Demystifying Basic HTML

<div>

After completing this lesson, you will be able to:

- *Understand the basics of HTML coding.*
- *Use HTML tags.*
- *Plan an HTML site.*
- *Create a table with HTML.*
- *Assemble a home page and subpages using HTML.*

</div>

Can you remember the intense and conscious effort it took to learn the sounds and shapes of the alphabet way back when? Eventually, after lots of practice, you got it. After you learned about letters, you slowly began to understand how to combine the letters into words, words into sentences, sentences into paragraphs, and so on. At this point, reading probably seems natural to you. Most likely, you read the newspaper and surf the Web without thinking much about individual letter sounds and shapes. That's because over the years, you've developed your foundation in letters and words into a seemingly innate ability to read. Learning how to create HTML documents from scratch—by using tags and understanding how HTML works—is a lot like learning to read. The process might take some patience and lots of practice at the beginning, but if you take the time to learn about HTML now, you'll eventually be able to use HTML to create Web pages almost as naturally as you read.

Gathering Project "Supplies"

To create the Web pages described in this lesson, you'll need the following "supplies":

- A basic text editor, such as Microsoft Notepad or WordPad (applications that are included with Microsoft Windows) or TextEdit (which comes with Macintosh OS X)

- A browser.

- 27 practice files, all of which can be found in the Lesson06 folder in the Web Design Fundamentals Practice folder on your hard drive. You use these files to create the Web pages in this lesson, listed on the following page..

b_background.gif	b_lessons2.gif	picture.gif
b_background2.gif	b_performances.gif	sendnote.gif
b_competitions.gif	b_performances2.gif	t_background.gif
b_competitions2.gif	b_recitals.gif	t_competitions.gif
b_contact.gif	b_recitals2.gif	t_contact.gif
b_contact2.gif	bg.gif	t_home.gif
b_home.gif	footer.gif	t_lessons.gif
b_home2.gif	logo.gif	t_performances.gif
b_lessons.gif	p_chris.jpg	t_recitals.gif

For detailed steps on setting up your practice files, see the section "Getting Your Folders and Graphics in Place" later in this lesson.

Why HTML?

In Lessons 1 through 5 we briefly introduce you to HTML. Basically, we explain that you construct Web page documents by including HTML commands (also called HTML tags) within the body of a basic text document. Although other technologies (such as XML and CSS, also known as Extensible Markup Language and Cascading Style Sheets) are beginning to have a significant impact on Web page development, HTML is the foundation of most Web pages today—and if you're going to create Web pages, you need to know about HTML.

Now we understand that you might not feel ready to create a Web page from scratch using only a blank document and a vaguely familiar-sounding technology named HTML. But believe it or not, you're ready—so for now, just go for it. We're having you use HTML in the first major Web page project in this course because you'll find that understanding HTML's basics will come in extremely handy whenever you create Web pages in the future—no matter how you create those pages. Therefore, we're taking a baptism-by-fire approach in this lesson, which means that you'll be writing your own HTML documents before long.

In some ways, you might find that the project in this lesson is the most important project in this half of the course. If you're going to create Web pages, you'll be well ahead of the game if you master some basic HTML commands and concepts. Someday, when you're much more comfortable with Web page creation, knowing at least some HTML will enable you to modify and tweak pages to suit your preferences, even if the pages you're modifying have been generated by an HTML editor. Furthermore, knowing HTML means that you'll be able to remove unnecessary (and sometimes proprietary) HTML commands that many HTML editors tend to add to Web page documents. Removing unnecessary codes can make your HTML documents smaller, which in turn means that your pages will load faster. Finally, as you become more proficient using HTML commands, you might find that you can make changes more quickly and precisely by adding, deleting, or modifying HTML code instead of modifying a Web page in an HTML editor.

We're now ready to get started. The first order of business, before we commence creating a Web site, is to briefly (very briefly) go over some basic HTML theory. By the way, when we say theory, we're talking clear, helpful information—not complex rhetoric. Think of the upcoming theory discussion as spreading a blanket before picnicking—you might as well get somewhat comfortable and discourage at least a few of the pests up front.

So You've Heard about Cascading Style Sheets

Cascading style sheets (or CSS) is a technology that enables you to define a Web site's page formatting in a single place—either in a separate document or in a special area within each Web page document. In many Web designers' and developers' eyes, everyone should be using CSS as soon as possible. But the reality of the situation is that the Web is a ways off from being fully CSS-compliant, and many Web sites don't incorporate style sheets. Fortunately, the Web continues to support older formatting techniques along with the newer CSS commands.

In relation to you, the goal of this lesson is to teach you HTML; therefore, we decided to omit teaching you how to use style sheets in this lesson. After you learn HTML, you'll be able to move on and learn CSS at your leisure (who knows—maybe we'll write a book about CSS to help you along!). To help whet your CSS appetite, we show you how to modify the autogenerated style sheets in the project in Lesson 7.

In relation to CSS, you might hear that some of the HTML tags and attributes we show you in this lesson are *deprecated*, such as the <FONT> tag. A deprecated element can be (and is) used on the Web; it is simply earmarked for eventual elimination. You can see a list of HTML tags along with notations regarding which are classified as deprecated at *www.w3.org/ TR/1999 /REC-html401-19991224/index/elements.html*.

Finally, due to the various levels of CSS and HTML compliance on the Web these days, you can help browsers interpret your Web pages by specifying whether your page is strictly compliant with the latest standards, transitional (includes deprecated HTML elements), or framed (includes deprecated HTML elements and frames within the Web site). To do this, you insert a particular version of the <!DOCTYPE...> HTML tag in your Web pages, as described here as well as in this lesson's project steps:

- If your Web page adheres to CSS and HTML 4.01 standards, insert:
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

- If your Web page adheres to HTML 4.01 standards and includes deprecated HTML elements and attributes (most of which concern visual presentation), insert:
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://
www.w3.org/TR/html4/loose.dtd">
```

- If your Web page adheres to HTML 4.01, includes deprecated HTML elements and attributes, plus uses frames, insert:
```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frame-
set//EN"
"http://www.w3.org/TR/html4/frameset.dtd">
```

Because the project in this lesson includes deprecated HTML elements but not frames, your documents will use the second statement—the transitional statement—in the preceding list. We'll show you how to add this statement to your HTML document in the project steps later in this lesson.

HTML Basics

Fundamentally, HTML commands serve as instructions that tell a browser how to display a Web page's content. In other words, HTML commands provide format information that controls the display of your Web page's text and graphics. Keep the purpose of HTML commands in mind. You'll see later how HTML commands weave their way in and around your Web page's content in an HTML document, but, basically, an HTML document contains two types of information:

    –   Content information, including text and pointers to graphics

    –   HTML commands, which are used to manipulate how content displays

In this lesson, we show you how to enter HTML commands and page contents into a plain-text document to create Web pages. Furthermore, you'll link the pages you create so that they can work together to create a Web site. To accomplish this feat, you'll need to use a text editor, such as those mentioned in Lesson 4, including Notepad or WordPad (if you're running Windows) or Text Edit (if you're using a Mac OS X, or SimpleText if you're using Mac 9x or earlier operating systems). Figure 6-2 shows how fully coded HTML documents appear in TextEdit, WordPad, and Notepad, respectively. When you start this lesson's project, you'll start with a blank page. To open Notepad, click Start, point to Programs, point to Accessories, and then click Notepad. To open WordPad, click Start, point to Programs, point to Accessories, and then click WordPad. To open TextEdit, double-click the TextEdit icon on your hard disk.

Note

Don't be alarmed at the seemingly incomprehensible conglomeration of HTML commands shown in Figure 6-2. HTML can look complex, but it really consists only of combinations of letters, numbers, and symbols with a little organization thrown in. You're obviously familiar with letters, numbers, and symbols, so rest assured that learning to use HTML commands is well within your skill set.

As we mentioned, HTML tags take care of formatting your page. In contrast, your content is the information that displays on your page (text, graphics, headings, and so forth). In other words, HTML takes care of how information displays (bold, italic, left-aligned, and so on), and content specifies what is displayed. Knowing how to incorporate the proper HTML tags throughout a Web page's content is the key to making a Web page in a text editor. So let's look at how to use HTML tags.
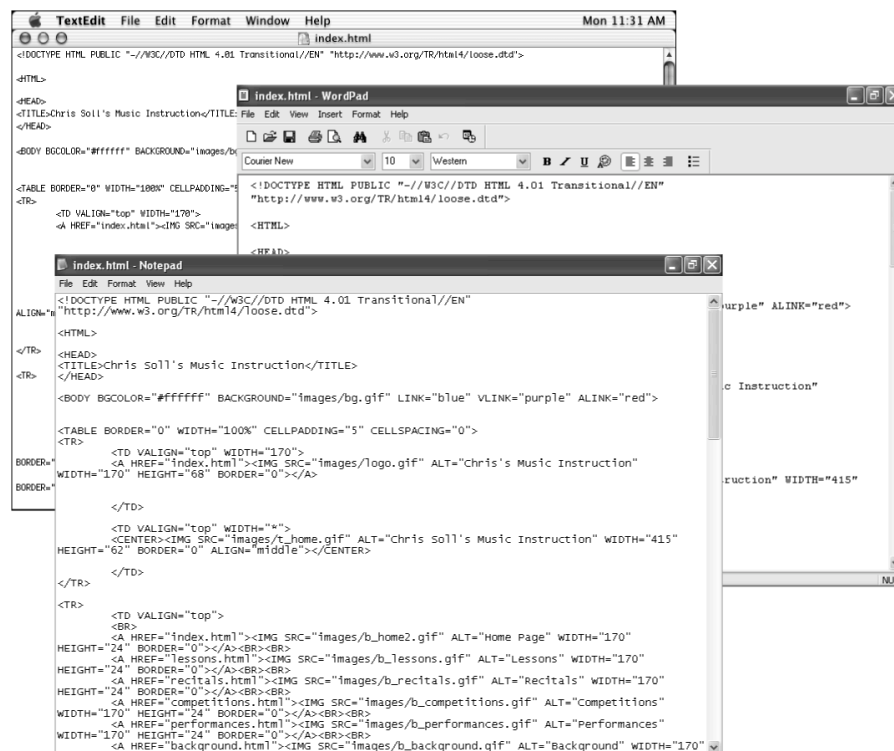

Figure 6-2.
Viewing an HTML document in TextEdit, WordPad, and Notepad.

Using HTML Tags

In this section, we introduce the basic rules of HTML along with a few common tags. Keep in mind that this section does not define every HTML tag out there; quite a few HTML tags exist, and plenty of books devoted to HTML provide comprehensive command lists. (If you want to find out more about HTML than what we cover here, check out any of our favorite HTML references, which are listed in the section "Additional Resources" near the end of this lesson.) Our philosophy is that if you learn the basic rules of using HTML tags, you'll be able to use any of the tags you discover online or in HTML books.

Let's start our discussion of HTML tags with a simple rule: *HTML tags consist of commands that appear within angle brackets (<>).*

> **Lingo:** *Source code* refers to the contents of the HTML document that creates a Web page. Most browsers enable you to view a Web page's source code. For example, to display a Web page's source code in Microsoft Internet Explorer, display a Web page, click the View menu, and click Source.

For example, one of the first tags in a Web page's source code is typically <HTML>. This tag tells a browser right off the bat that the text document is an HTML document. The browser knows that any text within angle brackets (<>) is an HTML command that needs processing and that all text outside angle brackets (<>) is content that needs to be displayed.

Here's the second rule you need to remember: *HTML tags are not case-sensitive*.

This rule isn't earth shattering, but it's convenient to know. It means that browsers don't care whether the text between the angle brackets is capitalized or lowercased. Therefore, <B> and <b> are essentially the same tag (which, incidentally, is a tag that instructs the browser to display the text following the tag in boldface). Similarly, <HTML>, <html>, <HtMl>, and any other combination of capital and lowercase letters represents the same tag.

Here's rule number three: *HTML tags almost always come in pairs*.

Because most HTML tags are used primarily for formatting purposes, HTML tags often come in twos: a starting tag and an ending tag (also referred to as an *opening tag* and a *closing tag*). This pairing enables you to tell browsers where a particular formatting attribute (such as boldfacing) should start and where it should end. Think about when you go to the movies with a few friends and two friends go in to save seats while the rest of the group goes to the concessions counter. The two people saving the seats sit separately at each end of the saved seats to mark a span of seats that will contain the friends. If the seat-savers were HTML tags, they'd tell the browser that all the seats between them should be formatted as their friends' seats.

Starting tags and ending tags have very specific purposes—namely, a starting tag indicates when an action should start, and an ending tag indicates when an action should stop. (See, we're not talking rocket science here!) While starting and ending tags appear very similar they have a minor, albeit critical, difference. Ending tags are differentiated from starting tags by the inclusion of a forward slash (/) just after the left bracket, like this: </HTML>. The last element in HTML documents is usually the </ HTML> command, which indicates the end of the Web page's display. Going back to the movie theater example, let's say that one seat-saver is sitting in an aisle seat and the other seat-saver is sitting in the middle of the row. The seat-saver sitting in the middle of the row is wearing a red shirt. Suddenly, a new arrival asks the seat-saver sitting in the aisle seat whether the seats are taken. The aisle-side seat-saver would say something like, "Yes—all the seats down to the person in the red shirt are taken." That's the role of a starting tag. The red-shirted seat-saver serves as an ending tag. For example, a <B> tag tells a browser, "Please boldface all the text between me and that </B> tag over there."

For further illustration, let's look at an example of text that uses HTML tag pairs. The following sentence includes HTML starting and ending tags that format the sentence as a paragraph (<P></P>), display the phrase *butter flavoring* in italic (<I></I>), and format the word *popcorn* in boldface (<B></B>), as shown in Figure 6-3:

```
<P>Do you want <I>butter flavoring</I> on your
<B>popcorn</B> or do you like it plain?</P>
```
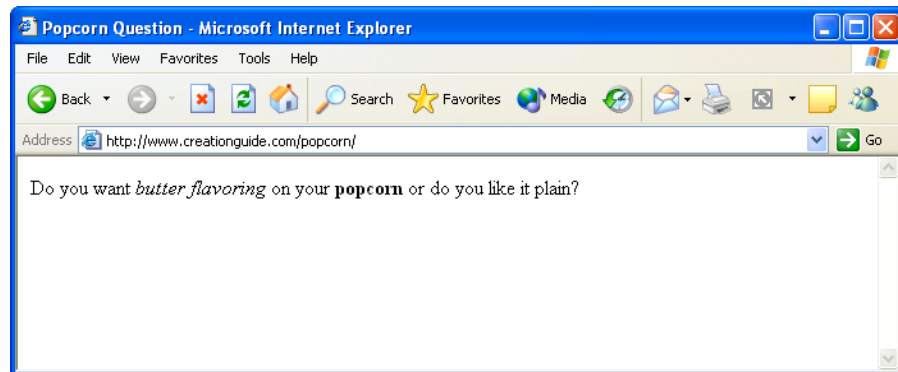

Figure 6-3.
If the popcorn question is included in an HTML document, you could view the formatted

question in a browser, as shown here .

The popcorn sentence also illustrates an interesting concept called *nesting*. In HTML documents, nesting has nothing to do with twigs and feathers and everything to do with the order in which HTML tags appear. In the popcorn sentence, the italic tag set (<I></I>) and the boldface tag set (<B></B>) are nested within the paragraph tag set (<P></P>). Here's a key rule you should follow when you're nesting HTML tags: *Nested HTML tags should close in the reverse order in which they open.*

That rule might seem a little confusing, so let's look at an example. Basically, opening and closing HTML tags shouldn't get their lines crossed. Here's a correct pattern:

```
<HTML>  <P>  <B>  </B>  </P>  </HTML>
```

In this example, the <B> (boldface) tags are nested within the <P> (paragraph) tags, which are nested within the <HTML> (document identifier) tags. This setup would re-sult in bold text within a paragraph within an HTML document. The following setup would also work:

```
<HTML>  <P>  <I>  </I>  <B>  </B>  </P>  </HTML>
```

Notice that this nesting example uses the same pattern as the popcorn sentence. In this ex-ample, the italic tag set and the boldface tag set aren't nested inside each other, but both tag sets are nested within the paragraph tag set.

Now let's lighten up the discussion a bit and look at a more clear-cut rule: *By default, HTML documents display a single space between text elements.*

This rule might seem odd to mention, but spacing issues are a great concern on the Web for a number of reasons (mostly because designers have to deal with content that resizes and reflows—issues that are nonexistent in printed documents). In an HTML document, adding any number of spaces within your code by using the spacebar, Tab key, or Enter key results in a single space. Therefore, you could embed the following four code snippets in an HTML document:

```
<I>Music Instruction</I>
<I>Music                        Instruction</I>
<I>          Music Instruction          </I>
<I>
Music Instruction
</I>
```
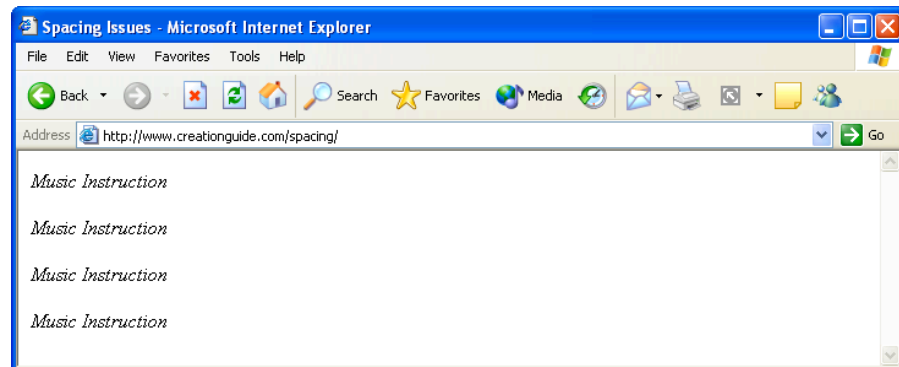
and the text will appear as shown in Figure 6-4.

Figure 6-4.
Text displays with a single space between words, even when extra space is added between words in the HTML document.

Now you're ready for the next rule, which adds some spice to HTML tags: *Some opening HTML tags can contain properties (also called attributes), which further refine an HTML tag's instructions*.

In other words, you can frequently customize the instructions related to an HTML formatting command. For example, you can add a COLOR attribute to the <FONT> command to change the display color of text, like this:

```
They say the <FONT COLOR="green">grass</FONT> is greener.
```

If you inserted the preceding sentence into an HTML document, the text would display the word *grass* in green.

Finally, here's the last rule in this section: *Numerous variations exist when it comes to the HTML nesting theme, properties, and use of tag sets*.

As with all rules, you'll find that although most of HTML is predictable, the technology is as consistent as spelling rules, which means that you'll frequently find exceptions to the rules. For example, if you want to add a line break in HTML, you enter <BR>. There's no closing tag for a line break—you either have a line break or you don't. Similarly, you insert a horizontal rule with the <HR> tag; again, no closing tag required.

Don't worry if you're feeling slightly confused. You'll start to get a feel for HTML as you work on the Web site project in this lesson. There's nothing like hands-on experience to gain knowledge. We'll introduce you to additional HTML tags and concepts in the project as we go. For added assistance, you might want to keep Table 6-1 handy while you work.

Table 6-1. HTML Tags Used in the HTML Project

Tags	*Function*
`<A HREF="xxx.xxx"></A>`	Marks the *anchor*, or clickable, portion of a hyperlink. The HREF attribute points to the information that should be displayed after the anchor's content is clicked. Anchor content is specified between the anchor tags (<A></A>) and can include text and images.
`<B></B>`	Indicates to display text between the <B> and </B> tags in boldface.
`<BLOCKQUOTE></BLOCKQUOTE>`	Offsets a paragraph from the regular body text, usually by indenting the paragraph's left and right margins.
`<BODY></BODY>`	Marks the start and end of the Web page's displayable content.
` `	Inserts a line break. The tag doesn't have a closing tag, and this tag is frequently used consecutively to create white space on a Web page.
`<CENTER></CENTER>`	Centers the enclosed information on the page or within a table cell.
`<!DOCTYPE...>`	Specifies the Web page's document type definition (DTD), such as whether the page uses strict HTML and CSS coding, transitional HTML coding (including deprecated HTML tags), or frames.
`<FONT></FONT>`	Enables you to specify the enclosed text's font color, face, and size.
`<H1></H1>`	Specifies heading text. Heading sizes range from H1 through H6, with H1 being the largest heading size.
`<HEAD></HEAD>`	Provides an area in which you can display your Web page's title, include search engine information, add advanced formatting information, embed CSS coding or link to a style sheet, and write scripts. Other than the text within the embedded <TITLE></TITLE> tags, most head information doesn't display directly to viewers.
`<HTML></HTML>`	Delineates the start and end of an HTML document.
`<I></I>`	Indicates to italicize the text appearing between the <I> and </I> tags.
`<IMG SRC="xxx.xxx">`	Displays an image on a Web page. The SRC attribute points to the particular image that should be displayed.

`<LI></LI>`	Identifies a list item within an unnumbered (bulleted) list <UL> or an ordered (numbered) list <OL>.
`<OL></OL>`	Specifies an ordered (numbered) list.
`<P></P>`	Indicates the start and end of a paragraph. By default, paragraphs display left-aligned. The closing </P> tag is optional. (In other words, you can simply insert the <P> tag at the start of each new paragraph to format your HTML contents without typing </P> at the end of each paragraph.) We've included the closing </P> tag throughout this lesson for added clarity. Browsers typically insert a blank line (plus a little extra space) before starting a paragraph.
`<TABLE></TABLE>`	Delineates the start and end of a table.
`<TD></TD>`	Defines the start and end of a cell within a table. <TD> tags are nested within <TR> tag sets.
`<TITLE></TITLE>`	Enables you to insert the Web page's title text that should display in the browser's title bar.

`<TR></TR>`	Indicates a table row. <TR> tags are nested within a <TABLE> tag set.
`<UL></UL>`	Specifies an unnumbered (bulleted) list.

Just as a last note in this section, we want to make a minor disclaimer. Although we're confident that you can create an HTML document from scratch, please keep in mind that this lesson serves only as an introduction to creating Web pages in HTML. Unfortunately, covering all the available HTML commands in a single lesson is unrealistic, but this lesson is packed with helpful coding tips and you'll find some leads on good HTML references in the "Additional Resources" section near the end of the lesson. If you create the Web site described in this lesson's project, you'll gain a strong foundation in HTML coding as well as have a template that you can customize to create unique Web pages. (We even tell you how to use the site as a template later in this lesson.)

Handling HTML Documents and Web Graphics

When you create Web pages, you usually work with multiple files. You'll have your home page HTML file (generally named index.html or index.htm), a graphics file for each graphical element on your page, and additional HTML files for linked pages. Therefore, before you start creating, you have to think of an organizational scheme so that you don't drive yourself crazy later. We highly recommend that you create a folder to contain all the HTML files used in your Web site, and, within the main folder, create a subfolder named images. Then you can store all your HTML documents in the main folder and place your graphics in the images folder. To illustrate, see Figure 6-5, which shows the HTML documents and images necessary to create this lesson's project site.

Keeping your files organized is imperative when you're adding graphics and creating hyperlinks because you must include instructions in your HTML document regarding where the browser should look for a particular graphic or linked page. Furthermore, being organized can greatly simplify the file uploading process when you're ready to go "live" by transferring your local files to a Web server. Your best bet is to create a folder that you can use consistently throughout the Web page and Web site creation process.

Along with being organized, you should religiously save and preview your Web pages throughout the development process.
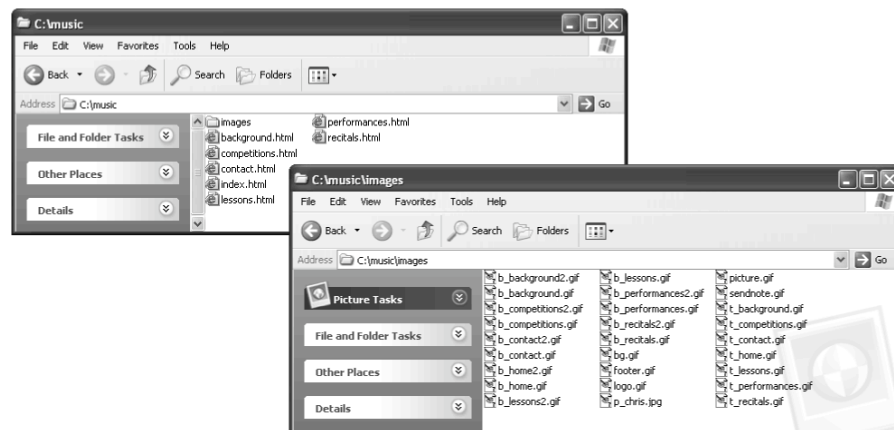

Figure 6-5.
You should keep your Web site's files and folders organized in a simple yet logical manner.

Saving and Previewing HTML Documents

When you create Web pages—especially when you're hand-coding HTML—you should save and preview with abandon. Speaking from firsthand "We can't believe we just lost all that data" experiences, we can recommend without reservation saving your work frequently—that pretty much goes without saying whenever you're working on any file on

"Save, save, save!" should be one of your mantras when you're working with computers.

any computer. (If monitors grew grass, our mice would wear out well-worn diagonal paths to the Save button—not to mention that we both knowingly contribute to the slow erosion of our *Ctrl* and *S* keys by pressing Ctrl+S every so often.)

In addition to frequently saving your files, you should preview the Web pages you build numerous times throughout the creation process. Previewing an HTML page simply means looking at your HTML document in a browser (or a couple browsers) as opposed to staring at the text and HTML code version of the document in a text editor. Performing this exercise, you can see how the HTML is formatting your content and you can troubleshoot display problems early. We'll often alter a site's layout simply because what looks good on paper doesn't transpose well to an online page.

To preview a Web page in your browser, use any of the following procedures after you've created an HTML file:

‒ Display the contents of the folder containing the HTML document, and double-click the HTML document's icon.

‒ Open your browser application (such as Internet Explorer), and type in the HTML file's location.

‒ Open your browser application, open the folder containing the HTML document, and drag the HTML file's icon from its folder into the browser window or the browser's Address bar.
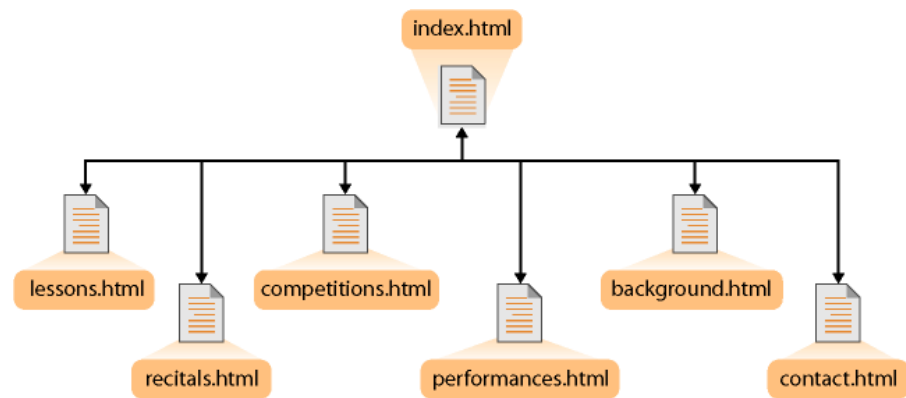

Note

We suggest specific points at which you should save and preview the project Web site in the next section. Feel free to save and preview your files more frequently than we suggest, however, especially if you take a break while creating.

We've covered a good bit of theory; now it's the witching hour. If you've read the previous few pages, you're ready to tackle the HTML Web page creation project. You should have a workable knowledge of basic HTML tags, realize that you should save your HTML documents and images in designated folders, and recognize the importance of frequently saving and previewing your Web pages throughout the creation process. We're satisfied that you're ready, so let's get the project rolling.
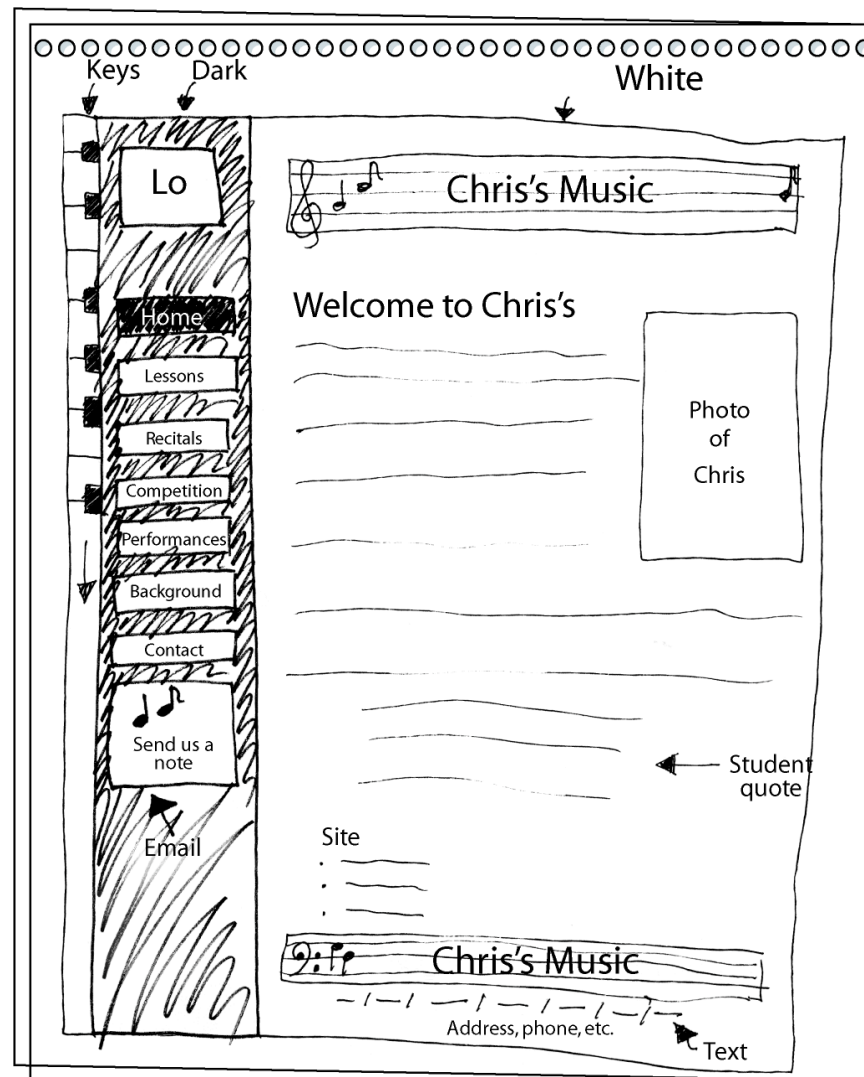
Planning the HTML Site

For the HTML project, we decided to create a Web page for Chris Soll, a professional musician and music instructor. Our first planning step involved meeting with Chris and finding out what types of information she wanted to include on her Web site. In our initial consultation, we found that she had a number of student-specific as well as professional-specific topics she wanted to incorporate into her site. Based on this information, we initially attempted to design a two-tier navigation bar, but the design started to look too cluttered. We determined that we could make a cleaner site by using specifically named buttons and providing a quick Site Overview section on the home page. The final design resulted in a clean, flexible layout.

After you create Chris Soll's Web site—which we quickly began to refer to as "the music site" during our consultation, based on the overwhelming presence of her baby grand piano, flute collection, and stacks of sheet music and music books in her studio—you can use your HTML document as a template to create similar Web sites that have a completely different look and feel. (Don't worry—we describe how to use the music site's code as a template later in this lesson.) Figure 6-6 shows the storyboard we came up with to illustrate the pages we wanted to include in the music site. Figure 6-7 shows a finalized sketch of the music site's home page.

Figure 6-6.
The site's structure enables access to any page in the site from every location.

Figure 6-7.
The sketch shows the basic layout for the music site's home page design.

Note

Notice in Figure 6-7 that we initially planned to insert the address information below the site's text links along the bottom of the page. Later, during the design process, we realized that placing the address above the links made better design sense. The links were already

listed in the navigation bar on the left, so we wanted to present the "new" and more important address information before repeating the text link information.

Getting Your Folders and Graphics in Place

As we mention earlier in this lesson, your first task is to create a folder for your Web files and organize your graphics. Here's the process we suggest you follow (though feel free to change the folder location and name to suit your preferences):

1   Create a folder on your C:\ drive and name the folder *music*.

2   Open the music folder, and create a subfolder named *images*.

3   Locate the Lesson06 folder in the Web Design Fundamentals Practice folder on your hard drive.

4   Copy all of the files in the Lesson06 folder to the images subfolder that you created in step 2.

After you finish this lesson and are done experimenting with the music Web site, you can delete the C:\music folder if you like.

As you copy files from the Lesson06 folder to the images subfolder, notice the naming scheme we've used to label images:

—   **b_xxx** Specifies that the image is a button. A *b_* graphic appended with the number *2* (for example, *b_background2.gif*) indicates a second version of the button that displays whenever the associated page is displayed. (You'll see what we mean later.)

Therefore, b_background.gif is the Background button, and b_background2.gif is the "current page" version of the Background button, as shown in Figure 6-8.

Figure 6-8.
Each button in the music site consists of a regular button and a current page version of the button.

—   **bg.gif** Specifies the background graphic. The music site uses the same background graphic throughout the site, so only one bg.gif file is required.

—   **footer.gif** Identifies the graphic as a running footer graphic used at the bottom of the site's pages.

—   **logo.gif** Identifies the logo graphic. The music site uses the same logo graphic throughout the site, but it's possible that you'd have a few ver-

sions of a logo graphic (especially if you're using a smaller or modified version of the logo on subpages).

The picture.gif file is a placeholder graphic used on the template discussed later in the lesson. You won't use the picture.gif file in the music site.

– **p_*xxx*** Specifies that the graphic is a picture. The music site has only one photograph, and it's on the home page.

– **sendnote.gif** Identifies the graphic as the "send mail" icon. Later, you'll link this icon on the music site so that users can click the sendnote.gif graphic to open a preaddressed e-mail message when they want to send a message to Chris Soll.

– **t_*xxx*** Specifies that the image is a title bar banner graphic. For example, t_background.gif is the Background page's title bar banner graphic (which displays the word Background and is placed at the top of the page that presents background information about Chris Soll), as shown in Figure 6-9.
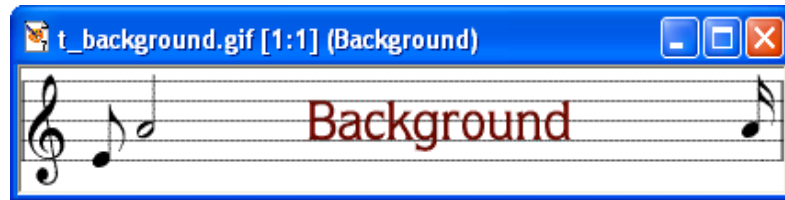
Figure 6-9.
A graphic file is used to create a title bar banner on each page in the music site.

When you create your own Web pages and Web graphics, you'll probably devise your own naming scheme. We've shown you the method we used to name our graphics to give you an idea of how helpful having a naming system can be. You'll see the benefit of a well-planned graphics-naming scheme as you start to insert HTML code in a few moments.

Preparing Your Home Page File

After you have your folders and graphic files in place, you're ready to begin creating your site's home page. To begin the creation process, you need to create an HTML document that contains the standard tags that appear in all HTML documents. The standard tags are (see Table 6-1 presented earlier in this lesson):

_   `<!DOCTYPE...>`

_   `<HTML></HTML>`

_   `<HEAD></HEAD>`

_   `<TITLE></TITLE>`

_   `<BODY></BODY>`

Notice that, with the exception of the <!DOCTYPE...> tag, the standard tags all come in sets. Figure 6-10 shows the proper way to nest the standard HTML tag sets in an HTML document and shows how to insert title text. After you type the standard HTML tags in a text document, you need to save the text document as an HTML document, as described in the upcoming procedure.
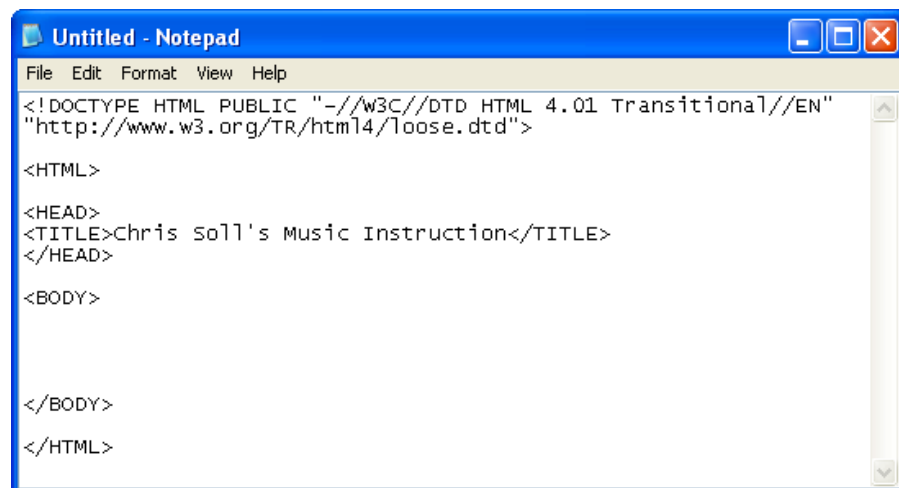

Figure 6-10.
This document contains the standard HTML tags with title text inserted between the

<TITLE> tags.

Note

Remember that HTML isn't case-sensitive, so you can type <HTML>, <html>, <HtMl>, or any other combination of capital and lowercase letters and your tag will be fine. In this lesson, we used uppercase letters for HTML tags throughout to make spotting them easy.

To begin creating the music site's home page and save it as an HTML file, follow these steps:

1   Open Notepad, WordPad, or TextEdit. We use Notepad throughout this project.

2   Click in a new blank document and type the following document type definition (which indicates that the document contains deprecated tags):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN" "http://
www.w3.org/TR/html4/loose.dtd">
```

3   Press Enter twice.

4   Type <HTML>, and then press Enter twice.

5   Type <HEAD> and press Enter.

6   Type <TITLE>Chris Soll's Music Instruction</ TITLE>, and then press Enter.

7   Type </HEAD> and press Enter twice.

8   Type <BODY>, press Enter a few times (to give you some breathing room when you enter your Web page's content information), type </BODY>, and press Enter twice.

Note

Content text—text that displays between and outside HTML tags—displays in the same case as it is entered in the text document. Thus, if you type *chris soll's* instead of *Chris Soll's* in step 4, Chris's name will appear in all lowercase letters on your Web page. Also, remember that you don't have to worry about putting too many spaces (including blank lines) between content text and HTML tags. When a browser displays the document, it ignores the extra spaces.

9   Type </HTML> to complete the standard HTML tag setup.

You're now ready to name and save the file to your C:\music folder.

10   On the File menu, click Save.

11   In the Save As dialog box, type **index.html** in the File Name box and ensure that Text Documents is selected in the Save As Type drop-down list. (If necessary, click the arrow to select the Text Documents option.)

12   Use the Save In drop-down list and navigate to the C:\music folder on your computer.

13   Click Save and close Notepad.

The file should now display as index.html in your C:\music folder, and the icon should indicate that the file is an HTML document. At this point, you can already view your file in your browser. To view your newly created HTML file, double-click index.html. The file

should open in your Web browser. Notice that the only content is the Web page's title text, which displays in your browser's title bar.

Now that the Web page's title and standard HTML tags are in place, let's add the page's background image and default link colors.

Specifying the Background and Link Colors

You can include attributes in the <BODY> tag to add background colors, background images, default text link colors, and so forth, as described in the following steps:

1   Open Notepad, WordPad, or TextEdit, and then open the C:\music\index.html file from within the text editor. If you don't see the file index.html listed in the Open dialog box, ensure that the Files Of Type list box displays All Files (*.*). If you're using Mac OS X, TextEdit automatically displays an HTML document as a Web page by default. To open and display the HTML code instead of the Web page, select the Ignore Rich Text Commands check box in the Open File dialog box (note that you can select this option in the program's Preferences dialog box as well, if you don't want to select the check box on a case-by-case basis).

Tip

You must open your HTML file by using the text editor's Open command, by dragging the document's icon into the text editor's window, or by right-clicking the HTML file and choosing Notepad on the Open With menu. If you double-click an HTML file in folder view, you'll display the HTML document in your Web browser.

---

2   In the <BODY> tag, click after the Y and before the >, press the spacebar, and then enter the following attributes and values, including the quotation marks:

```
BGCOLOR="#ffffff" BACKGROUND="images/bg.gif"
LINK="blue" VLINK="purple" ALINK="red"
```

The <BODY> attributes you just added are defined as follows:

‒   **BGCOLOR** defines a background color. Although the music site uses an image for the background, we defined a white background for folks who view the site with images turned off. In the color chart shown in palette216.gif, located in the Lesson03 folder for this course, you can see that #ffffff is a hexadecimal number that equates to white.

‒   **BACKGROUND** enables you to specify a graphics file to use as a background image. Remember that browsers automatically tile background images to fill the browser window. The music page uses the bg.gif file, which is stored in the images subfolder of the music folder, as a background image. Because both the index.html document and the images folder reside in the music folder, you don't have to indicate the image's complete address. If your image was saved elsewhere (that is, not within a subfolder of the folder that contains the index.html document), you'd have to enter the entire address that points to the image.

‒   LINK enables you to specify the color in which unvisited text hyperlinks display.

> **Lingo:** *Tiling* refers to repeating an image across a window's area and down until the entire window is filled with the repeating image.

‾ VLINK enables you to specify the color in which visited hyperlinks display. In other words, after a user visits a site's Contacts page, any text links pointing to the Contacts page will display in the visited link color (purple, in this case).

‾ ALINK enables you to specify the color in which links display while users click the links. Showing a different color while the users click links clearly indicates to users that they are activating a hyperlink.

Your HTML code should now display as shown in Figure 6-11.

```
index.html - Notepad
File  Edit  Format  View  Help
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<HTML>

<HEAD>
<TITLE>Chris Soll's Music Instruction</TITLE>
</HEAD>

<BODY BGCOLOR="#ffffff" BACKGROUND="images/bg.gif" LINK="blue" VLINK="purple" ALINK="red">

</BODY>

</HTML>
```

Figure 6-11.

Some page formatting attributes can be added within the <BODY> tag. (The newly added code is highlighted in orange.)

Tip

Whenever you enter HTML code, always verify that you've included all angle brackets (<>) and quotation marks ("") in your HTML code as well as spelled the HTML commands properly. Missing small elements or misspelling commands can cause your Web page to display incorrectly or not at all. We've included screen shots of the code you're creating throughout this lesson so that you can easily check your work.

If the changes you make in your HTML document don't show up on the preview page in your browser, click the Refresh button to update your view.

3 Save your HTML document, open your browser, and view index.html in your browser window. (You don't have to close your text document, but you do have to ensure that you've saved your most recent changes.) The index.html file should now display the background image in your browser window.

Now that the standard HTML tags, default background image, and link colors are in place, the next step is to begin to format your Web page's <BODY> area.

Creating a Table

In this section, we show you how to create a table that will contain all the elements of your Web page. Controlling elements on a Web page is a little tricky because of the variable nature of browsers and browser windows, so many sites are designed using tables with hidden borders to help lay out Web pages. Eventually, style sheets will take over the job of formatting and aligning Web pages, but for now thousands and thousands of Web pages rely on tables to control the display of Web page elements. Of course, not every Web page uses a table, but you'll find tables to be an extremely useful tool.

Basically, tables use three tags:

‾ <TABLE></TABLE> delineates the start and end of a table.

‾ <TR></TR> indicates a table row. <TR> tags are nested within a <TABLE> tag set.

—   <TD></TD> defines the start and end of a cell within a table. <TD> tags are nested within a <TR> tag set.

In the music site, you'll create a two-column, three-row table:

1   If necessary, open Notepad, WordPad, or TextEdit, and then open C:\music\index.html from within the text editor. If you don't see the index.html file, remember to ensure that the Files Of Type list box displays All Files (*.*).

2   In the index.html file, click below the opening <BODY> tag (the <BODY> tag should now contain the attributes you added in the previous section), type `<TABLE BORDER="1" WIDTH="100%" CELLPADDING="5" CELLSPACING="0">` and press Enter. Notice the quotation marks around the attribute's values. Make sure you include beginning and ending quotation marks throughout. Here's the purpose of each of the attributes included in the opening <TABLE> tag:

—   BORDER defines the width of the table's outline in pixels. For now, we're showing a 1-pixel border to aid in seeing the table while you design. After the page design is complete, you'll change BORDER="1" to BORDER="0" to hide the table borders on your Web page.

—   WIDTH defines the table's exact pixel width or specifies the percentage of the browser's window that the table should fill. We're using a table to format the entire page, so the table is sized to fill 100 percent of the browser's window space.

—   CELLPADDING creates a space (measured in pixels) between the cell contents and the table border. After sampling a few spacing parameters, we found that adding a CELLPADDING value of "5" did the trick. This type of setting exemplifies the value of testing settings, saving, and previewing your HTML page during the creation process.

> For added assistance, you can refer to Figure 6-12 while you work through the table creation process.

—   CELLSPACING specifies the amount of space (in pixels) between cells. In the music site, we didn't need to specify any spacing between cells, so we set the attribute to "0".

3   Type <TR> to start the first table row.

4   Press Enter, press Tab to make reading your code easier, and type `<TD VALIGN="top">` to begin the first cell in the first row. The VALIGN="top" attribute indicates that you want to align the cell contents to the top of the cell (by default, cell contents align to the middle). You can align cell contents using the values top, middle, bottom, or baseline. Also, by default, cell contents align at the left. (Later we show you how to change the default alignment by centering contents within a cell.)

5   Press Enter twice, press Tab, and type `</TD>` to mark the end of the first cell in the first row.

6   Press Enter twice, press Tab, and type `<TD VALIGN="top">` to create the second cell in the first row.

7   Press Enter twice, press Tab, and type `</TD>` to mark the end of the second cell in the first row.

8   Press Enter and type `</TR>` to complete the first row of your table.

9   Press Enter and repeat steps 3 through 8 to create the second table row (or copy and paste all the code starting with <TR> and ending with </TR> to create a second table row, as described in the tip).

10   Press Enter after creating the second table row, and then create a third
        row by retyping the commands in steps 3 through 8 or by copying and
        pasting the table row code.

11   After entering the third table row, press Enter and type </TABLE> to
        complete the table tags.

Tip

If you prefer not to retype the table row commands when creating the second and third table
rows, click before the first row's <TR> command and drag to select all the text up to and
including the </TR> closing command. Then press Ctrl+C to copy the selected code.
Click after the </TR> command, press Enter, and press Ctrl+V to paste the copied HTML
code into your text document. Press Ctrl+V again to create the third table row.
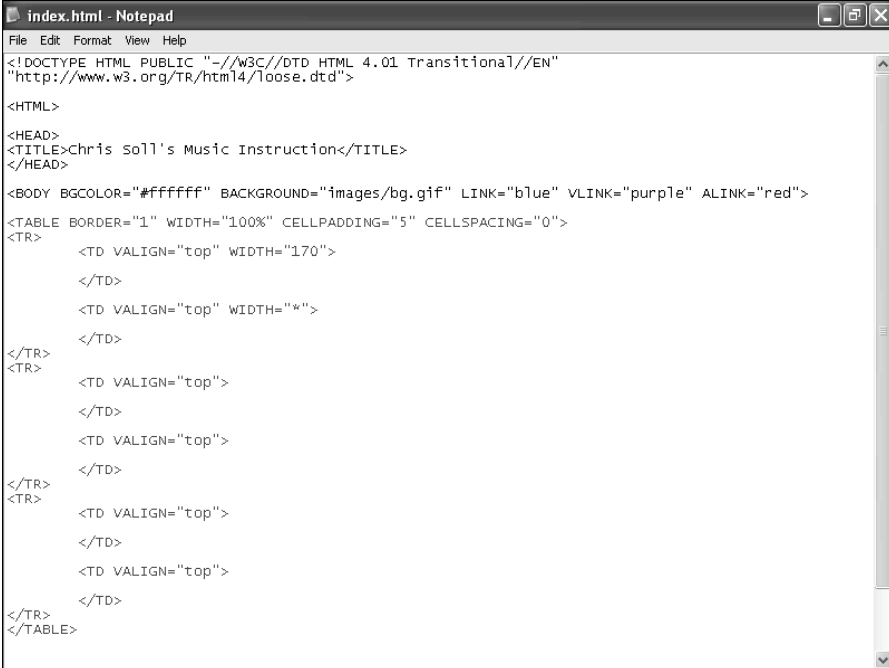
Finally, you'll add a WIDTH attribute to the two cell tags (<TD>) in the first row of the
table. You can assign column width by percentage (for example, the left column could be
assigned to take up 50 percent of the browser window), or you can insert an exact pixel
measurement. By default, if you don't include the WIDTH attribute in table cells, the table
sizes the columns based on the size of the cell content and the size of the browser window.
You need to add the WIDTH attribute to only one cell in a column (and if you have con-
flicting measurements in cells in the same column, the browser will use the largest setting
by default). To keep our setup orderly, we'll add the WIDTH attribute to the cells in the
first row:

Note

If you don't define column widths in tables by adding the WIDTH attribute to <TD>
tags, browsers automatically size the columns based on each column's widest item and the
browser's window size.

1   In the first cell in the first row, click after "top", press the spacebar, and
       type WIDTH="170" to set the first column to 170 pixels wide.

2   In the second cell in the first row, click after "top", press the spacebar,
       and type WIDTH="*". The asterisk in place of a pixel number indicates
       that the browser should allow the second column to be as wide as nec-
       essary to fill the remaining table width. Because this table is formatted
       as 100 percent of the browser window, the asterisk instructs browsers to
       expand the second column to fill the remainder of the browser window
       area.

3   Save your HTML document. Your HTML code should look like the code
       shown in Figure 6-12.

Now that your table is in place, you're ready to enter content into the table. The first order
of business is to insert the logo into the top-left corner.

```
index.html - Notepad
File  Edit  Format  View  Help
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<HTML>

<HEAD>
<TITLE>Chris Soll's Music Instruction</TITLE>
</HEAD>

<BODY BGCOLOR="#ffffff" BACKGROUND="images/bg.gif" LINK="blue" VLINK="purple" ALINK="red">

<TABLE BORDER="1" WIDTH="100%" CELLPADDING="5" CELLSPACING="0">
<TR>
        <TD VALIGN="top" WIDTH="170">

        </TD>

        <TD VALIGN="top" WIDTH="*">

        </TD>
</TR>
<TR>
        <TD VALIGN="top">

        </TD>

        <TD VALIGN="top">

        </TD>
</TR>
<TR>
        <TD VALIGN="top">

        </TD>

        <TD VALIGN="top">

        </TD>
</TR>
</TABLE>
```

Figure 6-12.
The table code creates the structure for your Web page's contents. (The added table code is shown in orange.)

Inserting and Linking the Logo

We're creating a standard page design, so we opted to insert the logo in the top-left (prime real estate) corner. We plan to use the home page as a template for all subpages, so we're going to link the logo to the home page. That way, when you use the home page as a template, all subpages will automatically include a logo that links to the site's index.html home page.

When you insert a logo, you're basically inserting an image. To insert an image in an HTML document, you use the <IMG> tag with the SRC attribute, which points to a particular graphic. For example, to specify the music site's logo, you'd type `<IMG SRC="images/logo.gif">`. Similarly, when you insert your logo and format it as a hyperlink, you use the same HTML codes that you use to link any graphic. So pay attention to the following steps—you'll find yourself using these commands quite a bit. First, let's insert the logo graphic (we take care of linking the graphic in just a bit):

1 If necessary, open your text editor and open index.html.

2 In the first cell in the first row, click after the <TD VALIGN="top" WIDTH="170"> tag, press Enter, press Tab, and type the following HTML tag, which points to the logo image:

<IMG SRC="images/logo.gif" ALT="Chris's Music Instruction" WIDTH="170" HEIGHT="68" BORDER="0">

Tip

Adding spaces and returns in your HTML code won't affect your Web page's appearance, so you don't need to add returns in your HTML document to match the examples in the text. Our text examples had to be shortened to fit properly within the course's page design. Your code can be entered as shown in the project's HTML reference figures included throughout this lesson.

With the exception of the SRC attribute, the <IMG> tag's attributes used in the music site are optional (but very useful) and are defined as follows:

_ SRC specifies the filename of the image (the source of the image) to be displayed.

_ ALT enables you to provide descriptive text that displays when the cursor is placed over the image area.

_ WIDTH and HEIGHT specify the image's width and height. You should specify the sizes of your images because doing so helps browsers display your Web page's layout faster. Keep in mind that any actual image resizing (as in making an image larger or smaller) should be done in your image editing program and not by using WIDTH and HEIGHT attributes in your HTML document—ideally, you want your images to be sized as closely as possible to the size you'll display the images on your Web pages.

_ BORDER specifies the thickness of the border around the image. By default, a 1-pixel border appears around graphics that are formatted as hyperlinks. Generally, designers change the default by setting the BORDER attribute to "0".

Next you'll format the logo.gif image to serve as a hyperlink to the home page. Basically, creating a hyperlink entails marking some text or a graphic as an *anchor* by using the

Lingo: An *anchor* is either the clickable text or graphic component of a hyperlink or a specified target area within a document. Most notably, an-

<A></A> tag set around the text or image you want to serve as a hyperlink, and then specifying to the browser what should be displayed after the anchor element is clicked. To make the logo a hyperlink, follow these steps:

1   Click before the <IMG...> tag and type <A HREF="index.html"> to specify that when users click the logo, they will be taken to the home page. (As mentioned earlier, this linking information will come in handy when you copy the home page to create subpages.)

2   Click after the closing > of the <IMG...> tag and type </ A> to specify the end of the anchor's contents.

Tip

If you're using Notepad and your code goes beyond the edge of the window, open the Format menu and choose Word Wrap. When Word Wrap is activated, Notepad automatically wraps your text and displays all your text in the current window.

3   Save index.html. Your HTML code should look similar to the code shown in Figure 6-13.
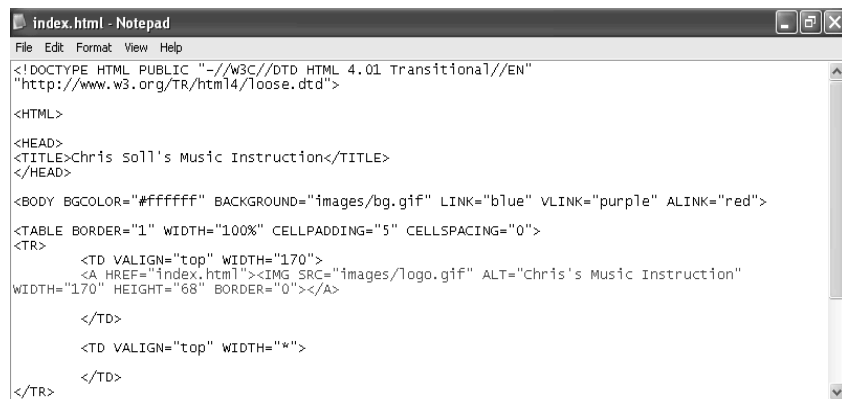
Figure 6-13.
The newly added code inserts a logo and links the logo image to the home page.

Inserting the Home Page Banner Graphic

After inserting the logo, inserting the home page's banner graphic will be a piece of cake. This step entails inserting an image in the second cell of the first table row. You won't have to link this graphic, so the procedure is fairly straightforward. The only twist to inserting the banner graphic is that you'll want to center the graphic within the table cell by nesting the <IMG> tag within the <CENTER></CENTER> tag set. To insert a banner graphic on the home page, perform the following steps:

1   If necessary, open your text editor and open index.html.

2   In the second cell of the first row, click after the <TD VALIGN="top" WIDTH="*"> tag, press Enter, press Tab, and then type the following:

    <CENTER><IMG SRC="images/t_home.gif" ALT="Chris Soll's Music Instruction" WIDTH="415" HEIGHT="62" BORDER="0" ALIGN="middle"></CENTER>

3   Save index.html. Your HTML code should display as shown in Figure 6-14.
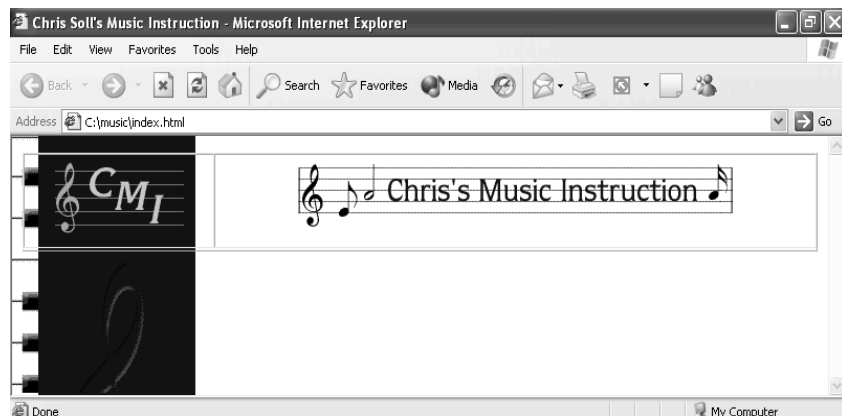
Figure 6-14.
The banner image code specifies to browsers which image should display as the banner on the index.html Web page.

4   Open index.html in your browser. (If index.html is already open in your browser, click the Refresh button to update your view.) The browser should display your version of index.html as shown in Figure 6-15. In the figure, notice that the table borders for the first row display around the inserted graphics. Hold your cursor over the logo or banner graphic to display the image's ALT text.

Tip

You must save your HTML document before you can view the document's changes in a browser window. If your most recent changes aren't displaying in your browser, ensure that you've saved your HTML document. If you still aren't seeing the changes, click the Refresh button in your browser to ensure that you're viewing the most up-to-date version of your page.

Figure 6-15.
At this point, index.html displays in a browser with the linked logo and home page banner graphic, which were positioned using a table.

In the next section, you'll see how to create a navigation bar that you can use throughout the music site.

Adding Navigation Links

Creating a navigation bar for the music site entails inserting button graphics and linking each graphic to a Web page. You've already inserted the logo and banner graphics, so you

know how to use the <IMG> tag. Furthermore, you've linked the logo, so you're also familiar with the <A></A> anchor tags. The only slightly tricky part about using navigation buttons in the music site is that each page shows a custom button for the current page. For example, whenever a user visits the home page, the black version of the Home button displays, and whenever a user visits another page in the site, the standard maroon Home button displays. This setup will become clearer as you progress through this section. To create a navigation bar on the home page, follow these steps:

1   If necessary, open your text editor and open index.html. You'll place the navigation bar in the first column of the table's second row, which will cause it to display on the left side of the page.

2   After the second <TR> tag, click after the first <TD VALIGN="top"> tag, press Enter, press Tab, and type <BR> to insert a blank line between the logo graphic and the upcoming navigation bar.

In the next few steps, you'll embed each button's image tag within an anchor tag that links the button to an appropriately named Web page. You haven't created the subpages yet, so take note of the filenames provided in the anchor tags. The subpage's filenames will have to match the filenames in the anchor references. In all, you need to add seven buttons. You'll start by adding the current-page version of the Home button. (Recall that you downloaded two versions of each button—the current-page version of each button has a "2" at the end of the button's filename.)

3   Press Enter, press Tab, and type the following:

         <A HREF="index.html"><IMG SRC="images/b_home2.gif" ALT="Home Page" WIDTH="170" HEIGHT="24" BORDER="0"></A>

4   Type

 to add two line spaces, press Enter, and press Tab.

You're now set up to enter the next button link and graphic.

Tip

You can repeatedly type the HTML code with varied HREF file references, SRC filenames, and ALT information, or you can copy the code you type in steps 3 and 4 and paste the copied code into the table cell six times. Then, replace the HREF file references, SRC filenames, and ALT text definitions in each entry, as shown in Figure 6-16.

---

5   Type the following:

         <A HREF="lessons.html"><IMG SRC="images/b_lessons.gif" ALT="Lessons" WIDTH="170" HEIGHT="24" BORDER="0"></A>

6   Press Enter, press Tab, and type:

         <A HREF="recitals.html"><IMG SRC="images/b_recitals.gif" ALT="Recitals" WIDTH="170" HEIGHT="24" BORDER="0"></A>

7   Press Enter, press Tab, and type:

         <A HREF="competitions.html"><IMG SRC="images/b_competitions.gif" ALT="Competitions" WIDTH="170" HEIGHT="24" BORDER="0"></A>

8   Press Enter, press Tab, and type:

         <A HREF="performances.html"><IMG SRC="images/b_performances.gif" ALT="Performances" WIDTH="170" HEIGHT="24" BORDER="0"></A>

9 Press Enter, press Tab, and type:

<A HREF="background.html"><IMG SRC="images/b_background.gif" ALT="Background" WIDTH="170" HEIGHT="24" BOR-DER="0"></A>

10 Press Enter, press Tab, and type:

<A HREF="contact.html"><IMG SRC="images/b_contact.gif" ALT="Contact" WIDTH="170" HEIGHT="24" BORDER="0"></A>

Notice that there are four <BR> tags at the end of step 10. The extra line spaces are included because we also want to insert the Send Us A Note icon to give users an easy way to send e-mail messages to Chris. When you create an e-mail link, you use a special HREF format in the anchor tag, as follows.

11 Press Enter a couple times to separate the Send Us A Note HTML code from the main navigation bar HTML code, press Tab, and then type

<A HREF="mailto:mm@creationguide.com">

except replace mm@creationguide.com with your own e-mail address so that users will open a blank e-mail message addressed to you when they click the Send Us A Note icon.

12 Insert the Send Us A Note icon and close the anchor reference by typing:

<IMG SRC="images/sendnote.gif" ALT="E-mail Chris" WIDTH="170" HEIGHT="77" BORDER="0"></A>

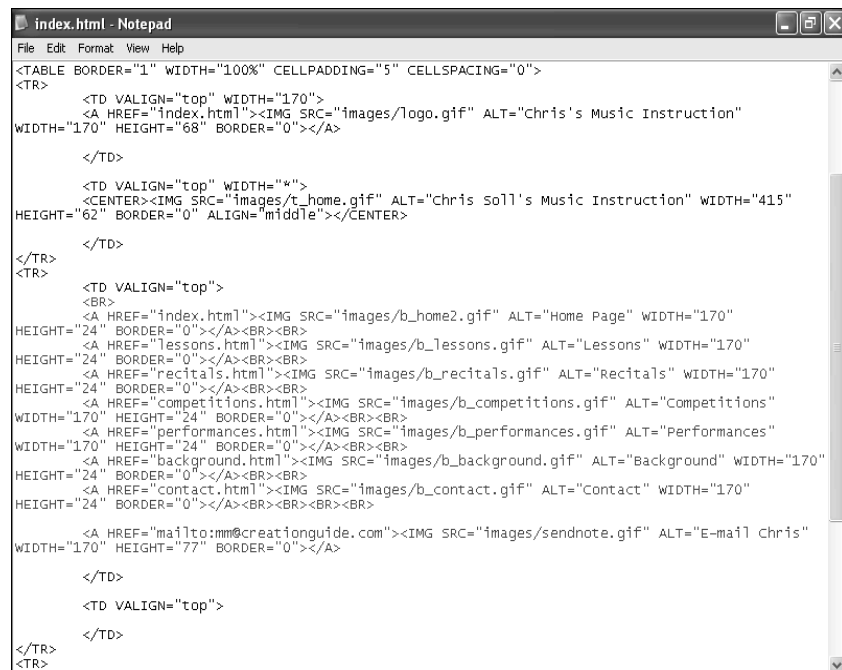**13** Save index.html. Your HTML code should look similar to the code shown in Figure 6-16.

```
index.html - Notepad
File  Edit  Format  View  Help
<TABLE BORDER="1" WIDTH="100%" CELLPADDING="5" CELLSPACING="0">
<TR>
        <TD VALIGN="top" WIDTH="170">
        <A HREF="index.html"><IMG SRC="images/logo.gif" ALT="Chris's Music Instruction"
WIDTH="170" HEIGHT="68" BORDER="0"></A>

        </TD>

        <TD VALIGN="top" WIDTH="*">
        <CENTER><IMG SRC="images/t_home.gif" ALT="Chris Soll's Music Instruction" WIDTH="415"
HEIGHT="62" BORDER="0" ALIGN="middle"></CENTER>

        </TD>
</TR>
<TR>
        <TD VALIGN="top">
        <BR>
        <A HREF="index.html"><IMG SRC="images/b_home2.gif" ALT="Home Page" WIDTH="170"
HEIGHT="24" BORDER="0"></A><BR><BR>
        <A HREF="lessons.html"><IMG SRC="images/b_lessons.gif" ALT="Lessons" WIDTH="170"
HEIGHT="24" BORDER="0"></A><BR><BR>
        <A HREF="recitals.html"><IMG SRC="images/b_recitals.gif" ALT="Recitals" WIDTH="170"
HEIGHT="24" BORDER="0"></A><BR><BR>
        <A HREF="competitions.html"><IMG SRC="images/b_competitions.gif" ALT="Competitions"
WIDTH="170" HEIGHT="24" BORDER="0"></A><BR><BR>
        <A HREF="performances.html"><IMG SRC="images/b_performances.gif" ALT="Performances"
WIDTH="170" HEIGHT="24" BORDER="0"></A><BR><BR>
        <A HREF="background.html"><IMG SRC="images/b_background.gif" ALT="Background" WIDTH="170"
HEIGHT="24" BORDER="0"></A><BR><BR>
        <A HREF="contact.html"><IMG SRC="images/b_contact.gif" ALT="Contact" WIDTH="170"
HEIGHT="24" BORDER="0"></A><BR><BR><BR><BR>

        <A HREF="mailto:mm@creationguide.com"><IMG SRC="images/sendnote.gif" ALT="E-mail Chris"
WIDTH="170" HEIGHT="77" BORDER="0"></A>

        </TD>

        <TD VALIGN="top">

        </TD>
</TR>
<TR>
```

Figure 6-16.
The navigation bar HTML code is inserted into a single table cell.

14 Open index.html in your browser. Your home page should now include a navigation bar on the left, as shown in Figure 6-17. Notice that the Home button displays differently than the other buttons to show that users are currently viewing the home page. Click the logo and Home button to make sure that the home page redisplays. (Clicking any other

button in the navigation bar will display an error because you haven't created those pages yet.) Hover your cursor over each button to ensure that the ALT text is correct throughout. Click the Send Us A Note icon to ensure that a blank e-mail message opens and is addressed to you.
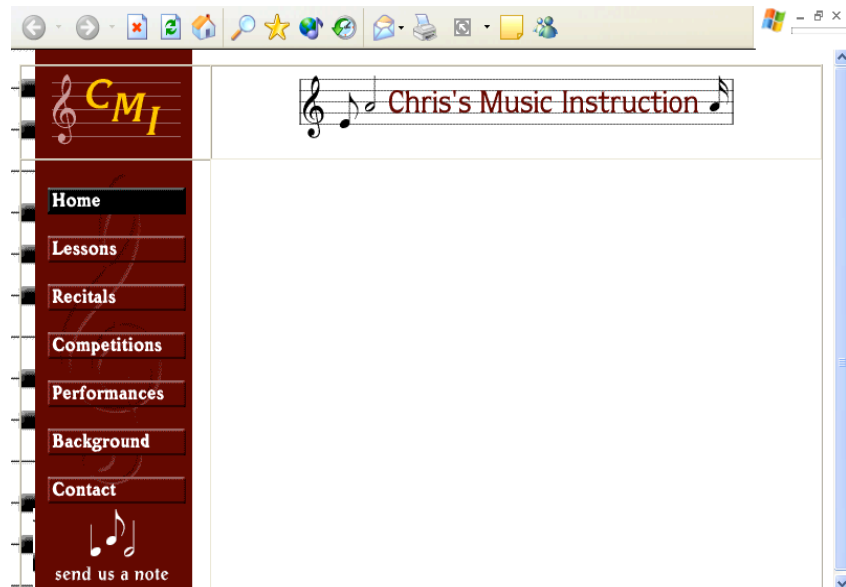

Figure 6-17.
You can test some of your HTML coding by clicking the Home button and the Send Us A Note image in the navigation bar.

Inserting Footer Information

You next need to add the footer information. The music site's footer includes a footer graphic, the address and phone number information, text links that correspond to the navigation bar links, and some copyright text.

The process of inserting a footer graphic is similar to the process of inserting the header graphic described earlier in this lesson. The main difference in the footer area is that you're going to place the footer graphic in the second cell in the third row and you're going to insert some additional information in the cell along with the graphic (as described in the next section). Because you're going to include other information below the footer graphic, you'll nest the footer graphic within paragraph tags ($<$P$><$/P$>$) to ensure that space will be included above the graphic. Let's start to create the footer element by inserting the footer graphic:

1   If necessary, open your text editor and open index.html.

> For added assistance, refer to Figure 6-18 as you insert the Web site's footer information.

2   In the third row of the table, click after the second $<$TD VALIGN="top"$>$ tag, press Enter, press Tab, and then type $<$P ALIGN="CENTER"$>$ to indicate the start of a paragraph and to specify to center align the paragraph's contents. (By default, paragraphs are left-aligned.)

3   Specify the footer image's information by inserting the image tag with the following attributes:

     $<$IMG SRC="images/footer.gif" ALT="Chris Soll's Music Instruction" WIDTH="415" HEIGHT="62" BORDER="0" ALIGN="middle"$>$

4   Type $<$/P$>$ to indicate the end of the paragraph section. The closing $<$/P$>$ is optional, but adding it makes the code clearer.

Next you'll insert a second paragraph that centers the address and phone number information.

5   Press Enter twice, press Tab, and type:

<P ALIGN="CENTER">1234 Songbird Alley, Mesa, AZ 85201
555 555-5555</P>

Notice the <BR> tag in the preceding HTML code between the address text and the phone number. Adding the <BR> tag inserts a line break and displays the phone number below the address on the next line.

Now you'll add some text links in the footer area that correspond to the navigation bar's buttons—that way, if any users have graphics turned off, they can still navigate around your site. To add text links, you follow a procedure similar to creating graphic links. The difference between the two tasks is that you enclose text between anchor links instead of enclosing an IMG tag, as shown in the following steps. First, you'll center the paragraph that will contain the text links.

6 Press Enter twice, press Tab, and type <P ALIGN="CENTER">.

7 Press Enter, press Tab, and type:

<A HREF="index.html">Home Page</A> |

In the preceding link, the words *Home Page* will display as hypertext on the Web page. If users click the words Home Page, the index.html file will display. Also notice the pipe symbol (|). You type this symbol between each text link to make differentiating the links easier. The symbol is optional but popular. You're now ready to add the remaining text links.

Tip

You don't have to press Enter and Tab between each of the text link entries in steps 8 through 14. Browsers will show the links in a row regardless of the space you add in the HTML document. We chose to format the text links in the described way to make working with the information easier.

8 Press Enter, press Tab, and type:

<A HREF="lessons.html">Lessons</A> |

9 Press Enter, press Tab, and type:

<A HREF="recitals.html">Recitals</A> |

10 Press Enter, press Tab, and type:

<A HREF="competitions.html">Competitions<A> |

11 Press Enter, press Tab, and type:

<A HREF="performances.html">Performances</A> |

12 Press Enter, press Tab, and type:

<A HREF="background.html">Background</A> |

13 Press Enter, press Tab, and type:

<A HREF="contact.html">Contact Information</A> |

The final text link entry is the text equivalent of the Send Us A Note icon you inserted earlier. Thus, this text link uses the mailto: component in the HREF attribute. Remember to replace mm@creationguide.com with your own e-mail address in the following step.

14   Press Enter, press Tab, and type:

<A HREF="mailto:mm@creationguide.com">E-mail Chris</A>

15   To complete the text link paragraph, press Enter, press Tab, and type </P>.

The final component of the footer is the copyright information. In this section, you create a centered paragraph and enter the copyright information. One interesting twist here is that you can use a special character entity reference to create a copyright symbol.

Tip

A character entity reference is a special key combination that includes the ampersand (&) symbol and enables you to display nonstandard characters—such as accent marks, registered trademarks, and so forth—in Web pages.

---

16   Press Enter twice, press Tab, and type:

<P ALIGN="CENTER">&copy 2002 Chris Soll. All Rights Reserved</P>

17   Save index.html. Your HTML code should look similar to the code shown in Figure 6-18.

18   Open index.html in your browser. The footer in the index.html file should look similar to the footer information in the page shown in Figure 6-19.

Figure 6-18.
For the music site, the footer information is added to the last cell in the table.

Figure 6-19.
You can see how the HTML footer code is interpreted by a browser by previewing the music site's index.html page.

You might have noticed that you've designed everything on the home page except the main content. You'll be happy to hear that there's a method to our madness. Now that you have the basic structure of the home page created, and because you're planning to mimic the home page structure on the subpages, you can use the index.html file to quickly create the foundation pages for the subpages, as described in the next section.

Copying the Home Page Framework to Subpages

By now you've probably realized that we need to make some pages for the navigation bar and text links to link to. Namely, the music site calls for the following pages:

_ index.html (which you're already in the process of creating)

_ lessons.html

_ recitals.html

_ competitions.html

_ performances.html

_ background.html

_ contact.html

In this section, you're going to create the six additional HTML pages that make up the music site. You could copy all the code from index.html, paste it into a blank text document, and then save the text document as an HTML file, but we're much lazier than that! Here's how we went about creating most of the code for the subpages:

1   Open the C:\music folder.

2   Right-click the index.html file and click Copy on the shortcut menu.

3   Rick-click in the folder and click Paste.

4   Repeat step 3 five times (so that you have six copies of the index.html file). Your music folder should display as shown in Figure 6-20.

5   Right-click the first copy of index.html, click Rename on the shortcut menu, type **lessons.html**, and press Enter. Ensure that you rename

the copied files accurately and using all lowercase letters; if you change the names of the files at this point, the links you created in your HTML code won't work.
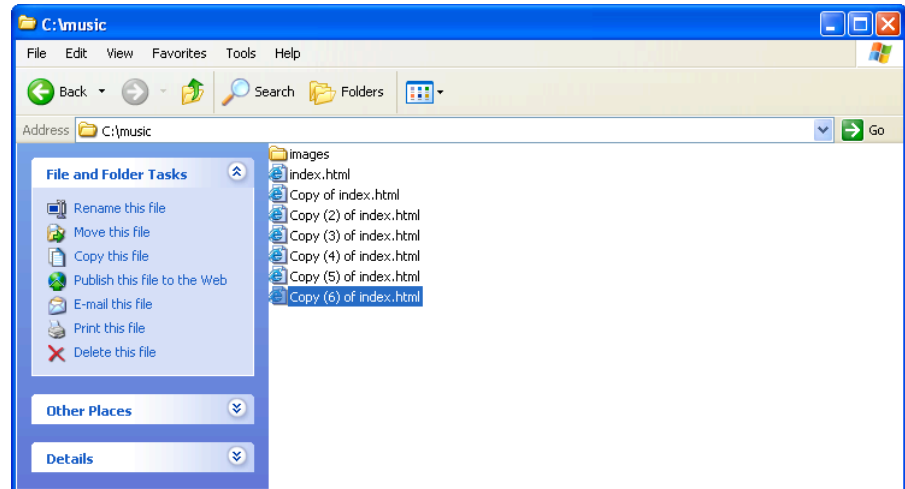

Figure 6-20.
Copying index.html six times creates templates for your site's subpages.

6   Using the procedure described in step 5, rename the subsequent files *recitals.html*, *competitions.html*, *performances.html*, *background.html*, and *contact.html*.

7   Open your text editor, open the lessons.html file, and then perform the following six changes, which are highlighted in color in Figure 6-21:

In the <TITLE> tag, click after the word Instruction and type: Lessons.

_   In the <TABLE> tag, change the BORDER attribute to "0".

_   In the second cell of the first table row, change t_home.gif to t_lessons.gif.

Also in the second cell of the first table row, replace the text ALT="Chris Soll's Music Instruction" with the text ALT="Lessons".

_   In the navigation bar HTML code, change the b_home2.gif Home button text to b_home.gif.

_   Also in the navigation bar HTML code, change the b_lessons.gif Lessons button text to b_lessons2.gif.

Figure 6-21.
To use a copy of index.html as a subpage, you need to slightly modify the HTML code to suit each page.

8   Save lessons.html.

You've just completed the foundation document for the Lessons page. Not too bad! To check your page, double-click lessons.html to view the page in your browser. If everything looks good (as shown in Figure 6-22), you're ready to move on. Your next step is to repeat the short customization process in the remaining subpage documents.

9   Open recitals.html in your text editor, and make the following changes:

In the <TITLE> tag, click after the word Instruction and type: Recitals.

–   In the <TABLE> tag, change the BORDER attribute to "0".

–   In the second cell of the first table row, change `t_home.gif` to `t_recitals.gif`.

–   Also in the second cell of the first table row, replace the text `ALT="Chris Soll's Music Instruction"` with the text `ALT="Recitals"`.

–   In the navigation bar HTML code, change the `b_home2.gif Home button` text to `b_home.gif`.

–   Also in the navigation bar HTML code, change the `b_recitals.gif` Lessons button text to `b_recitals2.gif`.

10   Save recitals.html.

11   Open competitions.html in your text editor and make the following changes:

In the <TITLE> tag, click after the word Instruction and type : `Competitions`.

–   In the <TABLE> tag, change the BORDER attribute to "0".

In the second cell of the first table row, change `t_home.gif` to `t_competitions.gif`.

–   Also in the second cell of the first table row, replace the text `ALT="Chris Soll's Music Instruction"` with the text `ALT="Competitions"`.

> Using a completed foundation page to create subpages is fast and promotes consistency throughout your site.

   —   In the navigation bar HTML code, change the `b_home2.gif` Home button text to `b_home.gif`.

   —   Also in the navigation bar HTML code, change the `b_competitions.gif` Lessons button text to `b_competitions2.gif`.

12   Save competitions.html.

13   Open performances.html in your text editor and make the following changes:

In the <TITLE> tag, click after the word Instruction and type : Performances.

   —   In the <TABLE> tag, change the BORDER attribute to "0".

   —   In the second cell of the first table row, change `t_home.gif` to `t_performances.gif`.

   —   Also in the second cell of the first table row, replace the text `ALT="Chris Soll's Music Instruction"` with the text `ALT="Performances"`.

In the navigation bar HTML code, change the `b_home2.gif` Home button text to `b_home.gif`.

   —   Also in the navigation bar HTML code, change the `b_performances.gif` Lessons button text to `b_performances2.gif`.

14   Save performances.html.

15   Open background.html in your text editor and make the following changes:

In the <TITLE> tag, click after the word Instruction and type : Background.

   —   In the <TABLE> tag, change the BORDER attribute to "0".

   —   In the second cell of the first table row, change `t_home.gif` to `t_background.gif`.

   —   Also in the second cell of the first table row, replace the text `ALT="Chris Soll's Music Instruction"` with the text `ALT="Background"`.

   —   In the navigation bar HTML code, change the `b_home2.gif` Home button text to `b_home.gif`.

   —   Also in the navigation bar HTML code, change the `b_background.gif` Lessons button text to `b_background2.gif`.

15   Save background.html.

16   Open contact.html in your text editor and make the following changes:

In the <TITLE> tag, click after the word Instruction and type: `Contact`.

   —   In the <TABLE> tag, change the BORDER attribute to "0".

   —   In the second cell of the first table row, change `t_home.gif` to `t_contact.gif`.

   —   Also in the second cell of the first table row, replace the text `ALT="Chris Soll's Music Instruction"` with the text `ALT="Contact"`.

      —   In the navigation bar HTML code, change the `b_home2.gif` Home button text to `b_home.gif`.

      —   Also in the navigation bar HTML code, change the b_contact.gif Lessons button text to b_contact2.gif.

17   Save contact.html.

18   Open index.html in your browser. Click each navigation bar button to check your work. A foundation page should display after you click each button; each subpage should include the current page title bar text, a custom title banner graphic, and a current page (black) navigation bar button that corresponds to the displayed page. Also, the table border lines shouldn't display on the subpages (because you changed the BORDER attribute in the <TABLE> tag to "0" on each subpage). Figure 6-22 shows how the Lessons page should display in your browser window.

Figure 6-22.
Each subpage now displays a custom title banner and "active" black button.

Congratulations! You've created the structure for the entire music site. Now we'll insert some content on the home page.

Inserting Body Text on the Home Page

In this section, you're going to insert some body text into the music site's home page. What this endeavor actually boils down to is a practice in formatting text. The main content of the home page consists of a couple headings, paragraph text, colored block quotes, and a linked unnumbered list. So as you can imagine, we'll be discussing how to create these types of elements over the next couple pages.

Creating Headings

In HTML coding, you can define six heading levels by using <H1></H1>, <H2></H2>, and so on through <H6></H6> tag sets, with the size 1 heading being the largest and size 6 being the smallest. Figure 6-23 shows a sample of the various heading sizes.

Figure 6-23.
Standard HTML headings come in six default sizes.

The music site uses one heading tag on the home page, which formats the welcome message at the top of the page:

1 Open your text editor, and then open C:\music\index.html.

2 In the second row and second cell of the table (the cell after the navigation bar information), click after <TD VALIGN="top">, press Enter, and press Tab.
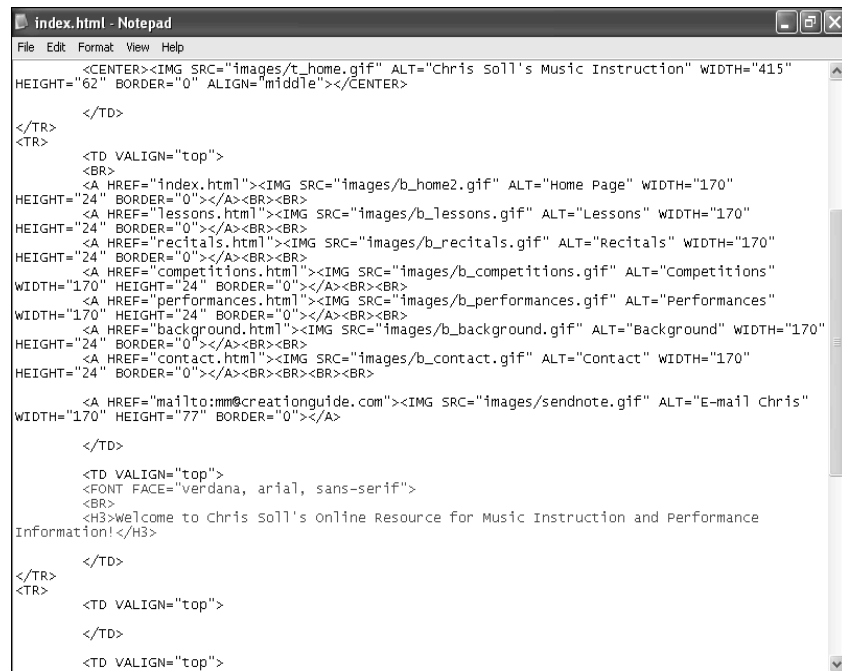
First you'll define the font style for the cell's body text:

3 Type the following:

```
<FONT FACE="verdana, arial, sans-serif">
```

Notice in step 3 the three font names in the FACE attribute. Because you never know what fonts users will have installed on their systems, you should provide a backup plan when you define font styles. In step 3, if a user's computer has the Verdana font installed, the browser will display upcoming body text in the Verdana font style. If Verdana isn't installed on the user's computer, the browser will look for the Arial font family. If neither Verdana nor Arial is installed on the user's system, the browser will display the body text in a sans serif style. If fonts listed in the font tag can't be found on the user's computer, the browser will display the text in the browser's default font style.

Next you'll add a line break and insert a "welcome" heading on the music home page.

4 Press Enter, press Tab, type <BR> to add a line break, press Enter, press Tab, and then type the following size 3 heading information:

```
<H3>Welcome to Chris Soll's Online Resource for Music
Instruction and Performance Information!</H3>
```

5 Save index.html. Your code should appear as shown in Figure 6-24.

Figure 6-24.
The added code specifies a font style and inserts a size 3 heading.

Adding Paragraph Text

Below the welcome heading you created in the previous section, you'll add some body text. The <P></P> tags are the main tags you'll use when entering paragraph text, which you'll do next.

1   Click after the </H3> tag you created in the previous section, press Enter, press Tab, and type **<P>Meet Chris!</P>.**

2   Press Enter, press Tab, and type the following:
<P>She's an internationally certified senior music instructor who teaches piano and flute to budding musicians, hobbyists, and professionals.</P>

3   Press Enter, press Tab, and type the following:
<P>Chris complements her highly regarded instructional program by regularly performing throughout the world as well as orchestrating annual student recitals and competitions.</P>

4   Finally, press Enter, press Tab, and type the following:
<P>Student testimonials sum up Chris's teaching success best:</P>

5   Save index.html.

Your HTML code should appear as shown in Figure 6-25. Furthermore, if you preview index.html in your browser, your home page should look similar to the page shown in Figure 6-26. Your page is all set up except for adding a block quote or two, which is the topic of the next section.
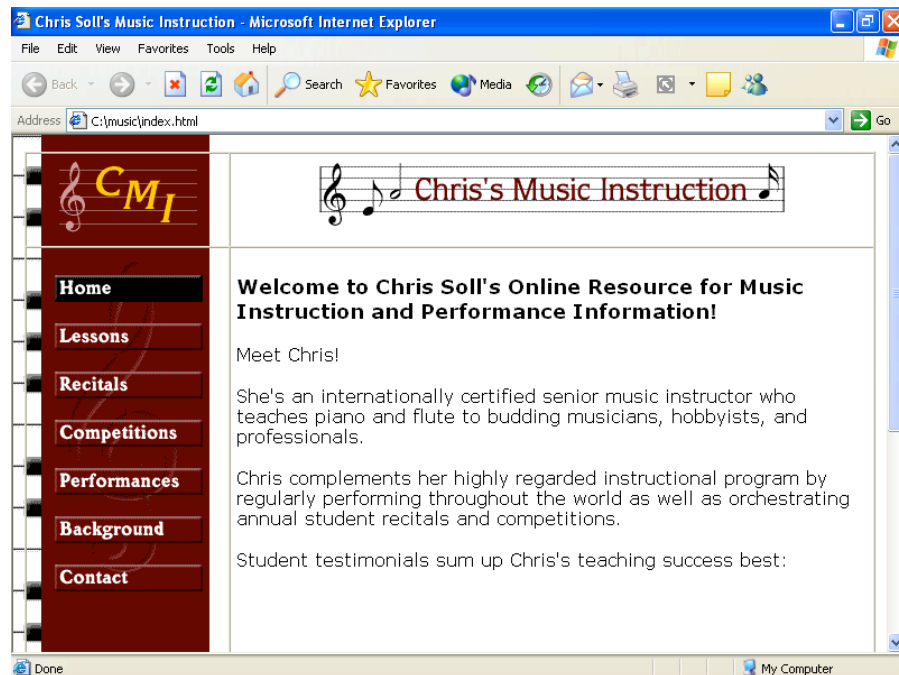
Figure 6-25.

This code shows the paragraph text added to the music site's home page.

Figure 6-26.

At this point, heading and paragraph text should display on your home page.

Formatting Block Quotes and Coloring Text

You can format text as a block quote to set off the text from the surrounding body text. Generally, browsers interpret the <BLOCKQUOTE></BLOCKQUOTE> tag set by indenting the enclosed text's left and right margins. If you really want to indent your paragraph, you can nest block quote commands inside one another, as here: <BLOCKQUOTE><BLOCKQUOTE></ BLOCKQUOTE></BLOCKQUOTE>.

In this section, you'll create block quotes with maroon text:

1 Open index.html in a text editor if necessary, click after best:</P> in the second cell of the table's second row, press Enter twice, and press Tab.

2 Type **<BLOCKQUOTE><BLOCKQUOTE>** to create a block quote nested within a block quote.

3 Press Enter, press Tab, and type the following:
```
<FONT COLOR="maroon"><P><I>Chris is by far the best mu-
sic teacher I've ever had! She taught me more than I
could have learned in a lifetime from Viktor McToned-
eaf, the Royal Music Instructor!</I></P>
```
Notice that the COLOR attribute in the FONT tag is used to modify the color of the paragraph text.

4 Press Enter twice, press Tab, and type the following:
```
<P ALIGN="RIGHT"><I>- Moe Zart</I></P>
```
This inserts a right-aligned, italicized name, which is associated with the block quote entered in step 3.

5 Press Enter twice, press Tab, and type the following:
```
<P><I>Chris Soll is the best teacher I had before "the
incident" that ended my professional music career--
she's a true master!</I></P>
```

6 Press Enter twice, press Tab, and type:
```
<P ALIGN="RIGHT"><I>- Vincent Vanngo</I></P>
```

7 Press Enter, press tab, and type **</FONT>** to end the maroon font color formatting.

8 Press Enter twice, press Tab, and type **</BLOCKQUOTE></ BLOCKQUOTE>** to end the block quote formatting setting.

9 Save index.html.

Your block quote text should display in your HTML document as shown in Figure 6-27. Figure 6-28 shows the block quote text when it's viewed in a browser.

Figure 6-27.
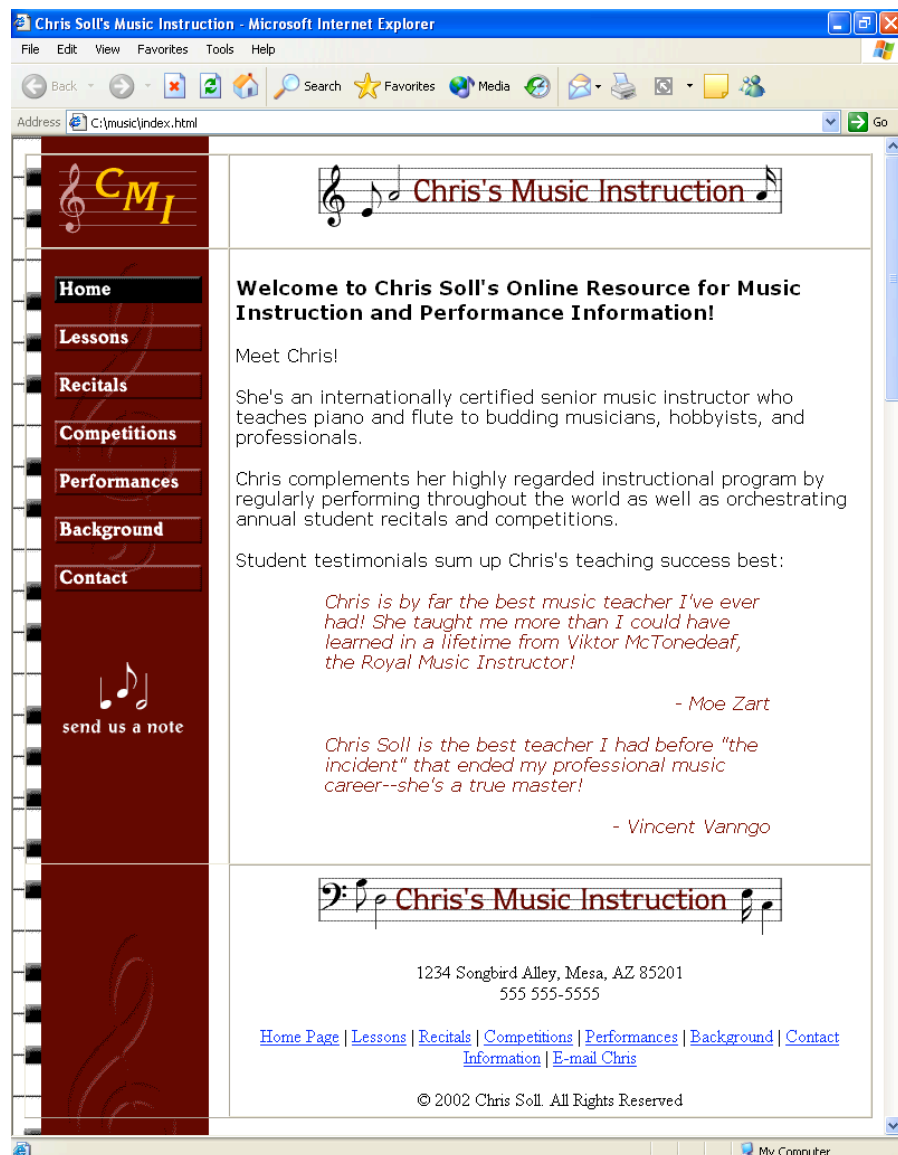*You add block quotes to Web pages to offset text.*

Figure 6-28.
Internet Explorer displays block quotes by indenting the text on the left and right margins.

Creating a Linked Unnumbered List

The final type of body text you'll create on the music home page is an unnumbered list. An unnumbered list appears as a bulleted list on a Web page, as shown in Figure 6-29.

Figure 6-29.
By default, unnumbered lists display with bullets.

Note

You can use <OL></OL> HTML commands to create numbered (ordered) lists, or you can use <UL></UL> commands to create unnumbered (bulleted) lists. Each list item within either type of list is identified by the <LI></LI> tag set.

In this section, you'll first add a small amount of text that introduces the unnumbered list, and then you'll create the list (which includes links to appropriate pages):

1 Open index.html in a text editor if necessary, click after the final </BLOCKQUOTE> in the second cell in the second table row, press Enter twice, press Tab, and then enter the following paragraph:

`<P>Apply for lessons today, and join the elite group of world-class pianists and flutists who study under one of North America's top music instructors.</P>`

2 Press Enter twice, press Tab, and then type the following:

`<P><B>Site Overview</B></P>`

In step 3, you'll begin the unnumbered list by inserting the <UL> tag.

3 Press Enter twice, press Tab, and type **`<UL>`**.

In steps 4 through 9, you'll create list items by surrounding each list item with the <LI></LI> tag set. Furthermore, you'll format the first word of each list entry as a hyperlink to another page in the site.

4 Press Enter, press Tab, and type the following:

```
<LI><A HREF="lessons.html">Lessons</A> Information
about receiving music instruction from Chris,
including availability and fees.</LI>
```

5   Press Enter, press Tab, and type the following:

```
<LI><A HREF="recitals.html">Recitals</A>
Schedules, featured students, and performance piece ti-
tles for student recitals.</LI>
```

6   Press Enter, press Tab, and type the following:

```
<LI><A HREF="competitions.html">Competitions</A> Dates,
fees, rules, schedules, and award ceremony information
for upcoming student
competitions.</LI>
```

7   Press Enter, press Tab, and type the following:

```
<LI><A HREF="performances.html">Performances</A> Calen-
dar of Chris Soll's personal performances, including
dates, locations, performance pieces, and ticket infor-
mation.</LI>
```

8   Press Enter, press Tab, and type the following:

```
<LI><A HREF="background.html">Background</A> Chris's
education, performance, and instructional back-
ground.</LI>
```

9   Press Enter, press Tab, and type the following:

```
<LI><A HREF="contact.html">Contact</A> E-mail
addresses, physical addresses, phone numbers, and map
links that can assist you in applying for
lessons, attending performances, and contacting
Chris.</LI>
```

10   Press Enter, press Tab, and type `</UL>` to end the unnumbered list.

11   Press Enter, press Tab, and type `</FONT>` to complete the body text and end the font family specification.

12   Save index.html.

Your unnumbered list code should display as shown in Figure 6-30 (and your index.html page should display as shown earlier in Figure 6-29).

```
index.html - Notepad
File  Edit  Format  View  Help

        <P><I>Chris Soll is the best teacher I had before "the incident" that ended my
professional music career--she's a true master!</I></P>

        <P ALIGN="RIGHT"><I>- Vincent Vanngo</I></P>
        </FONT>
        </BLOCKQUOTE></BLOCKQUOTE>

        <P>Apply for lessons today, and join the elite group of world-class pianists and flutists
who study under one of North America's top music instructors.</P>

        <P><B>Site Overview</B></P>

        <UL>
        <LI><A HREF="lessons.html">Lessons</A> Information about receiving music instruction from
Chris, including availability and fees. </LI>
        <LI><A HREF="recitals.html">Recitals</A> Schedules, featured students,and performance
piece titles for student recitals. </LI>
        <LI><A HREF="competitions.html">Competitions</A> Dates, fees, rules, schedules, and award
ceremony information for upcoming student competitions.</LI>
        <LI><A HREF="performances.html">Performances<A> Calendar of Chris Soll's personal
performances, including dates, locations, performance pieces, and ticket information.</LI>
        <LI><A HREF="background.html">Background</A> Chris's education, performance, and
instructional background.</LI>
        <LI><A HREF="contact.html">Contact</A> E mail addresses, physical addresses, phone
numbers, and map links that can assist you in applying for lessons, attending performances, and
contacting Chris.</LI>
        </UL>
        </FONT>

        </TD>
</TR>
<TR>

        <TD VALIGN="top">

        </TD>

        <TD VALIGN="top">
        <P ALIGN="CENTER"><IMG SRC="images/footer.gif" ALT="Chris Soll's Music Instruction"
WIDTH="415" HEIGHT="62" BORDER="0" ALIGN="middle"></P>

        <P ALIGN="CENTER">1234 Songbird Alley, Mesa, AZ 85201<BR> 555 555-5555</P>
```
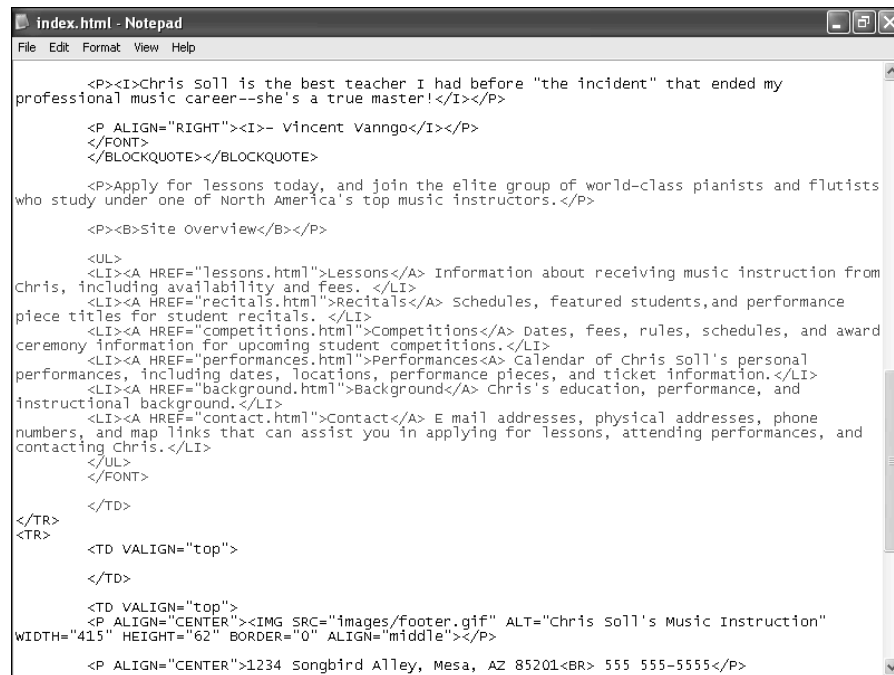
Figure 6-30.
The highlighted HTML code includes paragraph text and an unnumbered list, which contains hyperlinks.

Last but not least, you'll insert a linked photograph into the body area of the music site's home page.

Inserting and Linking a Picture

By now, you should be very comfortable with inserting and linking graphics. (You had a lot of practice just a little while ago when you were creating the navigation bar!) For this page, we opted to display the picture below the size 3 heading and along the right side of the page. Furthermore, because the photograph is a picture of Chris, we linked the photograph to the Background page.

To insert a linked picture within your page's body text, follow these steps:

1   Open index.html in a text editor if necessary, click after the </H3> tag in the second cell in the second row of the table, press Enter, and press Tab.

2   Type the following link and image information:
```
<A HREF="background.html"><IMG
SRC="images/p_chris.jpg"
ALT="pic: Chris Soll" WIDTH="170" HEIGHT="250"
BORDER="0" HSPACE="25" VSPACE="10"
ALIGN="right"></A>
```
In this step, you can see some added attributes to the <IMG> tag, which are defined as follows:

–   HSPACE enables you to specify extra space (in pixels) between the image and text on the left and right sides of the image.

–   VSPACE enables you to specify extra space (in pixels) between the image and text above and below the image.

–   ALIGN indicates to align the picture on the page, and text wraps accordingly. In this example, the picture is aligned to the right side of the page.

3   Save index.html.

Finalizing the Home Page

Finally, you need to remove the table borders on the home page before you can proclaim your project complete. In addition, you should check your links and view all your pages to ensure that you've entered accurate HTML code. First let's get rid of those borders on the home page:

1   Open index.html in a text editor.

2   In the $<$TABLE$>$ tag, change the BORDER attribute from "1" to "0".

3   Save index.html.

Now let's click around and check your links, graphics, ALT text, and other page elements. For example, you need to make sure that none of your pages display with two black buttons.

4   Open your browser, display index.html (or if the document is already open in your browser, click Refresh), and then click every link (including the linked picture, logo, Send Us A Note icon, and text links) to verify that your links work properly and your pages display correctly. If any links don't respond as expected, open the proper HTML document in your text editor and check the HTML code carefully. Check your banner graphics and button graphics to ensure that you've included the proper graphics on each page.

Note

You might have noticed that we provided content only for the home page. At this point, we think you're well enough prepared to enter content into the other pages if you desire more practice using HTML.

Using the Music Site's Framework as a Template

After all your hard work creating the music site, we wanted you to have a useful HTML template that you can easily customize. Therefore, we're going to let you in on a little secret. You can create a Web site using the music template even if you don't have any graphics. You don't have to have a background image, banner graphics, or buttons. Instead, you can create an entire Web site using text links and color backgrounds by replacing the content elements in the music site's Web pages. Figure 6-31 illustrates a Web page that uses the music Web site without graphics.
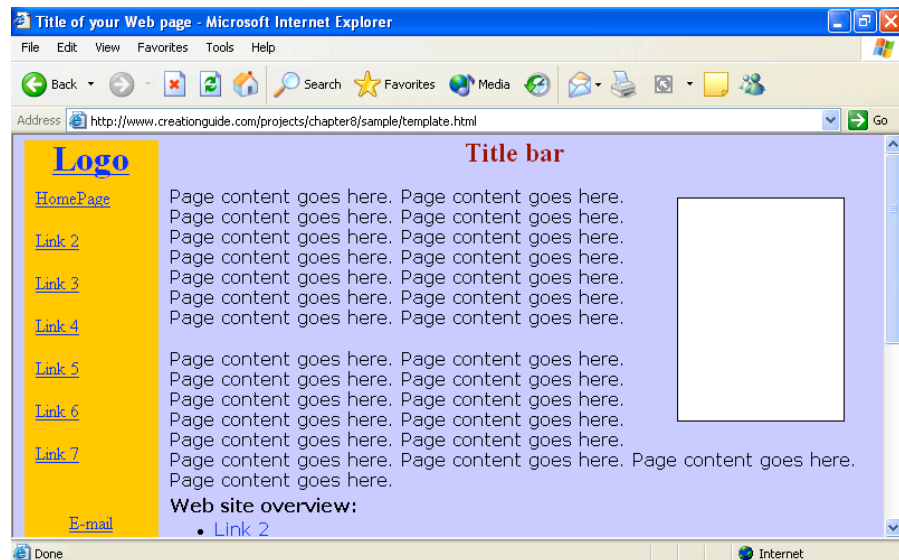

Figure 6-31.
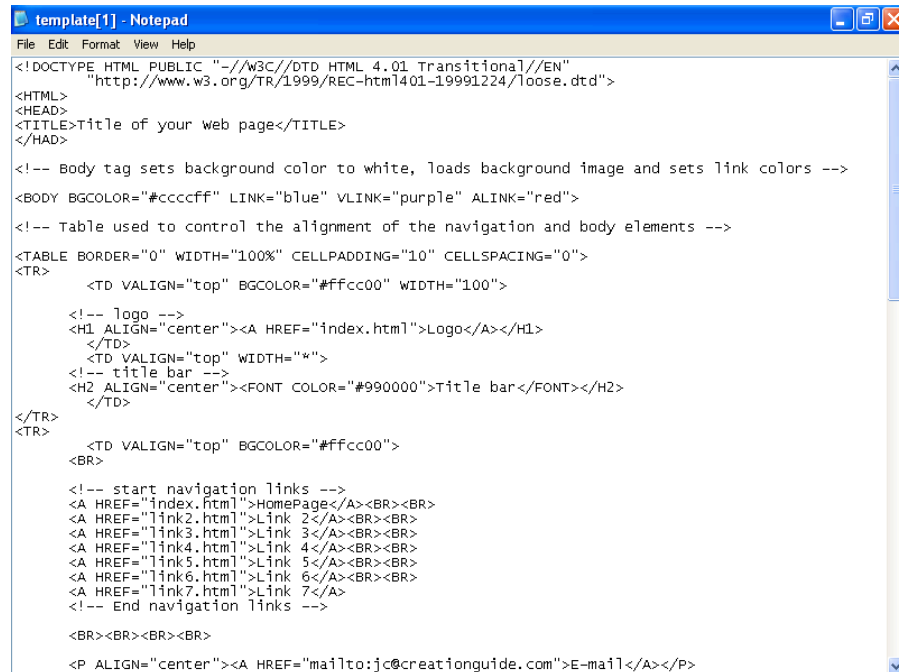*You can create a text-based Web site that uses the music site as a template.*

Tip

The small graphics on the sample page in Figure 6-35 are clip-art images downloaded and pasted together to create a simple picture. If you prefer not to include a graphic on your page, simply delete the $<IMG>$ tag in your HTML document.

Figure 6-32 shows the template page, and Figure 6-33 shows some of the template's source code. To speed along your progress when using a template to build a custom page, display the template's source code by clicking the View menu and then Source. Then in the source code window, click the File menu, click Save As, and save the source code to your computer. At that point, you're free to modify your local version of the template by replacing the placeholder text with your custom content.

Figure 6-32.
You can modify the text-based template to create a custom page.

Figure 6-33.
You can copy the template's source code to get a good start on the code for your custom page.

Tip

In Figure 6-33, you can see HTML code entries that begin with an exclamation point and two dashes: $<!--$ logo $-->$. These types of entries are called *comments*. Comments don't display on a Web page; they're used as notes to developers to help label parts of the HTML document. Comments have been added to the sample template to help you identify HTML components when you customize the source code.

Finally, notice in Figure 6-33 that the <BODY> tag's BGCOLOR attribute has been changed from white (#ffffff) to purple (#ccccff), and that the first <TD> tag in each table row contains a BGCOLOR attribute set to gold (#ffcc00). You can use the BGCOLOR attribute to color your page's background as well as to color your table's cells if you're not using a background image. You can get fairly creative with color by coloring the background one color and then selectively coloring table cells various other colors. (You don't have to color all your table cells the same color.)

Additional Resources

As we mentioned, HTML books and Web pages abound. Here are a few of our favorite HTML resources:

- Castro, Elizabeth. *HTML for the World Wide Web*. 4th ed. Berkeley, CA: Peachpit Press, 2000. (ISBN: 0-201-35493-4) This book does a nice job of visually showing how to use most HTML commands. The textual explanations are brief yet helpful when used in conjunction with the visual references.

- Morrison, Michael. *Faster Smarter HTML & XML*. Redmond, WA: Microsoft Press, 2002. (ISBN: 0-7356-1861-5) This book targets interested but inexperienced users who want to develop Web pages by using HTML. A nice feature of this book is that it offers numerous opportunities for hands-on learning.

- Creation Guide Resource page (*www.creationguide.com/resources*) is continually updated to include current Web development resources.

- World Wide Web Consortium (*www.w3.org*) Web site is the online home for the official most current HTML specifications. As of this writing, HTML 4.01 is the currently accepted specification. Use the Table of Contents, Elements, and Attributes links along the top of the page to find HTML commands and command descriptions. You can find a helpful list of HTML elements at *www.w3.org/TR/1999/REC-html401-19991224/index/elements.html*.

- Web Developer's Virtual Library (*www.wdvl.com*) provides resources, sample code, and tutorials for Web developers.

- webmonkey (*www.webmonkey.com*) provides numerous resources for online developers.

Key Points

- HTML commands serve as instructions that tell a browser how to display a Web page's content.

- HTML commands appear between < > marks, usually come in pairs, and are not case-sensitive.

_ Opening HTML tags frequently contain attributes to further refine the tag's instructions.

_ The standard tags for HTML documents consist of the following:

_
```
<!DOCTYPE...>
<HTML>
<HEAD>
<TITLE></TITLE>
</HEAD>
<BODY>
</BODY>
</HTML>
```

_ If possible, store all your Web site documents and graphics within the same master folder. Create an images folder for your Web site's images within the master folder.

_ To make life easier, devise a graphics naming system to help differentiate various image types, such as buttons, banners, pictures, and so forth.

_ When creating Web pages, save your HTML documents and preview your pages in a browser frequently.

_ Always verify that you've included all angle brackets (<>) and quotation marks ("") in your code. Missing small elements or misspelling HTML commands can cause your page to display incorrectly (or not at all) in a browser.