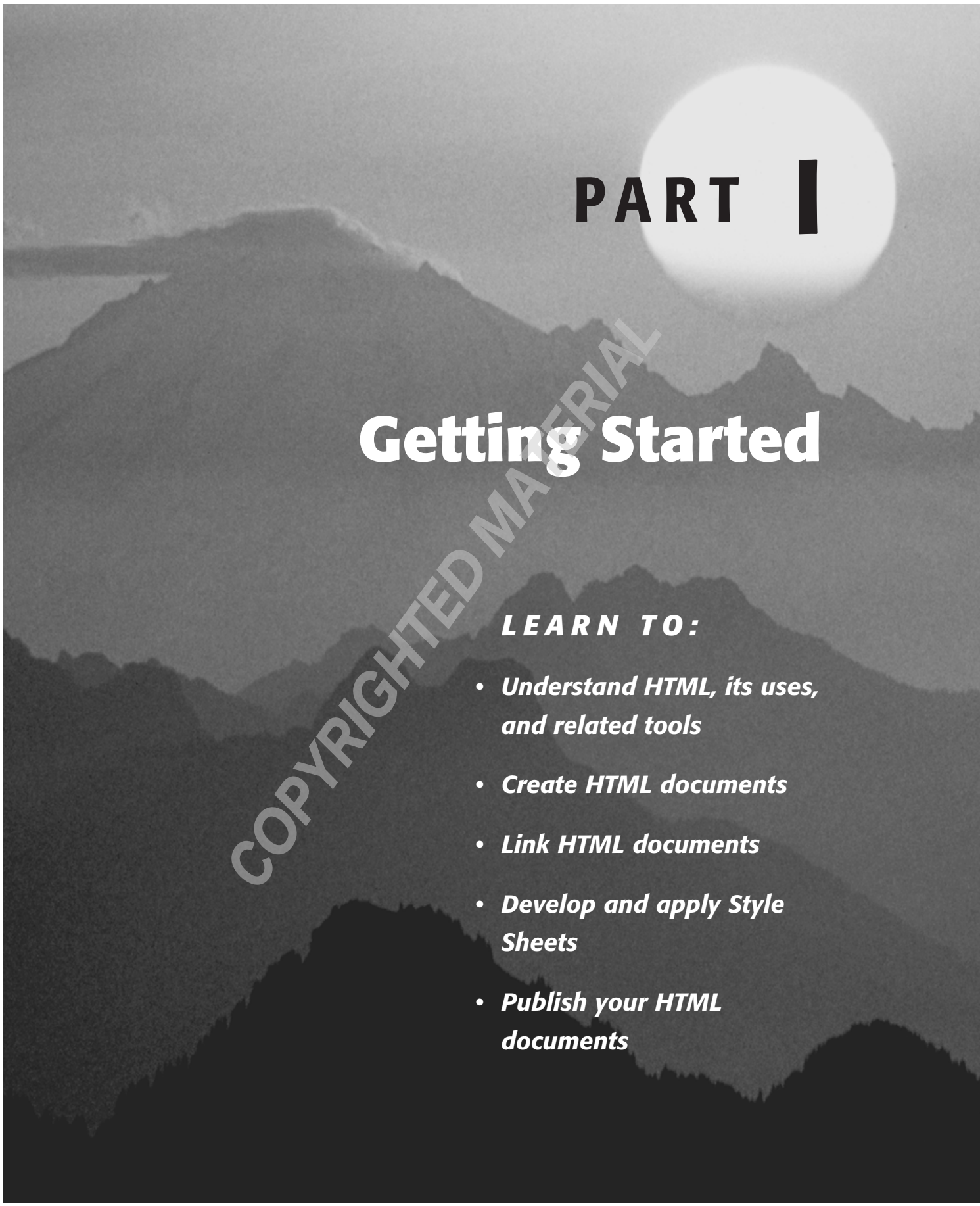


PART I

Getting Started

LEARN TO:

- ***Understand HTML, its uses, and related tools***
- ***Create HTML documents***
- ***Link HTML documents***
- ***Develop and apply Style Sheets***
- ***Publish your HTML documents***

CHAPTER 1

Getting Acquainted with HTML, Its Tools, and Its Resources

FEATURING:

Understanding HTML and what it looks like 4

Examining some HTML uses 9

Finding the tools you'll need to get started 9

Using other helpful resources 14

HyperText Markup Language, or HTML, is a system of codes that you use to create interactive, online pages. You're probably familiar with the most common HTML application—World Wide Web pages. Visitors can jump from topic to topic, fill out forms, submit information, and search databases, among other things, rather than slogging through information linearly. HTML, though, is not what you see onscreen as a Web page. Instead, HTML is the behind-the-scenes code that tells browsers what to display.

NOTE Throughout this book, we use the term *visitors* to describe the people who use the HTML documents you develop.

In this chapter, we'll introduce you to HTML—what it looks like, what you use it to accomplish, and what tools you need to get started—and give you a foundation for learning to use HTML throughout the rest of this book.

What Is HTML?

HTML is a system of codes that identify parts and characteristics of documents. As Figure 1.1 shows, HTML documents are plain text files. They contain no images, no sounds, no videos, and no animations; however, they can include "pointers," or links, to these file types, which is how Web pages end up looking as if they contain nontext elements.

As you can see, HTML documents look nothing like the Web pages you have likely seen before. Instead, HTML is made up of *tags* and *attributes* that work together to identify document parts and tell browsers how to display them. Figure 1.2 shows how the tags and attributes of Figure 1.1 work together to create a Web page.

FIGURE 1.1

HTML documents are just text files, containing the code and content you provide.

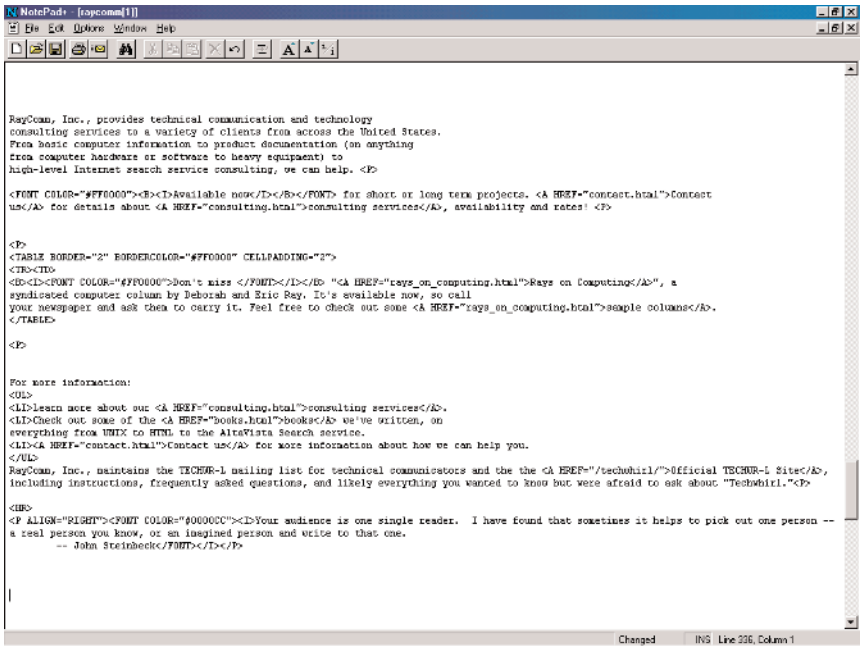
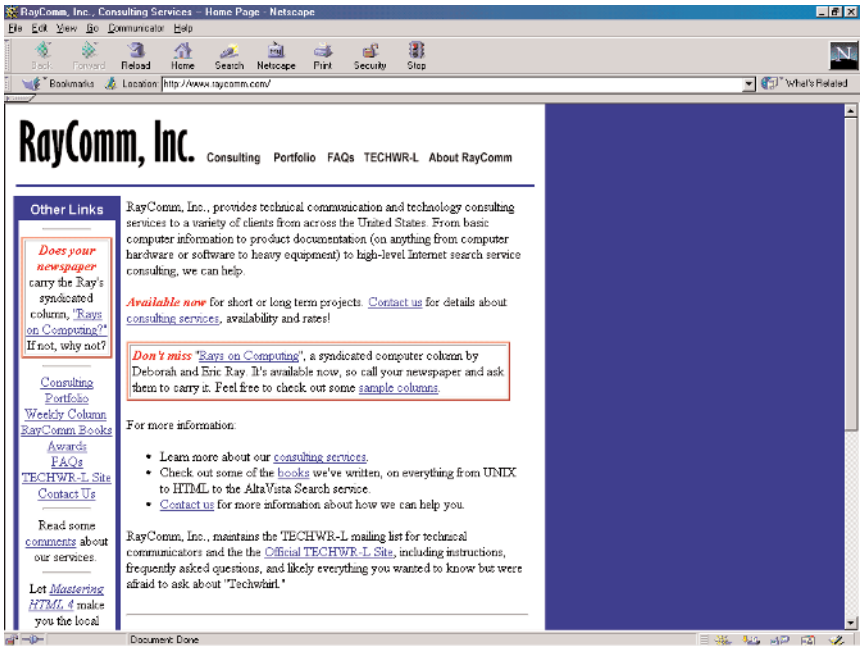


FIGURE 1.2

Browsers interpret the code to determine how to display HTML documents.



Understanding Tags

HTML tags actually serve two purposes. First, they identify logical document parts—that is, the major structural components in documents such as headings (<H1>, for example), numbered lists (<OL>, for Ordered List), and paragraphs (that is, <P>). So, for example, if you want to include a heading, a paragraph, and a list in your document, you type the text and apply the appropriate tags to it (we'll show you how in the next sections). And that's it.

Second, tags in HTML documents also reference other elements—that is, tags can include pointers and links to other documents, images, sound files, video files, multimedia applications, animations, applets, and so on. For example, if you want to include an image of your company's product in your HTML document, rather than pasting in an image (as you might in a word processing document), you include a tag that points to the image filename, as shown here:

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

In this example, the <IMG> (image) tag points to a logo file (`logo.gif`) that the browser should display. So, here again, browsers rely on the information contained within an HTML document to tell it what to display as well as how to display it.

Understanding Tag Components

As you can see from these examples, HTML tags are fairly intuitive. Although tags are occasionally cryptic, you can usually get an idea of a tag's function just from its name. Let's take a look at tag components, which should help you learn and apply tags to your HTML documents.

First, all tags are composed of *elements* that are contained within *angle brackets* (< >). The angle brackets simply tell browsers that the text between them is an HTML command. Some sample tags look like these:

- <H2> (for heading level 2)
- <BODY> (for document body)
- <B> (for bold)

NOTE You'll learn more about these tags and their uses in Chapter 2.

Second, most tags are paired, with an opening tag (for example, <H1>) and a closing tag (</H1>). Both tags look alike, except the closing tag also includes a forward slash (/). To apply tags to information in your document, place the opening tag before the information, and place the closing tag after the information, like this:

```
<H1>information that the tags apply to</H1>
```

Enter both the opening and closing tag at the same time so that you don't forget the closing tag. If you happen to forget it, most paired tags will run on and on until the browser finds a matching closing tag.

NOTE Less commonly, you'll use nonpaired tags, which do not include the closing tag. We'll point those out throughout this book and show you how to use them appropriately.

To apply more than one tag to a chunk of information, you nest the tags. *Nesting* means placing one set of tags inside another set. For example, to apply italic to a heading, you nest the tags, like this:

```
<H1><I>information that the tags apply to</I></H1>
```

or like this:

```
<I><H1>information that the tags apply to</H1></I>
```

When you nest tags, the first tag is also the last tag, and the second tag is also the second to last tag, and so on.

Typing Tags Correctly

When typing tags, be particularly careful not to include extra spaces. If you do so, a browser may not recognize the tag and will not display the information correctly. Or, the browser might display the tag itself.

For example, a title should look like this:

```
<TITLE>Correctly Formed Title</TITLE>
```

Do *not* include spaces within the tags, like this:

```
< TITLE >Incorrectly Formed Title< /TITLE >
```

Improving Readability

You'll find it easier to read and use tags if you follow a few conventions. In particular, we suggest that you type tags using all caps and that you use hard returns to create shorter lines. These conventions do not affect how browsers display code—they just make it easier for you to read.

The following two examples show you how using caps and hard returns can improve readability.

```
<!doctype HTML public "-//w3c//dtd HTML 4.0//en"><html><head>
<title>Mastering HTML Document Title</title></head><body>Mastering HTML
Document Body</body></html>
```

or

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>
    <TITLE>Mastering HTML Document Title</TITLE>
  </HEAD>
  <BODY>
     Mastering HTML Document Body
  </BODY>
</HTML>
```

No question, right?

Understanding Attributes

Some tags work in conjunction with attributes, which provide additional information about an element, such as how elements should align, what other files should be accessed, or even the color of an element. For example, an attribute might indicate that a heading should appear centered in the browser window, that the browser should load an image file from the Web, or that the Web page background should appear sky blue.

So, suppose you want to center a heading in the browser window. You start with your heading and tags, like this:

```
<H1>A heading goes here</H1>
```

and then add the ALIGN= attribute to the opening tag, like this:

```
<H1 ALIGN="center">A centered heading goes here</H1>
```

All attributes go in the opening tag and are separated from other attributes and the tag itself by a space. Some attributes require quotes; some don't. As a general rule, most attributes—those that include only letters, digits, hyphens, or periods—work fine without quotes. For example, you can type ALIGN=CENTER or ALIGN="CENTER"; all browsers should display these in the same way.

Attributes that have other characters, such as spaces, % signs, or # signs, however, *always* require quotes. For example, if you use the WIDTH= attribute to indicate a percentage of the document window, type WIDTH="75%".

TIP When in doubt, use quotes with attributes. Although they aren't always necessary, they never hurt. Throughout this book, we've included the quotes—for good practice and good example.

You can include multiple attributes in a tag by using one space between each attribute, like this:

```
<H1 ALIGN="center" SIZE="+2" COLOR="#FF0000">A wildly formatted heading goes here</H1>
```

NOTE In HTML tags, the attributes can go in any order after the element, but the element must always go first.

What Can You Do with HTML?

You're likely most familiar with HTML as it's used to create Web pages; however, HTML has expanded to include many other uses:

Developing intranet or extranet sites HTML is commonly used to develop intranet and extranet sites, which are company-wide Web sites that are accessed by people within the company from one or more locations.

Developing help files HTML is also used to develop online help files, allowing developers to inexpensively produce documentation that is accessible on any platform.

Developing network applications HTML is particularly suitable for creating entire applications, such as training programs, interactive chats, or databases that are available through Web pages.

Developing kiosk applications Finally, HTML can also be used to create kiosk applications—those stand-alone computers with the neat touch-screen capabilities.

What Tools Do You Need?

For your first documents, you need only two basic tools:

- An HTML editor, to create and save your HTML documents
- A Web browser, to view and test your HTML documents

HTML Editors

In general, HTML editors fall into two categories:

- Text- or code-based, which allow you to see the HTML code as you're creating documents
- WYSIWYG (What You See Is What You Get), which show the results of code, similar to the way it will appear in a browser, as you're formatting your document

Although dozens of excellent WYSIWYG editors are available, you should learn to code HTML using a standard text editor. Text editors force you to *hand-code* HTML, meaning that you, not the software, enter tags and attributes. Hand-coding helps you learn HTML tags and structure and lets you see where you've made mistakes. Also, with hand-coding, you can easily include the newest HTML enhancements in your documents. Notepad for all Windows versions, vi or pico for Unix, and TeachText or SimpleText for Macintosh are good choices.

TIP Learning to hand-code is essential for using the latest-and-greatest HTML effects—whether it's the current HTML 4 or a future version. Most new HTML versions are not immediately supported by WYSIWYG editors, so you would need to hand-code those tags and attributes in your documents.

WARNING Using a word-processing program such as Word, WordPerfect, or even WordPad to create HTML documents can often introduce extra formatting and control characters, which will cause problems. HTML requires plain text with no formatting at all, so either make special effort to save all documents as plain text, or just use a text editor.

Simple WYSIWYG editors, such as Netscape Composer and Microsoft FrontPage Express (included with Internet Explorer 4 and 5), are good for quickly generating HTML documents. These editors only give you a close approximation of page layout, design, and colors, but are good for viewing the general arrangement of features. They do not, however, give you, the author, as much control over the final appearance of your document as code-based editors do.

After you've developed a few HTML documents and understand basic HTML principles, you may choose to use both a WYSIWYG editor and a code-based editor. For example, you can get a good start on your document using a WYSIWYG editor and

then polish it (or fix it) using a code-based HTML editor. For now, though, we recommend that you hand-code HTML using a standard text editor.

Web Browsers

If you've ever surfed the Web, you've used a Web browser to view HTML documents. The most common browsers are Netscape Navigator and Microsoft Internet Explorer, although a variety of browsers are available for virtually all computer platforms and online services.

Exactly how your documents appear, though, will vary from browser to browser and from computer to computer. For example, most browsers in use today are *graphical browsers*: They can display elements other than text. A *text-only* browser can display—you guessed it—only text. How your HTML documents appear in each of these types of browsers differs significantly, as shown in Figures 1.3 and 1.4.

FIGURE 1.3

An HTML document displayed in Netscape Navigator

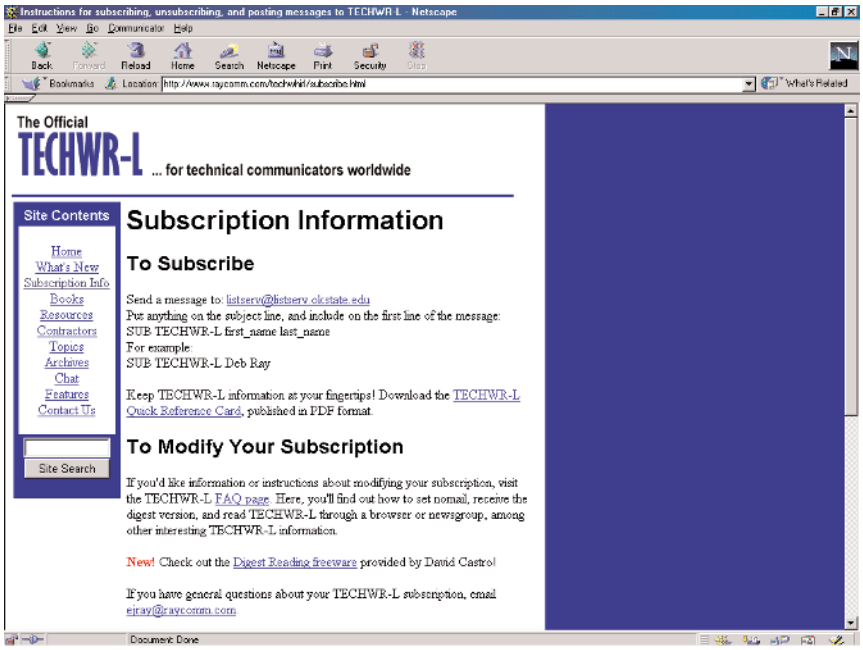
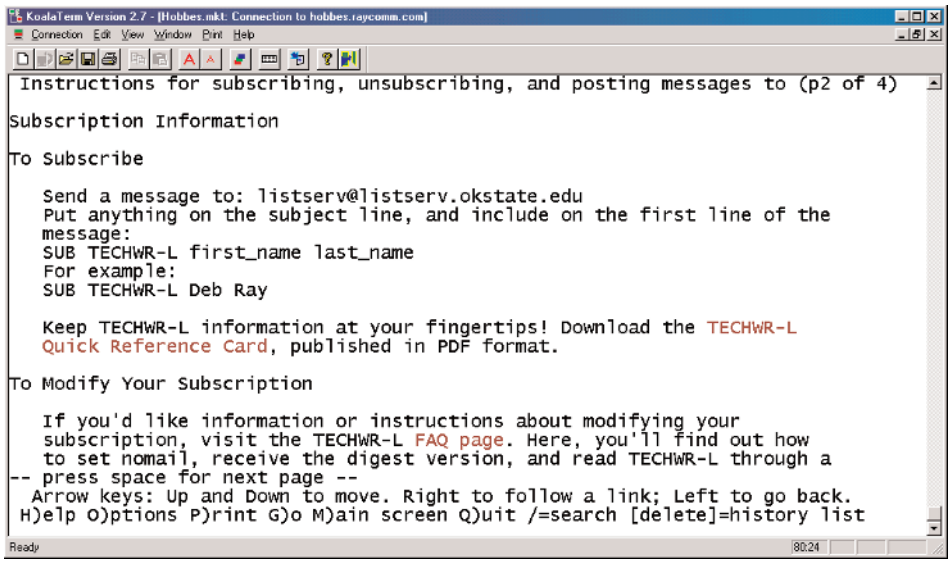


FIGURE 1.4

The same HTML document viewed in Lynx, a text-only browser



Additionally, even graphical browsers tend to display things a bit differently. For example, one browser might display a first-level heading as 15-point Times New Roman bold, whereas another browser might display the same heading as 14-point Arial italic. In both cases, the browser displays the heading as bigger and more emphasized than regular text, but the specific text characteristics vary. Figures 1.5 and 1.6 show how two other browsers display the same HTML document.

Finally, your visitor's computer settings can also make a big difference in how your HTML documents appear. For example, the computer's resolution and specific browser settings can alter a document's appearance, as shown in Figure 1.7.

So, as you're developing and viewing your HTML documents, remember that your pages will likely look a bit different for your visitors. If possible, test your documents in as many different browsers at as many different resolutions and color settings on as many different computers as possible. You won't be able to test for all possible variations, but you will be able to get a good idea of what your visitors might see.

FIGURE 1.5

The old NCSA Mosaic browser displays the HTML document with a few problems.

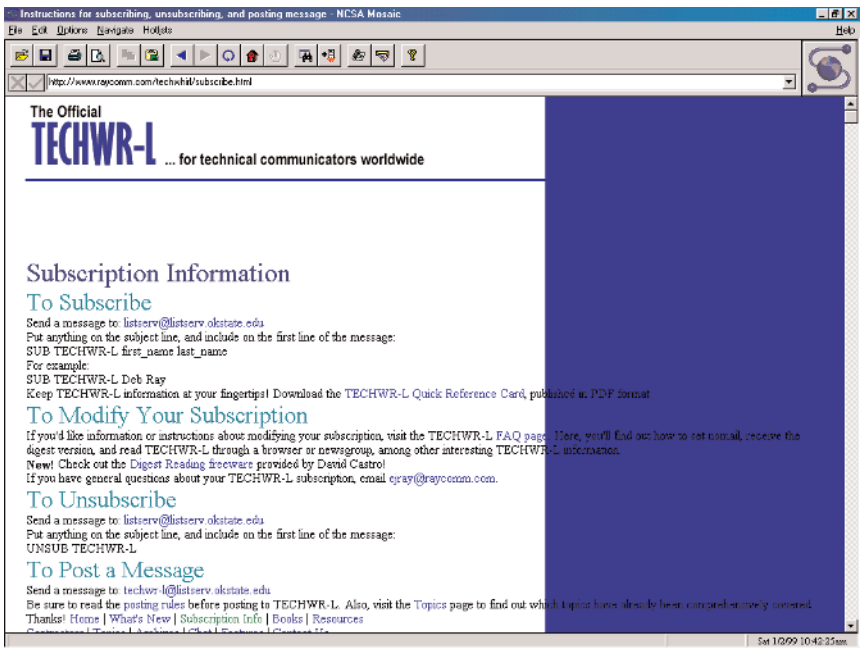


FIGURE 1.6

The Opera browser shows the same document with slightly different formatting.

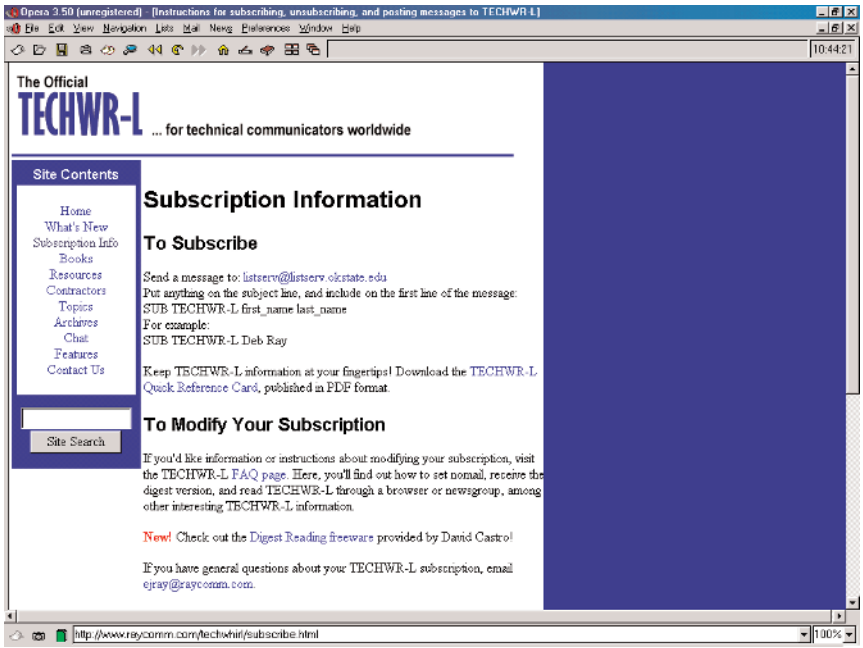
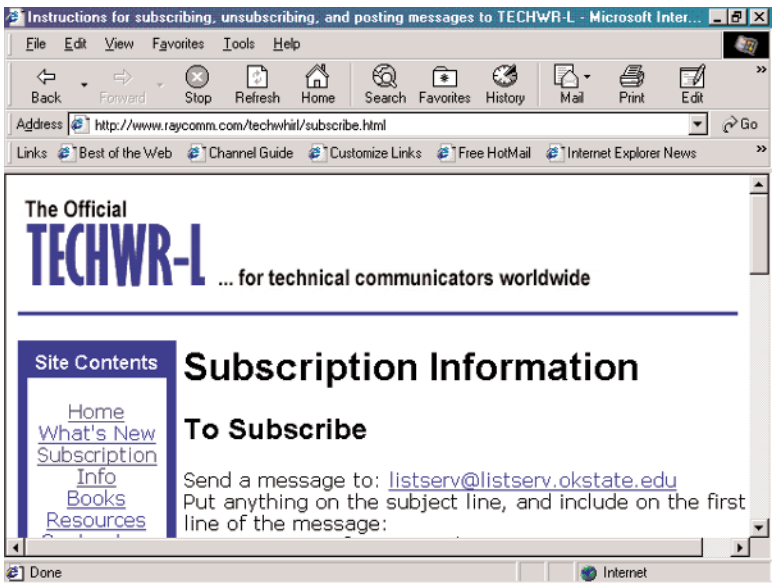


FIGURE 1.7

Internet Explorer and very low resolution combine to make the document look completely different.



What Other Resources Can Help?

In addition to this book, you can find information, resources, and specifications on the Web. In particular, the World Wide Web Consortium site, as well as several product-specific Web sites will help you learn, use, and keep up with changes in HTML.

Visit the W3C

The *World Wide Web Consortium (W3C)* was founded in 1994 at the Massachusetts Institute of Technology (MIT) to oversee the development of Web standards, including the HTML standard. This consortium defines and publishes HTML standards, the tags and attributes within HTML documents. So, an excellent way to monitor HTML changes is to visit the W3C site:

```
www.w3.org/
```

Here, you'll find new releases of HTML standards, proposed standards, and other developments in Web-related specifications, such as the Cascading Style Sheets (CSS) and eXtensible Markup Language (XML) specifications.

Can you use new tags and attributes as they become available? For the most part, yes. By the time many popular tags and attributes become part of the standard, they already have browsers' support. Some tags and attributes, however, including some that were introduced with HTML 4, did not have wide or stable browser support when the specification was released and do not to this day have nearly the breadth of support that other tags do. We'll point these out throughout this book and show you how they differ from the previous versions of HTML.

Monitor Netscape and Microsoft Sites

Each time that Netscape and Microsoft release a new browser version, look for new HTML *extensions*, which are browser-specific tags and attributes. Some of these extensions are useful, and some less so, but as a whole any nonstandard tags introduced into HTML cause problems both for Web developers and for visitors. Fortunately, fewer extensions seem to be introduced with each new release, but you should still be aware of what's added with each release.

If you're considering using extensions in your HTML documents, keep in mind that they're not standard HTML. That is, the extension is specific to a particular browser (say, Netscape) and will likely not work in other browsers (say, Internet Explorer or Opera or others). For this reason, we generally recommend that you don't use extensions and, instead, use only HTML standard tags and attributes so that all your visitors can access the information you provide.

TIP At the time of this writing, about 50 percent of Net surfers used Netscape Navigator, about 40 percent used Internet Explorer, and the remaining 10 percent used a variety of other browsers. Realistically, about 80 to 90 percent of all Web users can access the majority of sites incorporating the latest HTML tags and recent enhancements.

You can find Netscape's extensions at:

```
developer.netscape.com/index.html
```

And you will find Microsoft's extensions at:

```
www.microsoft.com/windows/ie/
```

Monitor Other Sites

Although definitive information comes from the W3C, Microsoft, and Netscape sites, you can reference other reliable resources too. Table 1.1 gives you a list of some to check regularly.

TABLE 1.1: SOME SITES THAT PROVIDE UP-TO-DATE HTML INFORMATION

Organization	URL
Web Design Group	`www.htmlhelp.com/`
Web Developer's Virtual Library	`www.stars.com/`
HTML Writer's Guild	`www.hwg.org/`
C|Net's Builder.com	`www.builder.com/`

Where to Go from Here

This chapter gave you a brief overview of HTML—what it is, what it's used for, and how it came about. Although you haven't yet done any HTML coding, you should have a good foundation for getting started with it.

From here, you have several options:

- If you are an HTML beginner, move on to Chapter 2 to create your first HTML document, building on the information you've learned here.
- If you are an intermediate HTML user, you might move on to Chapter 3, which addresses linking HTML documents, or you might select topics from Part II.
- If you are an advanced user, you might browse the Index or Table of Contents and select topics that interest you.