Demystifying Your (Future) Home Page

> **After completing this lesson, you will be able to:**
>
> - *Explain the difference between the Internet and the World Wide Web.*
> - *Understand the anatomy of a typical Web page,*
> - *Create a simple HTML page and view it in a browser.*

Lingo: The *Internet* is the hardware that's connected together to create a massive worldwide network.

Most likely, you're fairly familiar with the not-so-newfangled invention called the *Internet*. Further, we're willing to bet that if you're contemplating the idea of creating a Web presence, then you know how to use a computer on some level. We're also assuming that you've surfed the Web, you can use basic applications (such as word-processing packages), and you can click a mouse with the best of 'em. Fortunately, your basic computing knowledge is all you need to be able to create Web pages—well, your basic computing knowledge along with these lessons, of course!

Your first job on the road to becoming a Web page developer entails building on what you already know. For instance, in addition to moderate computing capabilities, you should have an inkling of how the Internet, the Web, and Web pages relate to one another. Therefore, in the spirit of our goal of clarity and simplicity, we'll cut to the chase in this lesson and briefly describe the main elements of the world's largest network—the Internet, the Web, and Web pages. After we get the fundamentals out of the way, we'll spend the remainder of the lessons talking about planning and building your Web pages.

The Internet—Just a Bunch of Hardware

To put it simply, the Internet, or the Net, is hardware—lots of hardware—connected together to create a massive worldwide network. The Internet's hardware encompasses all the components a person can physically touch, including computers, *routers*, cables, telephone lines, high-speed data circuits, and other physical network pieces.

Lingo: *Routers* are relay components between networks.

For now, that's really all you need to know about the Internet—it's the hardware. No need to regale you with a long diatribe about how the U.S. government's Cold War paranoia spurred the development of a noncentralized computer network. If you're curious about the history of the Internet, you can find information online and at your local bookstore or library. (Also, see the resource section at *www.creationguide.com/resources* for some useful links.) Now that we've clearly identified that the Internet is the hardware, let's take the next logical step. Like all computer hardware (think of your desktop or laptop computer), the Internet needs software—otherwise, the Internet's hardware components would simply sit and gather dust (for the most part) on a worldwide basis. Enter the World Wide Web.

The Web—Some Software for the Hardware

Lingo: The *Web* consists of software that enables information sharing on the Internet.

The World Wide Web (also known as WWW or just "the Web") is a little more esoteric than the Internet. That's because the Web consists of software (including programs, documents, and files) that enables information to travel along the Internet's hardware. To help illustrate the Web's role relative to the Internet, here's a short story we first told a few years ago when explaining the role of the Web to Internet newbies:

> Long ago (back when insects and arachnids could talk), there lived a spider of unusually bright intellect named Tim. After watching the ants work all day, Tim

met up with the lead ant at the time, Bill. The ants, as usual, were incredibly successful at gathering and storing food, but Tim thought the spiders could team up with the ants to make life easier for both groups. Tim approached Bill with this plan, and Bill saw the logic in it. In fact, Bill suggested that they incorporate other creatures into the workgroups as well. Soon, Tim and Bill recruited grasshoppers, flies, and earthworms to become partners in the food-gathering venture. The creatures thought it was a splendid idea, so they got together and created an elaborate labyrinth of anthills, spiderwebs, burrows, and tunnels to assist in the food-gathering venture. The system was in place; it looked perfect; it was time for the work to begin. But, much to the creatures' disappointment, chaos ensued. Even though all the paths and connections were in place, flies had a hard time navigating the tunnels, grasshoppers had difficulty staying in line, earthworms were just too heavy to walk across the spiderwebs, and, of course, the ants' expectations were much too high for any of the other groups to meet. What the creatures had was a network. What they needed was something or someone who could cross all mediums of the network safely. They needed a universal creature.

This short story provides a good analogy of the Internet–Web relationship. As we said earlier in this lesson, the Internet is the infrastructure for transmitting information—an infrastructure made up of computers, routers, cables, telephone lines, high-speed data circuits, and information bases called *servers* (rather than anthills, spiderwebs, and tunnels). Unfortunately, just as spiderwebs can't support earthworms, not all computers can support all computer file formats. To include every available method (or *protocol*) for understanding the various document formats on all computers would be impractical. So, the Internet community devised its own universal creature, more commonly known as the World Wide Web.

> **Lingo:** *Servers* are powerful high-capacity network-linked computers that store files and respond to users' requests to view and access the stored files. A *protocol* is a set of rules that describe how data should be transmitted. The Web uses Hypertext Transfer Protocol (HTTP) to transmit Hypertext Markup Language (HTML) documents.

Initially, Tim Berners-Lee conceived and developed the Web at the CERN laboratory in Switzerland for the high-energy physics community. (By the way, although Tim is considered to be of extremely high intellect, he is not a spider!) The Web quickly attracted a great deal of attention and spread beyond the physics arena. As with the history of the Internet, you can find reams of information about the history of the Web online and in numerous computer books.

For our purposes, you only need to know that the Internet is the hardware and the Web is the software. Simple enough. Now, we're ready to move to the next level—the files the Web software supports on the Internet hardware.

Web Pages—A Few Files on the Net

Now we come face-to-face with the heart of the matter—Web pages. Basically, when you strip away all the highfalutin technobabble, Web pages are files. To be specific, Web pages are HTML files. No need for your eyes to glaze over at the sight of "HTML"; in Lesson 6, we clear up the mysteries of HTML. At this point, all you need to know is that Web pages are simply files that the Web supports, just like document (.doc) files are files that Microsoft Word supports.

Because Web pages are files, you don't have to stretch your imagination too far to realize that creating a Web page is simply the act of creating a specific type of file on your computer. Word documents, spreadsheets, databases, Web pages—they're all types of files. Clearly, you can see that Web pages aren't mysterious entities. They can't overwhelm you—they're computer files, and you've worked with computer files numerous times.

> **Lingo:** A *Web site* is a collection of related Web pages, usually including a home page and related subpages.

So, don't let Web pages intimidate you. Of course, this isn't to say that Web pages don't have a few idiosyncrasies that set them apart from other files. Namely, Web pages almost always incorporate multiple files and hyperlinks, and they are frequently rounded up into groups called *Web sites*.

The Multifile Nature of Web Pages

Granted, we just said Web pages are simply files, and we stand by that. But we should clarify a bit regarding the kinds of files we're referring to. While you read the next couple paragraphs, you might think we're providing a little too much information at this point—but we're really not. You should have at least an inkling (not necessarily a firm grasp, just yet) of Web page components and interactions before we get too far along. Enough of the disclaimer; on to the information.

First, at the most basic level, every Web page is a *text document*. A text document is a file that contains words, letters, and numbers without any formatting. For instance, opening Notepad or WordPad in Microsoft Windows (click Start, point to Programs or All Programs, click Accessories, and then select Notepad or WordPad) and typing your name, a catchy phrase, miscellaneous letters, a few numbers, or anything, really, creates a text document—not a Web page, mind you, just a text document. Figure 1-1 shows a simple example of a text document open in Notepad.
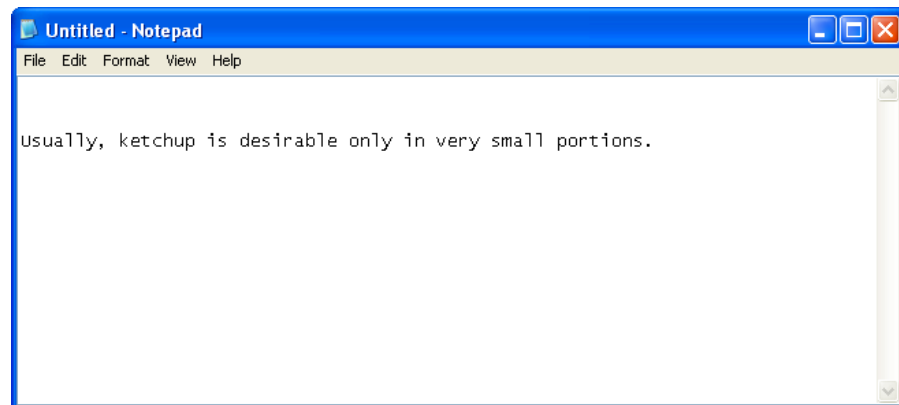
> **Lingo:** A *text document* is a file that contains words, letters, and numbers without any formatting.

Figure 1-1.
A text document contains just that—text!

To upgrade your text document to a potential Web page, you simply add specific HTML commands, as shown in Figure 1-2.
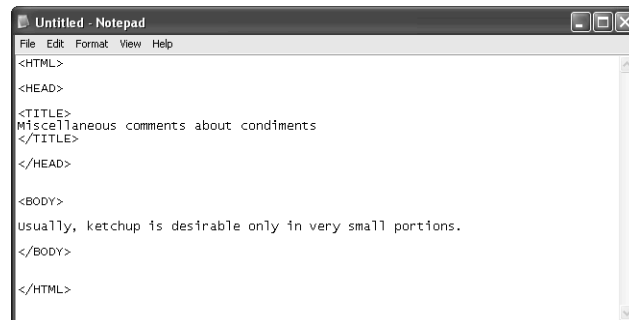

Figure 1-2.
This text document contains fundamental HTML commands along with title text and one line of body text.

Web Pages and Browsers

To view Web pages, you use a browser (such as Microsoft Internet Explorer). In most cases, a browser application resides on the local computer (the computer you're working on). You can delete, install, upgrade, and customize your browser just as you delete, install, upgrade, and customize other software applications on your computer (including Microsoft Office programs, such as Word and Microsoft Excel). One slight confusion occasionally crops up regarding where the Internet ends and your computer begins. Clarifying comes easily—when you view a Web page in your browser, the toolbars, menu bars, status bars, and so forth surrounding a Web page are part of the browser application, which

resides on your computer; the content within the browser's main window reflects the Internet content.

After you add HTML commands, you save the text document with an .html or .htm extension in place of .txt or .doc. (Don't sweat the specifics at this point.) Then, you can open the document in a browser application, such as Internet Explorer.

Figure 1-3 shows how the text document with the HTML commands shown in Figure 1-2 appears in a browser. Notice that only the body text and the title bar text, and not the HTML commands, appear in Figure 1-3. Just the body text and title bar text show because HTML commands merely provide instructions to browsers regarding *how* to display information, not *what* to display.
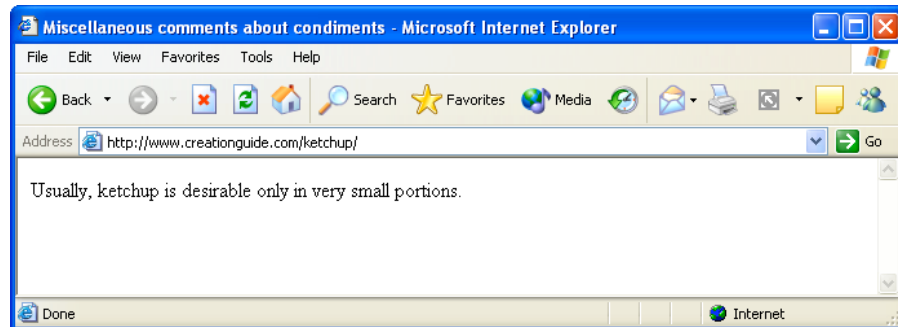

Figure 1-3.
You can view a properly configured basic text document containing HTML commands in a browser.

Don't worry if this HTML explanation seems a little vague at the moment. We walk you through the process of creating a Web site using HTML in Notepad or WordPad later in Lesson 6. You'll see then that HTML is fairly clear-cut if you take it one step at a time. (And if you want some added inspiration, in other lessons you'll find that you can create Web pages without knowing HTML at all.) At this point, you mainly need to recognize the following basic premise: *Web pages are text documents*.

You might've noticed that a paradox seems to be emerging here because we've adamantly stated that Web pages are text documents. But, if Web pages are text documents, why does the Web overflow with graphics? Fortunately, you can use HTML text documents in conjunction with specific graphics file types on the Web. (Namely, the Web supports graphics files with .gif, .jpeg, and .png extensions—but let's save the graphics file format discussion for Lesson 3.)

Here's the scoop. To show a graphic on a Web page, an HTML (text) document includes commands that tell a browser where to find a particular graphic and how to display it on the page (including position, size, and so forth). Thus, the multifile nature of Web pages is unveiled. Generally, when you look at a Web page online, you're looking at a few files—an HTML (text) file and some graphics files.

Try This!

You can see for yourself how HTML works. First, ensure that Windows is configured to show file extensions:

1   Open your Control Panel (in Windows XP, choose Control Panel from the Start menu; in earlier versions of Windows, click Start, point to Settings, and then click Control Panel).

2   In the Control Panel, double-click Folder Options, and then click the View tab.

3   Clear the Hide File Extensions For Known File Types check box, and then click OK.

After you've configured Windows to display file extensions, type the HTML text shown in Figure 1-2 into a Notepad document. Save the Notepad document to your desktop (so you can easily delete it later) as a text (.txt) file by selecting Text Document from the Save As Type box, and close Notepad.

Next, display your desktop, right-click the text file you just created, and select Rename. Replace the .txt extension with an .html extension. When Windows displays a message box asking if you're sure you want to change the file type (and warning you of potential "dangers"), click Yes—you're not wreaking any sort of havoc in this instance.

Now, you're ready to view the document in your browser. To do so, you can:

–   Double-click the HTML file you just created.

–   Open your browser, and drag the HTML file's icon into the browser window.

–   Open your browser, and type the path to the HTML file in the browser's Address bar.

Keep in mind that after you change a TXT file to an HTML file, you'll need to open the document from within Notepad if you want to edit the file's text.

To illustrate the multifile concept, take a look at a past version of the Arizona Film Society's home page shown in Figure 1-4. As you can see, the Arizona Film Society's home page consists of three files—an HTML document (index.html) and two graphics files (afs_title.gif and 4members.jpg). Figure 1-5 depicts a Windows folder view of the files used to create the home page illustrated in Figure 1-4. (Notice that the Windows folder contains the same HTML file and graphics files.)

Note

As you probably know, one of the Web's major draws is its dynamic nature. Many Web pages are frequently updated and modified. To aid our discussion, we've frozen a copy of one of the Arizona Film Society's past home pages at *www.creationguide.com/afs*. To see Web flux in action, visit the Arizona Film Society's current home page at *www.azfilmsociety.com* and notice that the page has been modified. (In fact, the page has been modified a number of times since we froze the sample home page for this lesson.)

A Little More HTML

The text and HTML commands used to create a Web page are collectively called the Web page's *source code*. (*Source code* refers to the text and HTML commands used to create a Web page.) Most browsers enable you to display a Web page's source code. For example, to display source code using Internet Explorer, you choose Source from the View menu, as shown here:
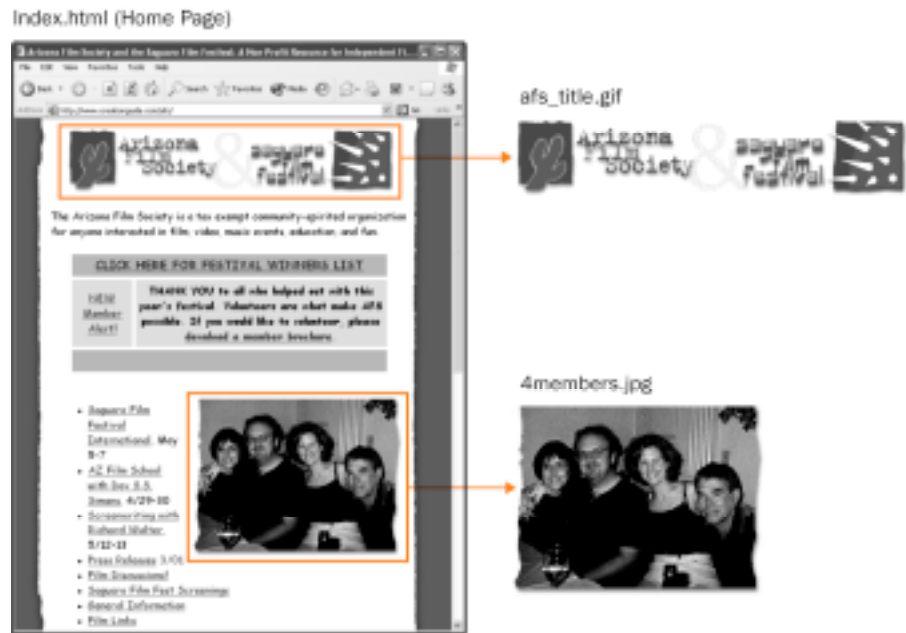
Index.html (Home Page)
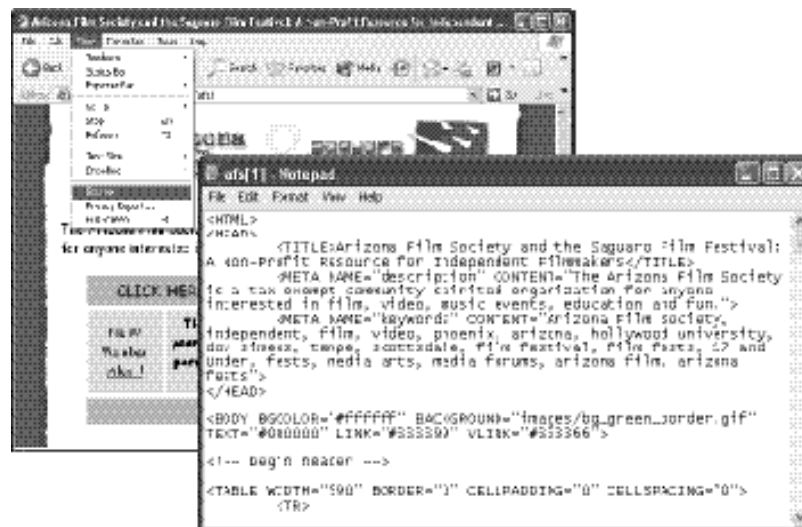
Figure 1-4.
An HTML text file and two graphics files combining to create the Arizona Film Society's home page (www.creationguide.com/afs).
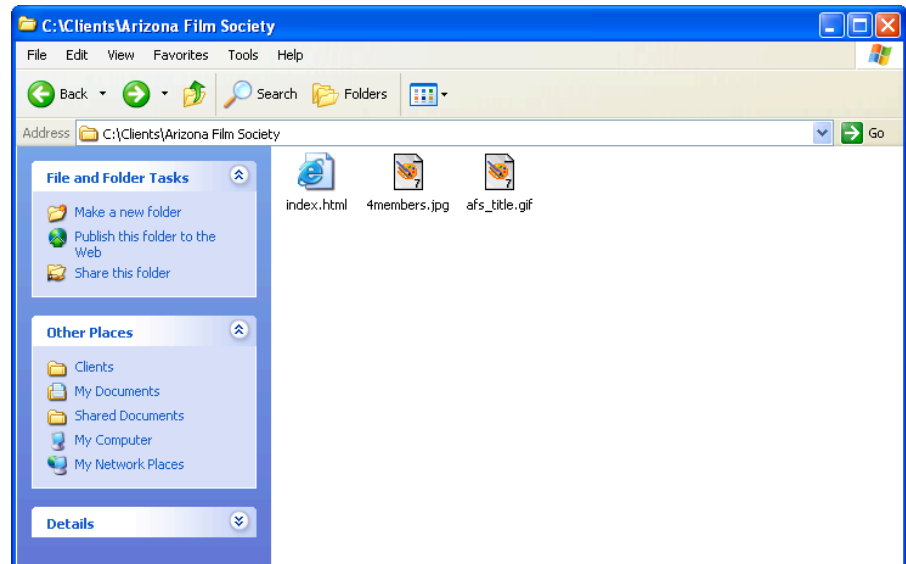

Figure 1-5.
The folder view of the Arizona Film Society's home page shows that three files combine to display the page online.

After reviewing Figures 1-4 and 1-5, you're ready for another "bottom line" blanket state-ment. Basically, you need to walk away from this discussion with the following informa-tion: *When you view a Web page in your Internet browser, you're usually viewing a number of files working together to create a single page.*
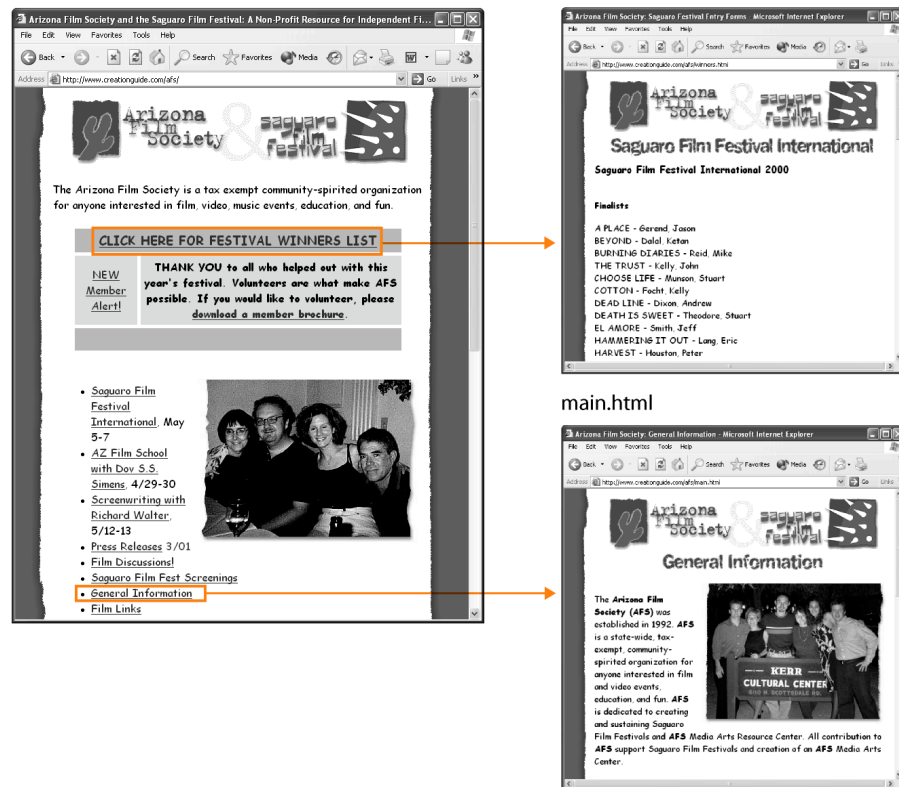
Having safely tucked away the knowledge that a Web page consists of multiple files, you should now consciously consider that a Web page isn't a solo form of communiqué, like a flyer on your windshield. Instead, a Web page almost always uses hyperlinks to link to other Web pages.

Hyperlinks and Web sites

Lingo: *Hyperlinks* are click-able text or graphics that en-able you to access additional Internet resources, such as another location on the current Web page, another Web page, or a file for downloading.

As we stated at the beginning of this lesson, we assume that if you want to create a Web page, you've surfed the Web. Thus, you've most likely clicked numerous *hyperlinks*. As you probably know, hyperlinks are clickable text or graphics that enable you to access addi-tional Internet resources and Web pages. More technically speaking, hyperlinks are ele-ments included in HTML documents that point to other Web pages or Internet documents (similar to how some HTML commands point to graphics files) or other areas on the same page. Figure 1-6 shows how a couple hyperlinks on the Arizona Film Society's home page point to other Web pages. Clicking a hyperlink displays another area on the current page or another Web page—which can be any page on the Internet (not just a Web page you've created), located anywhere in the world.

As a Web page developer, using hyperlinks naturally progresses to using multiple Web pages. Generally speaking, you usually won't want to place all your information on one big, long home page. Instead, you'll probably want to create a series of smaller Web pages that relate and link to one another. This collection of related pages forms your Web site.

Figure 1-6.
Hyperlinks take viewers to other Web pages, other areas on the same page, or other Inter-net resources.

From Your Head to the Web (and Back Again)

At this point in the lesson, the components are laid on the table: the Internet, the Web, browsers, Web pages, hyperlinks, and Web sites. This roll call of components is a good start, but we face the small detail of how a text file and a few graphics files that you've created on your computer are turned into a Web page on the Internet. Before we wade too deeply into the muck and mire of Web page transmissions, let's debunk a surprisingly popular myth: *People who view your Web pages have access to your desktop computer.* The preceding statement is *not* true! Rest assured, Web pages are not stored on personal computers. Instead, Web page files are stored on *servers*.

The Client/Server Nature of the Web

Servers are simply powerful computers that store Internet files and run special software designed to respond to *client requests*. Of course, now we've introduced the term *client*. Let's stop this circuitous approach and briefly indulge in some geekspeak.

Basically, Web files are transmitted using what is known as the *client/server model*. In the client/server model, one system (a server) connected to a network serves the request of another system (the client). For the purposes of Web design, a *client* is a fancy name for a browser (such as Internet Explorer) running on a user's computer, and a *server* is the combination of a powerful computer that stores Web pages and the software that responds to requests to display Web pages stored on that powerful computer. Therefore, when you access a Web page, the following process takes place:

1 You connect your computer to the Internet and open your browser. Then you enter a Web address (URL) in the Address bar and press Enter, or you click a hyperlink on your browser's start page.

2 The client (your browser) sends the typed URL or the URL associated with a hyperlink across phone lines, cables, and maybe routers to your Internet service provider (ISP). Your ISP is the company you pay to provide you with access to the Internet.

3 Your ISP then sends your URL request across the Internet through more cables, routers, and other high-speed data circuits to the system (the server) maintaining the requested Web page.

4 The server sends the Web page information across the Internet to your ISP, and, finally, your ISP forwards the information to your computer.

> **Lingo:** *URL* (pronounced "you-are-ell") stands for *Uniform Resource Locator*. A URL refers to an Internet address that tells your Web browser where to look on the Internet to find a specific Web page.

From a Web page developer's perspective, after you create a Web page, you copy your Web page's files to a server that will be hosting your Web page—similar to how you can copy a file from your hard disk onto a floppy disk (except that you copy your Web page's files across Internet lines, as described later in this book in Lesson 8). Using current File Transfer Protocol (FTP) applications, Web Folders (and My Network Places), or Web publishing wizards, the process of copying your Web page files to a server can be as simple as dragging files from your local folder into a folder on the server you're using to host your Web site. Therefore, when others view your published Web page, they access the server that stores copies of your files, not your computer.

> Please keep in mind that this lesson presents a simplified (albeit accurate) explanation of the basic Web page retrieval process.

That's a wrap on our fundamentals review. At this point, you're ready to forge ahead with the design and implementation of your Web pages, as described in the upcoming pages of this book. But before closing this lesson, we'd like to brief you on what's coming in the next few lessons.

Progressing at a Steady Clip

As you might suspect, much of the work of creating a Web page entails planning your Web page (and Web site) before you sit down at your computer. You need to spend at least a little time thinking about content—including text and graphics—as well as devising your page's layout. Although designing Web pages is a creative process, it's not a black art devoid of structure. In the course of this book, we pass along a few basic tenets that will help make the process of creating your Web page easier. Our expertise comes not only from our own years of online experience but also from numerous usability studies that many other designers, engineers, and information specialists have performed. From these sources, we have drawn some basic conclusions about text, graphics, and colors on the Web that we have proven in practice. Therefore, Lessons 2 through 5 address the information you should know about Web page design, including issues surrounding text, graphics, colors, helpful software programs, and Web page planning. You'll find the next few lessons packed with pertinent Web page creation information that will make your future Web design endeavors more successful. Therefore, we highly recommend that you read (or at least scan) Lessons 2-5 before diving into Lessons 6-8.

Finally, regardless of how you wind your way through this book and onto the Web, when all's said and done, remember to review Lesson 8. Lesson 8 describes how to go "live" (if you're using any method of Web page publishing other than MSN or another free hosting service).

All in all, by the time you complete these lessons, you'll have mastered the basics of creating Web pages in a number of ways. You'll no longer cringe when you see expressions like *HTML* and *domain name*, and your skills will serve as a strong foundation that you can build on to create a wide variety of more advanced Web pages.

Key Points

- The Internet is hardware.

- The Web is software (including programs and documents).

- Browsers are applications that enable you to view Web pages.

- Most basic Web pages consist of multiple files—an HTML (text) file and graphics files.

- A Web site is a group of related Web pages.

- Hyperlinks provide access to other Web pages, other locations on the same page, and other Internet resources.

- The Internet uses the client/server model, in which a server responds to client requests for information.

- Internet users access Web pages that are stored on servers.

- If you can use a computer, you can create a Web page!