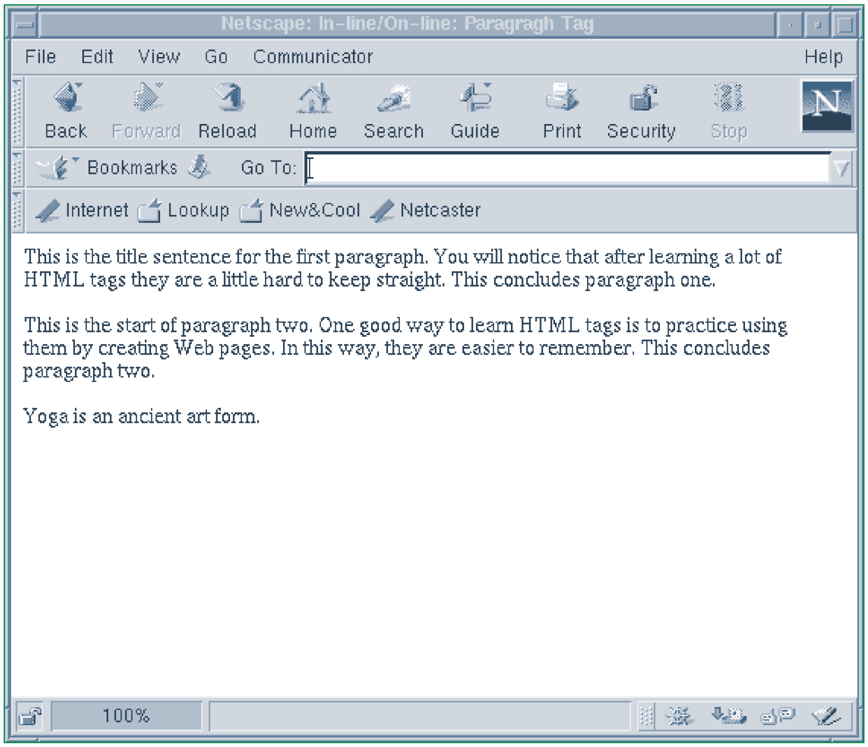
```
<P>
This is the title sentence for the first paragraph.
You will notice that after learning a lot of HTML tags,
they are a little hard to keep straight.
This concludes paragraph one.
</P>
<P>
This is the start of paragraph two.
One good way to learn HTML tags is to practice using
them by creating Web pages.  In this way, they are
easier to remember.
This concludes paragraph two.
</P>
<P>
</P>
Yoga is an ancient art form.
```

The last paragraph tag leaves a blank vertical space between the sample text and the succeeding HTML elements. One rendering of the HTML code for the paragraph example is depicted in Figure 2.7.

Some Web authors will put a series of paragraph tags adjacent to one another to skip some vertical space. For example,

```
<P> <P> <P> <P>
```

will cause most browsers to leave approximately 1 inch of vertical space. This is the type of HTML programming we generally recommend against. Do not try to force a precise spacing, because it may look exactly as intended on your desktop, but it could look quite different to someone else using a different window size and/or browser.


Netscape: In-line/On-line: Paragragh Tag
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Go To:
Internet Lookup New&Cool Netcaster
100%

This is the title sentence for the first paragraph. You will notice that after learning a lot of HTML tags they are a little hard to keep straight. This concludes paragraph one.

This is the start of paragraph two. One good way to learn HTML tags is to practice using them by creating Web pages. In this way, they are easier to remember. This concludes paragraph two.

Yoga is an ancient art form.

FIGURE 2.7 Sample rendering of the paragraph tag example.

2.7.2 Heading Tags

In many forms of writing, it is common to include section headings to provide the reader with a sense of the document's structure. When viewing any page of text, not just material on the Web, the first things most readers notice are headings and subheadings. As an example, glance at any page of this book containing a heading and observe that it stands out from the surrounding text.

Hot Topic

Most browsers support a hierarchy of six levels of HTML headings. The beginning tag for heading `i` is `<Hi>`, where `i` can be any value from `1` to `6`. The corresponding ending tag, as expected, is `</Hi>`. The largest heading is `<H1>` and the smallest is `<H6>`. Note that this is the reverse of the way sizes are specified in the font and basefont tags.

The heading tags are very useful for dividing a document into sections: the more important the section, the larger the heading tag. Subsections are usually less important and so receive smaller headings. Here is sample HTML code that illustrates the use of the heading tag.

```
<H1>Complete Sentences</H1>

Most of us would agree that well-written English ...

<H2>Fragments and Run-on Sentences</H2>

Keep in mind that rambling endlessly can lead to ...

<H3>The Phrase Fragment</H3>

Phrase fragment description goes here.

<H3>The Appositive Fragment</H3>

Appositive fragment description goes here.

<H4>Examples</H4>

There are two types of examples: sentence fragment and
sentence complete.

<H5>Sentence Fragment</H5>

<H5>Sentence Complete</H5>

<H6>Copyright, English Grammar for the Rest of Us</H6>
```

Figure 2.8 illustrates how one browser renders the example heading code. The browser determines the exact style of the headings; that is, the font size for each heading, whether the heading is boldface, whether the heading gets numbered, and so on. Notice that as the topics become more specialized, the heading sizes become smaller. Three or fewer levels of headings will suffice for most writing.

It is a fairly common practice to use level 5 or 6 headings for copyright notices and disclaimers, as Figure 2.8 shows. These headings are usually too small for anything else.

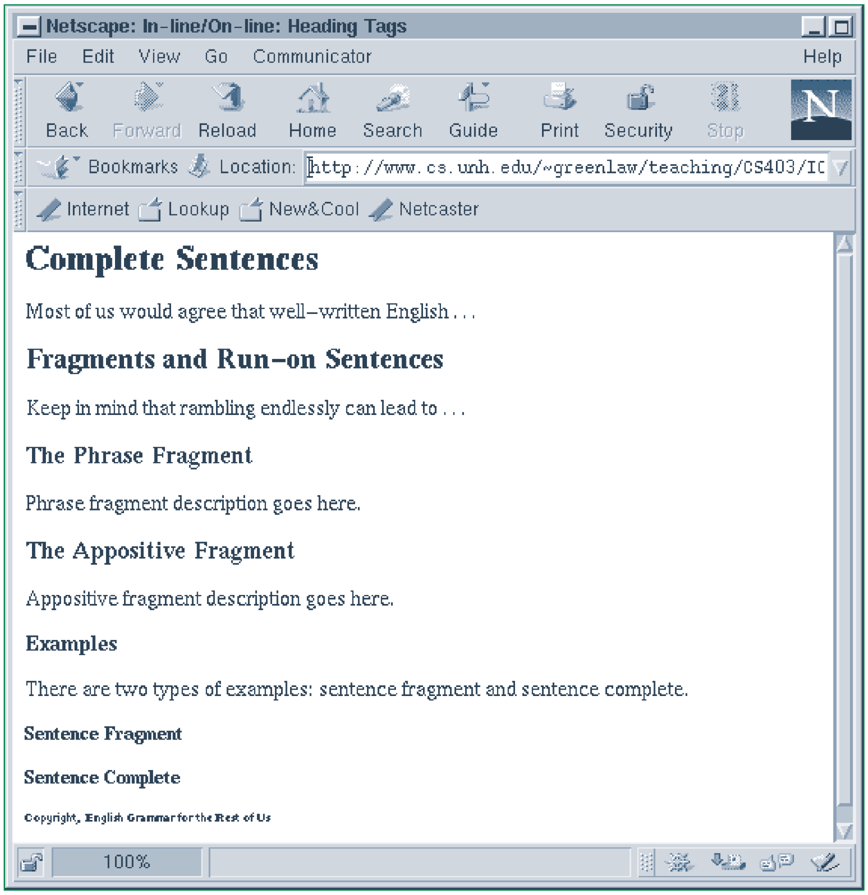
Useful Item

One important attribute of the heading tag is `ALIGN`, which can have values of `left`, `center`, or `right`. For headings that are not as long as the width of the document area, the `ALIGN` attribute has the expected effect. Caution should be exercised here. If the document is viewed in a small window, the `ALIGN` attribute may produce an undesirable appearance.

2.7.3 Anchor Tag

The anchor tag, `<A>` and `</A>`, is the mechanism by which hyperlinks are placed in hypertext documents. Its syntax is more complicated than that of most other tags. The term *anchor* is used because it indicates the static positioning of a hyperlink. In this section, we explain how to create clickable text hyperlinks, clickable images, `mailto` hyperlinks, and

FIGURE 2.8 Sample rendering of the heading tag example.


Netscape: In-line/On-line: Heading Tags
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Location: http://www.cs.unh.edu/~greenlaw/teaching/CS403/IC
Internet Lookup New&Cool Netcaster
100%

Complete Sentences

Most of us would agree that well–written English . . .

Fragments and Run–on Sentences

Keep in mind that rambling endlessly can lead to . . .

The Phrase Fragment

Phrase fragment description goes here.

The Appositive Fragment

Appositive fragment description goes here.

Examples

There are two types of examples: sentence fragment and sentence complete.

Sentence Fragment

Sentence Complete

Copyright, English Grammar for the Rest of Us

hyperlinks that point inside a document. We also provide some guidelines on hyperlink construction.

The three basic parts of a hyperlink are:

- The beginning and ending tag pair `<A>` · · · `</A>`.
- The `HREF` attribute that specifies the URL of the page to be loaded when the hyperlink is selected.
- The text (or graphic) that appears on-screen as the active link.

Clickable Text Hyperlinks

Consider the following example:

```
<A HREF = "http://www.usa.gov/documents/ ↩
         whitehouse.html">White House</A>
```

We have used the HREF attribute of the anchor tag to create a hyperlink labeled "White House." Several items about this hyperlink are worth pointing out. First, most graphical browsers will change the text color and underline the "name" of the hyperlink, which in this case is White House. When someone clicks on this phrase, the file /documents/whitehouse.html is requested from the server www.usa.gov. Second, notice that we did not leave any blank spaces before or after the text "White House." If you leave blank spaces, the hyperlink underline is drawn too wide. For example, the hyperlink,

```
<A HREF = "http://www.usa.gov/documents/ ↩
          whitehouse.html">   White House   </A>
```

results in the following underlining:

```
  White House  
```

Third, the phrase "White House" is short and descriptive. Most users would be able to guess quickly what type of information they would be retrieving by selecting this hyperlink. In your Web pages, use short, informative hyperlink names. Finally, notice that in our example, we used an absolute URL. It is possible to use relative URLs, as well.

Clickable Image Hyperlinks

The principles behind creating a clickable image are the same as for creating a clickable text hyperlink. The type of hyperlink we describe here consists of a single image, for which one mouse click returns an HTML document. (In Section 8.5, we discuss how to define images that can have a number of mouse-sensitive areas, each with the potential to return a different Web page.) The idea is simply to replace the clickable text with an image. As an example, consider using the file wheelbarrow.gif, which contains a 50-by-50 *pixel* image called "Under Construction."

```
<A HREF = "http://www.usa.gov/wogulis/notready.html">
   <IMG SRC =  "wheelbarrow.gif"
        ALT = "Under Construction"
        HEIGHT = "50"
        WIDTH = "50">
</A>
```

If a user clicks on the wheelbarrow image, the document named notready.html will be loaded. Browsers will typically draw a highlighted border two pixels wide around the image. It is not always completely obvious that an image is a hyperlink, so it is sometimes worthwhile to add text to alert the reader to this fact. This requires some thought in order not to defeat the purpose of using the image.

Mailto Hyperlinks

It is common practice to add a `mailto` hyperlink to a Web page. This provides a convenient method for someone viewing your page to send you email. Suppose Pascal Leno has an email address of

`leno@oli.rustica.it`

and wants to include a `mailto` hyperlink labeled "Contact Pascal" on his Web page. The following code does it:

```
<A HREF = "mailto:leno@oli.rustica.it">Contact Pascal</A>
```

Note that the syntax for the `mailto:` URL is different from the `http://` URL because the double slashes are not allowed.

When the user clicks on the hyperlink "Contact Pascal," a mail dialog box (this is not your usual mail client bundled with the browser) will be launched, with the *To* field filled in with `leno@oli.rustica.it`. All the user has to do is complete the remainder of the message and send it. This presupposes that you filled in your name and email address under options or preferences for your browser, or in a public room situation, that you have verified the settings and updated them to point to you.

Intradocument Linking

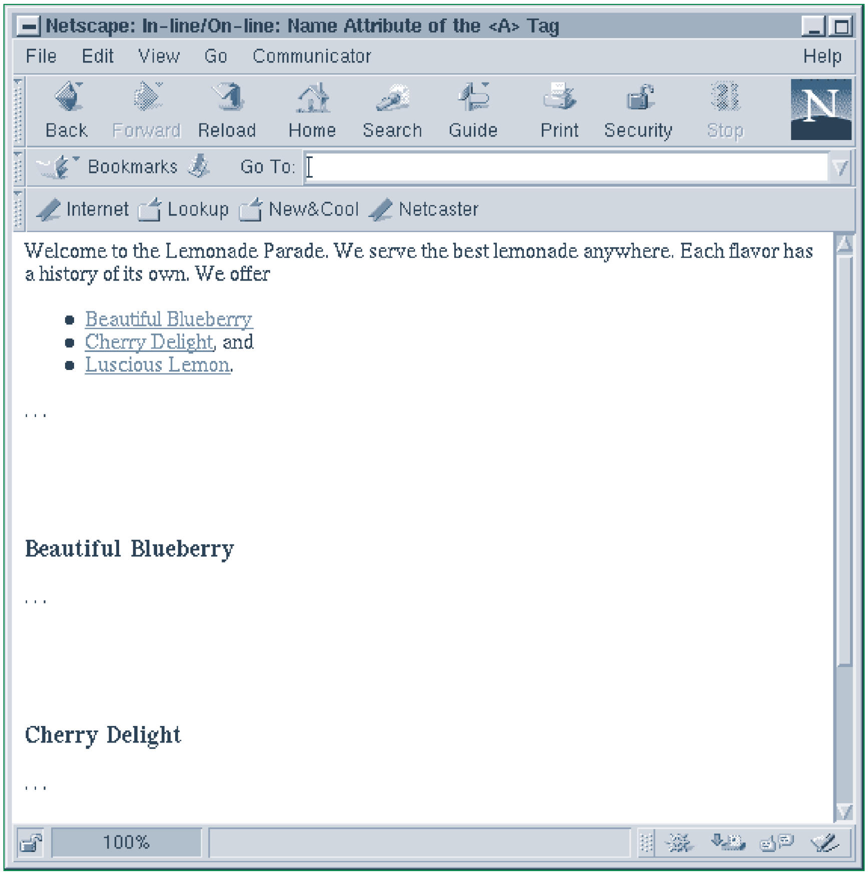
Another important attribute of the anchor tag is `NAME`. The `NAME` attribute lets you create a hyperlink to any part of your document, rather than just the beginning. That is, any portion of the document can automatically be displayed at the top of the browser's document area. This is particularly useful if you have a long Web page and would like users to be able to jump to various sections of it without scrolling. Many Web authors provide an index at the top of a page, with hyperlinks that jump to other parts of the document. Using `NAME` hyperlinks means you do not have to break the document down into pieces to allow the reader easy and rapid navigation.

How are `NAME` hyperlinks created? The following HTML code demonstrates the process:

```
Welcome to the Lemonade Parade.  We serve the best
lemonade anywhere.  Each flavor has a history of its
own.  We offer
<A HREF = "#blueberry">Beautiful Blueberry</A>,
<A HREF = "#cherry">Cherry Delight</A>, and
<A HREF = "#lemon">Luscious Lemon</A>.
...
```

```
<A NAME = "blueberry">
<H3>Beautiful Blueberry</H3>
</A>
...

<A NAME = "cherry">
<H3>Cherry Delight</H3>
</A>
...

<A NAME = "lemon">
<H3>Luscious Lemon</H3>
</A>
...
```

A rendering of this code is shown in Figure 2.9. Note that if the hyperlink is in the same file, as in our case, the URL does not need to be specified. You can begin the HREF value with the # symbol, as shown here.


Netscape: In-line/On-line: Name Attribute of the <A> Tag
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Go To:
Internet Lookup New&Cool Netcaster
100%

Welcome to the Lemonade Parade. We serve the best lemonade anywhere. Each flavor has a history of its own. We offer

- Beautiful Blueberry
- Cherry Delight, and
- Luscious Lemon.

...

Beautiful Blueberry

...

Cherry Delight

...

FIGURE 2.9 Sample rendering of the code used to illustrate the NAME attribute of the anchor tag.

The `NAME` attribute is used to label three separate sections of the document. Notice that between the beginning `<A>` and ending `</A>`, we have included the section heading (in this case, for the particular flavor of beverage described). You should always include at least one line of text between the anchor tags.

You can create a hyperlink to a label by using the URL for the file in which it is contained, followed by the `#` symbol and then the label name, as follows:

```
HREF = "http://systemname/docpath#labelname"
```

where `systemname/docpath` is the URL, and `labelname` is the actual label used.

What is the effect of selecting one of these hyperlinks? If a user clicks on the Luscious Lemon hyperlink, for example, the cursor immediately moves down to the section labeled Luscious Lemon. Here, this section is contained in the next screen. This feature obviates the need for scrolling and provides the user with a more convenient method of accessing different parts of a long document.

Hyperlink Creation Guidelines

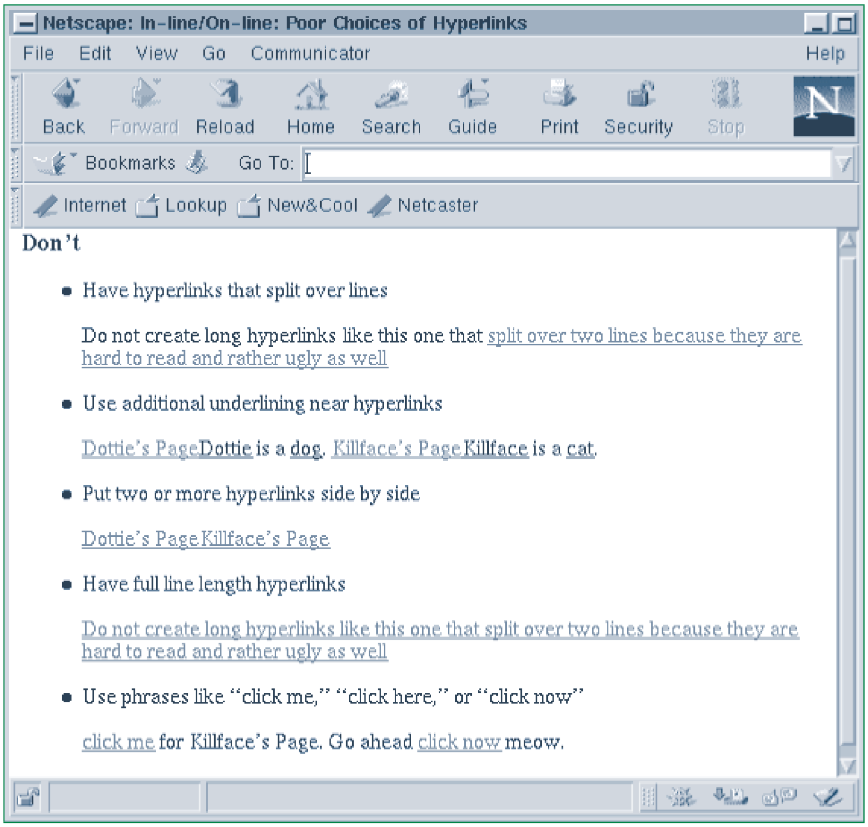
Do:

- Carefully choose the text that goes into a hyperlink.
- Create hyperlinks that are interesting.
- Keep the hyperlink text short and descriptive.
- Create hyperlinks that read well together, even without any intervening text.

Don't:

- Use hyperlinks that may split over two or more lines.
- Use additional underlining near hyperlinks.
- Put two or more hyperlinks side by side.
- Use full-line-length hyperlinks.
- Use such phrases as "click me," "click here," or "click now."

Figure 2.10 illustrates the Don't list.


Netscape: In-line/On-line: Poor Choices of Hyperlinks
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Go To:
Internet Lookup New&Cool Netcaster

Don't

- Have hyperlinks that split over lines

  Do not create long hyperlinks like this one that split over two lines because they are hard to read and rather ugly as well

- Use additional underlining near hyperlinks

  Dottie's PageDottie is a dog. Killface's PageKillface is a cat.

- Put two or more hyperlinks side by side

  Dottie's PageKillface's Page

- Have full line length hyperlinks

  Do not create long hyperlinks like this one that split over two lines because they are hard to read and rather ugly as well

- Use phrases like "click me," "click here," or "click now"

  click me for Killface's Page. Go ahead click now meow.

FIGURE 2.10 Examples illustrating poor choices of hyperlinks.

2.7.4 Image Tag

The image tag, `<IMG>`, is used for including in-line images in HTML documents. An example of the use of the tag is:

```
<IMG SRC = "wheelbarrow.gif"
     ALT = "Under Construction"
     HEIGHT = "50"
     WIDTH = "50">
```

We will explain the meaning of this code, a number of attributes of the image tag, and several style issues. Chapter 8 contains a more comprehensive treatment of images and graphics.

The most important attribute of the image tag is `SRC`, which is used to specify the image to be displayed. Any type of image can be specified, using either a relative or an absolute URL, where the relative URL would relate to the document in which the image appears. The most common image types on the Web are `gif`, `jpg`, and `png` where `png` is a distant third. To include the image `friend.gif` in a file located in the same directory, you could use the following HTML code:

```
<IMG SRC = "friend.gif">
```

This is the minimum amount of code you can use to include an image.

When a browser retrieves a Web page, it does not automatically get the images that go along with that page. Each image must be retrieved separately. To render the document on-screen, the browser must know the sizes of the images. The browser can obtain these sizes either from code entered by the image's developer (as we recommend), or by reading the sizes as the images are brought over. The latter case takes longer, because the browser has to do the interpreting. So, a Web page will render more quickly if you include the size, using the `HEIGHT` and `WIDTH` attributes of the image tag, for each image.

Image dimensions nearly always are expressed in *pixels*. Suppose the picture `friend.gif` is 60 pixels wide and 90 pixels high. The following code would include the image in the Web page and would specify its dimensions to the browser:

```
<IMG SRC = "friend.gif"
     HEIGHT = "90"
     WIDTH = "60">
```

While the order in which you specify the `HEIGHT` and `WIDTH` attributes is not significant, we usually specify them in the order shown. On the other hand, a browser will list image dimensions as $x \times y$, where x is the `WIDTH` of the image and y is the `HEIGHT`.

When a browser parses this HTML code, it can determine how much space to leave for the image. Thus, the surrounding text can be rendered immediately. The browser does not have to wait until the image itself arrives. This is why you often see all of the text in a page long before all of the images are rendered.

How can you determine the size of an image? Most browsers have a "document info" menu item where you can find the dimensions of an image (in pixels). Of course, if you create an image yourself, you can record its size at that time.

The `HEIGHT` and `WIDTH` attributes can also have percentages as values, allowing them to be used to scale an image relative to the size of the browser's window. For example, the following code produces a version of `friend.gif` that occupies 50 percent of the browser's window in each dimension.

```
<IMG SRC = "friend.gif"
     HEIGHT = "50%"
     WIDTH = "50%">
```

You can create some interesting scaling effects by using percentages. However, when you scale an image downward, you do not reduce the amount of disk space required to store the image. Thus, it is not possible to create a reduced-size *thumbnail sketch* by using percentage values for the HEIGHT and WIDTH attributes.

Useful Item

The image tag has another interesting attribute known as ALT, which is short for "alternative." The value of ALT is a text string that usually describes the image in words. In our wheelbarrow.gif example, we used the words "Under Construction" as a value of the ALT attribute, because wheelbarrow.gif contains the picture of a wheelbarrow at a construction site. You have probably seen many Web pages that have "Under Construction" images. If a browser has images turned off, or is text-only, the words in the ALT attribute will be displayed on-screen where the image would have been. Obviously, the size of the image will not exactly match the text replacing it. Most Web authors do not worry about this detail, as most of their effort goes into making the pages look good with the images displayed.

Go On-Line

In the latest versions of some browsers, when the user mouses over an image, the text in an ALT attribute is displayed in the form of a *tooltip*. This is usually a light-colored dialog box that is just large enough for the text. For pages that contain a lot of images, the tooltips can become a distraction. This has prompted some users to stop including ALTs. Remember, however, that the purpose of ALTs is to serve those users who are unable to display images, or who are using text-based browsers. In such cases, the text of an ALT can provide the reader with some continuity.

Occasionally, square brackets have been used around the ALT attribute value. This mimics the convention adopted by some text-based browsers. However, on some browsers on some systems, this can create a problem, so we recommend against this practice. For example, do not use:

Hot Topic

```
ALT = "[Under Construction]"
```

Instead use:

```
ALT = "Under Construction"
```

As another attempt at presenting an image in a text-based browser, some extremely clever Web authors place an ASCII graphic of the original image in the ALT tag. This is interesting, but very time consuming unless you can locate a free copy of the ASCII graphic you want to include.

Useful Item

With the material we have covered so far, you should be able to include images on your Web pages. Remember to set protections on your image files, similar to what you did for your index.html file.

EXERCISES 2.7

HTML Formatting and Hyperlink Creation

41. Create an HTML document that contains the paragraph tag. What happens if you include five paragraph tags in a row, with no intermediate text? Does the paragraph tag have any attributes? If so, explain and provide examples.

42. Create an HTML document that uses heading tags. As an example, the document could outline a recent or planned backpacking trip. Focus on the outline and on the use of the heading tags, rather than on the details of the trip.

43. Does your browser support six different-sized headings, or are some of them the same size? What happens if you try to close an `<H1>` tag with `</H2>` instead of `</H1>`? Report the URL of a Web page that makes effective use of headings.

44. Explain what happens if you use an `ALIGN` value of `center` in an `<H4>` tag that surrounds text that is wider than the browser's document area.

45. Create an HTML document containing hyperlinks to three Web pages about the Appalachian Trail.

46. Create a Web page containing a hyperlink to itself. What happens when you click on the hyperlink? Explain.

47. Produce a Web page that has hyperlinks to three of your friends' Web pages.

48. Design a Web page that contains a `mailto` hyperlink to you. Test it. Were you able to send yourself email?

49. Create a Web page that can be used to link in the assignments for this course.

50. Code an HTML document that contains a clickable image.

51. Locate an "Under Construction" image and include it on a Web page. In addition to `SRC`, be sure to use the `ALT`, `HEIGHT`, and `WIDTH` attributes of the image tag.

52. Experiment with scaling an image. How small can you scale an image before it becomes "fuzzy"? Does this depend on the image quality with which you started? How large can you expand an image? What happens to the quality of the image as it gets larger?