chapter 7

Interacting with Users

"Tell me and I will forget. Show me and I will remember. Involve me and I will understand."

—Aristotle

In this chapter, you will learn how to:

- Define the elements of an HTML form and create a Web form to collect information from site visitors.
- Create a PayPal button to enable someone to select a product advertised at your Web site and pay for it online.
- Define the elements of an HTML image map and create such a map to permit users to interact graphically with your Web site.
- Describe the concept of a frameset, define the HTML elements it comprises, and create a frameset that enables users to click items in a sidebar frame to display the corresponding content in the main content frame.
- List the steps involved in FTPing files to the Web and set file permissions that either make files publicly available or restrict access to authorized users.

To unleash the true potential of the Web, you need to make your pages interactive. Why? Because it is through interacting with your users that you learn things about them, such as their preferences and needs. Then you can act on the basis of those needs to make your site responsive to what your users want. This just-in-time responsiveness is what differentiates interactive sites from static pages that have no way of obtaining knowledge about users.

This chapter teaches you how to create four kinds of interactions onscreen. First, you learn how to create an HTML form that can collect information from your users. This information can consist of a textual response that the user types in response to an onscreen prompt, or it can be a choice the user makes by selecting an item in a menu of possible choices. Second, you learn how to use a preprogrammed e-commerce button that people can click to buy things you offer for sale at your site. Later in this book, Chapter 12 will teach you how to program your own buttons by writing scripts that can save results and query a database to power your site.

Third, this chapter steps you through the process of creating a dashboard-style graphic known as an image map where users can click hotspots to make selections at your site. This kind of image mapping enables you to create graphical interactions that cannot be expressed in words. Finally, you learn how to use a frameset containing multiple windows called frames that interact with each other to minimize the time required for users to navigate your site. Because not all users like framesets, however, I will share with you a strategy for designing site content in such a way that you can give users their choice of either using the frameset or not. Then I show you how to publish your site to the Web by using an FTP client window.

Creating HTML Forms

A **form** is an HTML Web page element that prompts the user to fill in one or more blanks and/or select items from a menu of possible choices, after which the user clicks a button to submit the filled-out form. Many forms also contain a Reset button that the user can press to clear the form and start over. The blanks the user fills in can be short-answer fields or text areas capable of holding larger amounts of text. Forms can display specific choices in dropdown menus, check boxes, or radio buttons. All of these form controls have HTML tags explained in the following sections.

Getting Information from Users

As you might expect, the <form> start and </form> stop tags mark the beginning and ending of an HTML form. Between these tags go the controls that prompt the user for specific information, along with any onscreen instructions you may wish to provide. This tutorial teaches you these controls in the context of a real-world example in which you will create a form that prompts users to tell you their name, e-mail address, and the frequency with which they would like to receive news from your school or company. When a user clicks the Submit button, the form uses a mailto to send you a message telling you what the user filled in on the form. After you complete the coding of this mailto example, other methods of processing forms will be discussed, along with more advanced controls you can display onscreen as part of the form.

Designing the Prompts

A **prompt** is an onscreen instruction that asks the user to provide some information, either by typing a response into a text field or by choosing something from a selection of possible choices. The secret to creating a user-friendly form is to design the prompts in such a way that users can easily understand what you are asking them to provide. In this example, you prompt the users to type their name and e-mail address and then tell you whether they want to receive your news daily, weekly, or monthly. Think about how you might word these prompts. Then have a look at Figure 7-1, which shows how I decided to word them onscreen. If you prefer alternate wording, you can reword these instructions when you learn how to write the code for this example. Read on.

Coding the Form

To create the form that appears in Figure 7-1, you first use the Notepad to create a new HTML file into which you will type the tags that create the form's controls. Follow these steps:

1. Pull down the Notepad's File menu and choose New to start a new file. Into this new file, type the following code, which creates the overall structure of the page:

```
<html>
<head>
  <title>Newsletter Subscription Form</title>
</head>
<body>
<h1>Newsletter Subscription</h1>
</body>
</html>
```

2. Pull down the Notepad's File menu and choose Save As to save the file. When the Save dialog appears, use it to save the file in your *website* folder as **subscribe.html**. Use your browser to preview the file and see how it looks onscreen.

FIGURE 7-1 *When you create an HTML form, you should word the prompts clearly and arrange the controls logically so users can easily intuit how to fill out the form. In this example, the default value of the Submit button has been changed to Subscribe, because clicking the button will subscribe the user to the newsletter.* ■

3. In the Notepad file, click to position your cursor after the heading's </h1> stop tag, but before the </body> stop tag. This is where the form tag is going to go. Every <form> start tag has two attributes, namely, a method and an action. Because this form is going to post the results to a mailto that will e-mail you the form results, you type your form tags as follows, replacing Santa's e-mail address by your own. In this tutorial, the gray lines already exist in your Notepad file; you type the new black lines between the existing gray lines:

```
<h1>Newsletter Subscription</h1>
<form method="post" action="mailto:santa@northpole.com">
</form>
</body>
```

4. Save the file.

Now you are ready to code the input fields into which users will type their name and e-mail address.

Coding the Text Field Input Controls

In HTML, you use the **<input> tag** to create the form controls that receive input from the user. The <input> tag has a *type* attribute that specifies the kind of control you want. In this example, you are about to create the text fields into which the users will enter their name and e-mail address. To create a text field, you set the <input> tag's type to *text*. The <input> tag also has a **name attribute** that you use to specify the name of the control, and a **size attribute** that specifies how many characters wide the field will be onscreen. There is also a **maxlength attribute** that specifies the maximum number of characters the user can type. You will see all of these attributes in action as you follow these steps to create the text field input controls:

1. Click to position your cursor between the <form> start and </form> stop tags, and type the following code, which displays a prompt asking users to type their name, followed by the <input> tag that creates the corresponding text field:

```
<form method="post" action="mailto:santa@northpole.com">
<p>What is your name?
<br>
<input type="text" name="Name" size="50" maxlength="150">
</p>
</form>
```

2. Save the file and preview it with your browser. Click Refresh to make sure you are viewing the latest version of the file, which should now display the Name field. Troubleshoot any problems that may appear onscreen.

3. You follow a similar process to create the text field into which users will be asked to type their e-mail address. In the Notepad file, click to position your cursor immediately before the </form> tag, and type the following code to prompt users for their e-mail address:

```
</p>
<p>What is your e-mail address?
<br>
<input type="text" name="Email" size="50" maxlength="150">
</p>
</form>
```

4. Save the file and preview it with your browser.

After you troubleshoot any problems that may appear onscreen, you will be ready to code the radio buttons with which the user will choose whether to receive your news daily, weekly, or monthly.

Coding the Radio Button Input Controls

A **radio button** is a form control that displays a small round button with which the user can select one, but not more than one, item from a list of possible choices. Alongside each item, the browser displays a radio button that gets filled in when the user clicks it. If the user clicks a different item in the radio button group, the newly clicked button gets filled in and the previously selected button gets cleared. This is how the buttons work on a push-button radio; hence the term radio button.

In an HTML form, you create a radio button by setting an <input> tag's type attribute to radio. You use the name attribute to assign a name to the radio button group, which includes any other radio button of the same name within the form. You will see how this works as you code the radio buttons for the subscribe form you are creating in this tutorial. Follow these steps:

1. Use the Notepad to open the *subscribe.html* file you have been working on in this tutorial. Click to position your cursor after the e-mail paragraph's close tag, and type the following code to create the first radio button, which users will click if they want to receive your newsletter on a daily basis. Notice that the <input> tag includes the *checked* attribute, which will cause this item initially to be selected as the default choice on the form:

```
<input type="text" name="Email" size="50" maxlength="150">
</p>
<p>
How often do you want to get our news?
<br>
<input type="radio" name="Frequency" value="daily" checked>Daily
</form>
```

2. Save the file and view it with your browser to make sure the form looks good so far. After you troubleshoot any problems, type the

following code, which adds the other two radio buttons to the form. Notice that these buttons have the same *name* attribute but different *value* attributes. Remember that the name attribute is the name of the radio button group, and the value attribute is the specific setting the user can choose from that group. Here is the code to type:

```
<input type="radio" name="Frequency" value="daily" checked>Daily
<br>
<input type="radio" name="Frequency" value="weekly">Weekly
<br>
<input type="radio" name="Frequency" value="monthly">Monthly
</p>
<p>
<input type="submit" value="Subscribe"> <input type="reset">
</p>
</form>
```

3. Save the file and preview it with your browser. Click Refresh to make sure you are viewing the latest version of the file. You should see three radio buttons onscreen. Troubleshoot any problems you may see onscreen.

Now you are ready to complete the form by coding the buttons that the user will click to submit or reset the data on the form.

Coding the Submit and Reset Buttons

The final step in creating a form is to create the **Submit button**, which is the onscreen control that users will click to submit the form. In an HTML form, you make a Submit button by creating an <input> tag that has a type attribute set to *submit*. By default, the value the browser prints inside the button is the same as its type. You can change this value via the value attribute. I like to make the buttons say what they are going to do. In the *subscribe.html* example you are creating in this tutorial, the button is going to subscribe the user to a newsletter. Therefore, we will make its value be "Subscribe" so the button will say Subscribe instead of Submit.

Many forms also have a **Reset button** users can click to return the controls to their default settings. In an HTML form, you make a Reset button by creating an <input> tag that has a type attribute set to reset. The *subscribe.html* example you are creating in this tutorial has both a Submit button and a Reset button. To code these buttons, follow these steps:

1. Click to position your cursor immediately before the </form> tag that ends the form, and type the following code:

```
<input type="submit" value="Subscribe"> <input type="reset">
</p>
<p>
<input type="submit" value="Subscribe"> <input type="reset">
</p>
</form>
```

2. Save the file and view it with your browser. Compare your screen to the model displayed previously in Figure 7-1 and see if everything looks okay. Troubleshoot any problems. Figure 7-2 shows the completed HTML source code for the model in Figure 7-1.
3. Now you are ready to fill in the form and click the buttons to see what they do. The first time you try this, click the Reset button. You will observe that it resets the form. Try clicking different radio buttons. Notice that clicking the Reset button always resets the form to the button you coded to be *checked* by default.
4. Fill in the form again, but this time, click the Subscribe button. Because the Subscribe button is of type submit, clicking the Subscribe button submits the form data. If this works properly, you have successfully coded the form.

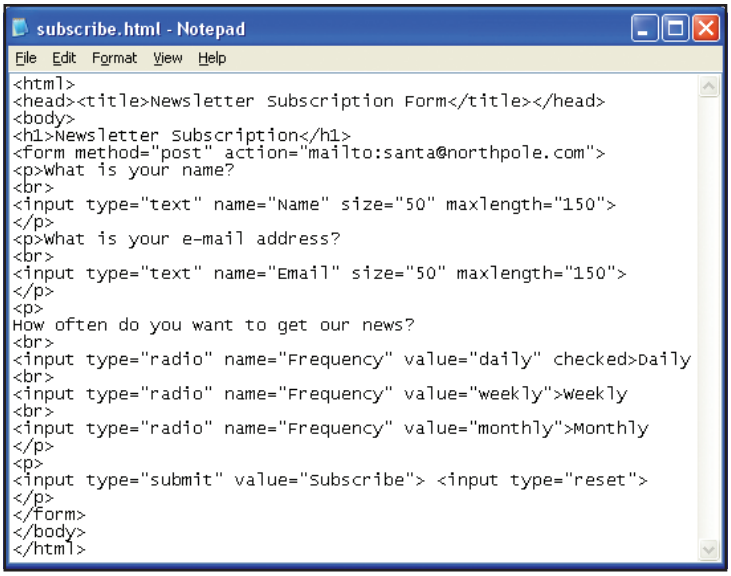


FIGURE 7-2 *This is completed HTML source code that creates the form displayed in Figure 7-1. If you have any trouble getting the form to work, compare your HTML code closely to the commands displayed here. Correct any problems, save the file, and then view it with your browser. Remember to click the browser's Refresh button to make sure you are viewing the latest version of the form.* ■

Congratulations! Now we must discuss what happens in processing the response to a form.

Processing the Response

In the *submit.html* tutorial you just completed, you programmed the form tag to post its data to a mailto, which caused your browser to send a message to the e-mail address you provided in your mailto code. Depending on the brand of browser you are using, either the mail message contained the posted data in the body of the message or the data appeared in a file attachment. The IE browser, for example, attaches form data in a file named *Postdata.att*, which is a plain text file you can open with any text editor, such as the Notepad. Although this is a simple way to handle a form, mailto is very powerful because it provides you with a way to collect information from a site visitor without needing to do any fancy scripting or server-side programming. Thus, mailto is handy for beginners.

In developing more advanced Web applications such as e-commerce solutions, however, you must post the form data to a server that is programmed to read the information, save the form data in a database, decide how to act on it, and respond appropriately to the user . You use the form tag's `method` and `action` attributes to post data to a server-side program. One of the most popular server-side programming environments is Microsoft's active server page (ASP) technology. Imagine that Santa Claus has an ASP script named WishList.asp to which good little girls and boys can post information about the toys they want. Suppose the HTTP address

of this script is http://toys.northpole.com/WishList.asp. To make a form post its data to this script, you would code the form tag as follows:

L 7-1
```
<form method="post" action="http://toys.northpole.com/WishList.asp">
```

The art of server-side programming is known as **common gateway interface (CGI)** scripting. There are many programming languages in which CGI scripts can be written. The CGI defines the protocol that these scripts use to send and receive data across the Internet. You learn more about CGI programming in Chapter 12.

Besides the **POST method** that sends the form data to a CGI script, there is also an HTTP **GET method** that appends the form data to the URL as a query string. You see forms with method="get" used especially at search engine sites, which use the query string to "get" what you are looking for. If you do a Google search, for example, take a look at the URL in the browser's Address field after you type your search terms and click the Search button to see how the form data gets appended to the URL. If you search Google for cars, for example, the form data gets appended to the URL as follows:

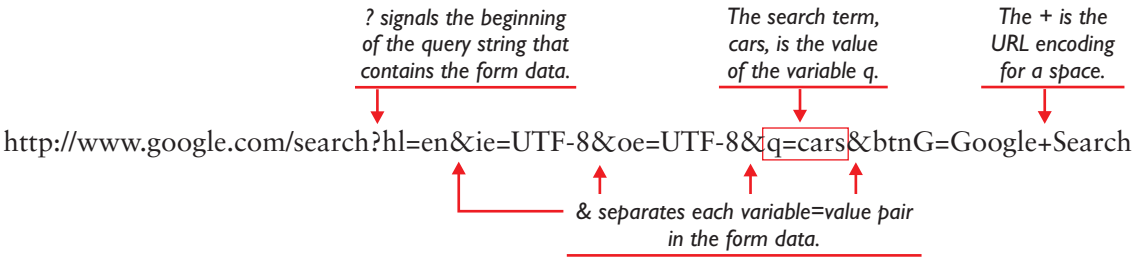


Table 7-1 summarizes the HTTP form submission methods. For more detailed information about these methods, go to www.w3.org/TR/html4 and follow the quick table of contents link to forms.

Creating Check Boxes

When you coded and tested the *subscribe.html* page earlier in this chapter, you observed that the radio button permits you to select only one of the items included in the control. For situations in which you want the user to be able to select more than one of the items contained in a form control, you use check boxes. A **check box** is a form control that presents the user

Attribute	What It Does
`method="get"`	The GET method puts the form data in a query string that gets appended to the URL specified in the form tag's `action` attribute.
`method="post"`	The POST method sends the form data to the process identified in the form tag's `action` attribute. This process is normally a server-side script that reads and processes the form data, saves appropriate information in a server-side database, and returns an HTML response to the user.

TABLE 7-1 *HTTP Form Submission Methods* ■

with a small box that the user can click to turn an option on or off. To create a check box, you make an <input> tag of type check box. The syntax is

Replace `VariableName` *with the name of the variable in which you want the form to return the value.*

Replace `VariableValue` *with the value the variable will be assigned if the user checks this*

L 7-2
```
<input type="checkbox" name="VariableName" value="VariableValue" checked>
```

Include the `checked` *attribute only if you want the check box to be selected by default.*

Imagine that you want to create check box controls that will enable users to tell what news topics they are interested in. You would code such controls with the HTML displayed in Figure 7-3. When viewed with a browser, the check boxes appear as illustrated in Figure 7-4. If you want to try these check boxes in a working example, type the HTML displayed in Figure 7-3 into your *subscribe.html* file right before the paragraph that contains the Submit button. Then save the file and view it with your browser.

Creating Single and Multiple Item Selection Menus

In user interface design, selection menus are a very popular and effective way of getting users to make choices. On a Web form, you can create selection menus that either drop down or display their contents in a scrolling menu onscreen. Furthermore, you can set up the menu for a single selection, or you can permit multiple selections.

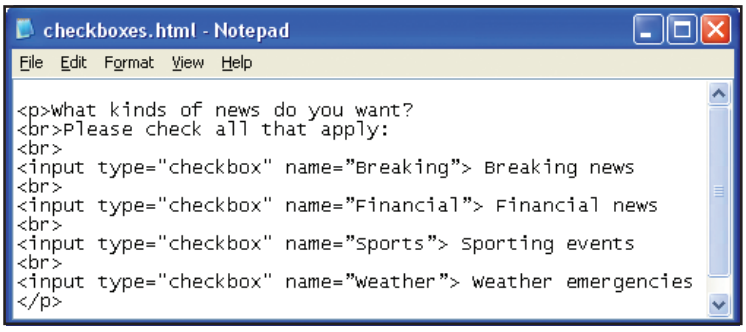
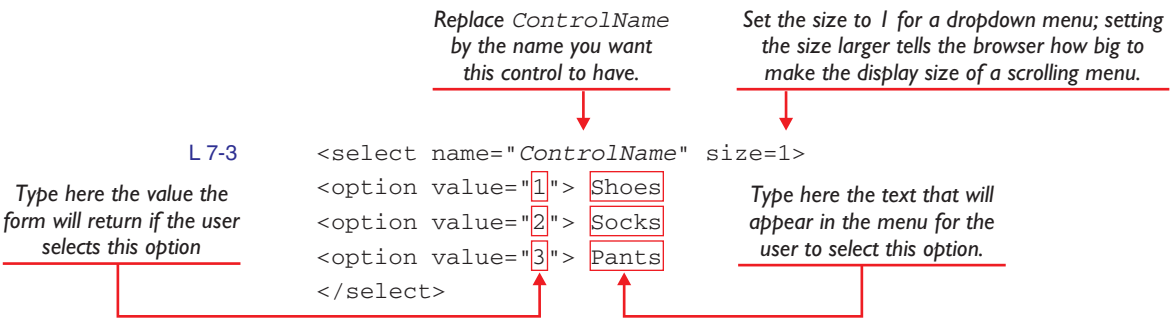


FIGURE 7-3 *You make an HTML check box control by creating an <input> tag of type check box. Compare this code to Figure 7-4, which shows this code being viewed by a browser.* ■

FIGURE 7-4 *You should word check box instructions in such a way that the user knows to check all that apply.* ■

You create a selection menu via the **<select> tag**, which has a corresponding `</select>` stop tag. Between the `<select>` start and `</select>` stop tags, you use the **<option> tag** to specify the menu options from which the user can make a selection. The command syntax is outlined as follows:



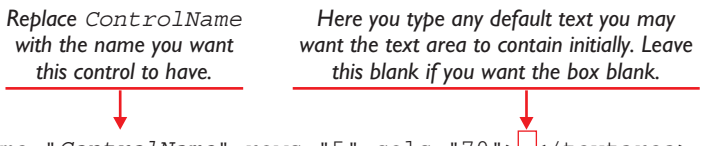
To permit multiple selections, you add the *multiple* attribute to the select tag. When you create a multiple selection menu, you normally set the size large enough to display all or many of the menu items at once. The following code, for example, creates a multiple selection menu that displays ten items at a time:

L 7-4
```
<select name="ControlName" size="10" multiple>
```

Be aware, however, that users must know how to CTRL-click to select multiple items from a menu. Naïve users will not know how to do this. Therefore, do not use the `multiple` attribute unless your site is for an experienced audience that will know how to make multiple selections from a menu.

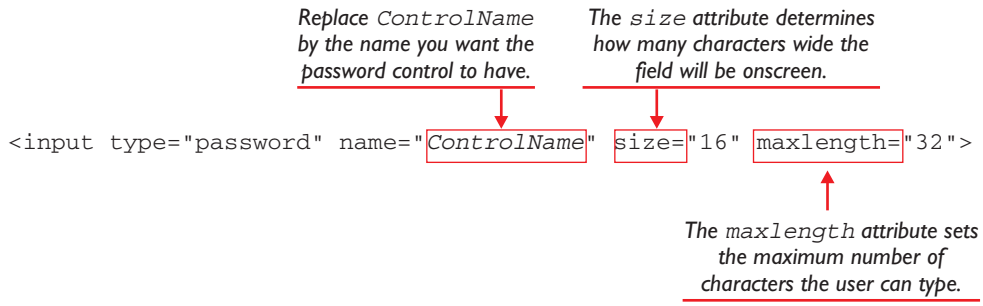
Creating Text Areas

A text area is an HTML form control that displays a scrolling entry box into which the user can type a larger amount of text than will fit in a text field. Thus, you use a text area when you need to provide space for the user to type more than a text field could easily hold. To create a text area, you use the **<textarea>** start and **</textarea>** stop tags. Between these two tags, you type any initial text you want displayed in the text area. If you want a blank text area, do not enter any text between these tags. To set the width, you use the *cols* attribute, which tells the browser how many characters wide to make the text box. To set the height of the text area, you use the *rows* attribute, which tells how many rows high to make the text box. The syntax is as follows:



Creating Password Fields

A **password field** is a text box that works like a text field except that when the user types the entry, the browser displays asterisks instead of revealing the sensitive information onscreen. The purpose is to prevent someone looking over the user's shoulder from seeing the password in clear text onscreen. The syntax is as follows:



HTML Web Form Control Summary

Table 7-2 summarizes the HTML Web form controls you have studied in this chapter. For more detailed information about these controls, go to www.w3.org/TR/html4 and follow the quick table of contents link to forms.

HTML Form Control	Purpose and Examples
Text field	Text entry box into which users can type characters onscreen. `<input type="text" name="ControlName" size="50" maxlength="150">`
Password field	Works like a text field except whatever the user types is displayed as asterisks onscreen. `<input type="password" name="ControlName" size="25" maxlength="50">`
Radio button	Displays a small round button that enables the user to select one, but not more than one, item from a list of possible choices. `<input type="radio" name="ControlName" value="daily" checked>Daily`
Submit button	Displays a button that, when clicked, causes the form to submit its data. By default, the button displays the text "Submit," which you can change via the `value` attribute. `<input type="submit" value="Subscribe">`
Reset button	Displays a button that, when clicked, clears the form and displays any default values of the form's controls. `<input type="reset">`
Check box	Displays a small box that the user can check to select one or more items onscreen. `<input type="checkbox" name="VariableName" value="VariableValue" checked>`

Single item selection menu	Displays a menu that lets the user select one option from a list of possible selection items.
	`<select name="ControlName" size="1">`
	`<option value="1"> Shoes`
	`<option value="2"> Socks`
	`<option value="3"> Pants`
	`</select>`

TABLE 7-2 *HTML Web Form Control Summary* ■

HTML Form Control	Purpose and Examples

Multiple item selection menu	Displays a menu that lets the user select more than one option from a list of possible selection items. Normally you size a multiple item selection menu to display all or most of the options onscreen.
	`<select name="ControlName" size="3" multiple>`
	`<option value="1"> Shoes`
	`<option value="2"> Socks`
	`<option value="3"> Pants`
	`</select>`

Text area	Displays a scrolling entry box into which the user can type a larger amount of text than will fit in a text field. `<textarea name="ControlName" rows="5" cols="70"> default text </textarea>`

TABLE 7-2 *HTML Web Form Control Summary* (continued) ■

Making a PayPal Buy Now Button

In Chapter 4, you learned about PayPal, a person-to-person payment system individuals can use to sell products over the Web without needing to set up a complicated e-commerce solution. Instead, you simply create an HTML form containing a PayPal button that, when clicked, posts data to the PayPal server indicating what product the user wants to buy from you. The PayPal system handles all the rest.

PayPal uses two special form techniques to power its button. First, PayPal uses hidden fields to identify the business that is selling the product, the name of the product being sold, and the selling price. Second, PayPal substitutes the PayPal button's image for the Submit button that normally appears on an HTML form. After explaining these two special form techniques, this chapter steps you through the process of creating a PayPal Web form.

note *Everyone who is working through this book can create the PayPal form that this tutorial shows you how to construct. To make the PayPal Buy Now button actually sell something, however, you need to have a PayPal account. Do not worry about this. The tutorial does not require you to have a PayPal account. The purpose of this exercise is to show you how to create a form that uses hidden fields and an image button. The tutorial does not require you to buy or sell anything online.*

Hidden Fields

A **hidden field** is an HTML form control that creates a variable name and a corresponding value that are not displayed onscreen but are posted along with the rest of the form data when the user clicks the button to submit the form. You create a hidden field by setting an `<input>` tag's `type` attribute to hidden. In the `name` and `value` attributes, you type the variable name and its corresponding value. The command syntax is

Replace `VariableName` with the name of the variable. *Replace `VariableValue` with the value of the variable.*

```
L 7-7   <input type="hidden" name="VariableName" value="VariableValue">
```

As you will see later in this tutorial, one of the hidden fields in a PayPal form identifies the business that is making the sale. In this particular field, the variable name is business, and the value is the business's e-mail address. The syntax is

Replace `me@mybusiness.com` with the e-mail address you registered at PayPal when you got your PayPal merchant account.

```
L 7-8   <input type="hidden" name="business" value="me@mybusiness.com">
```

The advantage of using a hidden field is that the form can send information to PayPal without cluttering the screen. Thus, you can design your page to present your product any way you want. The PayPal form displays only the PayPal button, which the user clicks to buy the product. All the other product information resides in hidden fields that do not appear onscreen.

Image Buttons

Although the default Submit button looks pretty good on a Web page, most people would agree that the customized PayPal button looks better. One of its advantages is that users can clearly see that clicking this button will trigger a PayPal transaction. To make such an image substitute for the default Submit button in a Web form, you use an **image button**, an HTML form element you create with an `<input>` tag of type image. The syntax is

Replace this by the complete http address of the image, including its filename extension, such as .gif or .jpg.

Set the border to 0 to prevent an unwanted box from getting printed around the button.

L 7-9
```
<input type="image" src="http://server/filename" border="0" name="submit">
```

Naming this image input "submit" makes it post the form data when clicked.

Coding a PayPal Buy Now Button

When clicked, the PayPal Buy Now button posts HTML form data to the PayPal e-commerce server. In addition to containing information about the product being purchased, the form data identifies the business that is making the sale. There are many options that can be present, plus a few that are always required. Table 7-3 lists the options that are always required and describes the form data that are typically used in a PayPal button.

Suppose you have a book for sale that is titled "Toymaking." The Web page on which you are advertising this book indicates that its sale price is $19.95. You have a PayPal merchant account registered under the e-mail address santa@northpole.com. To create a PayPal button to sell this product, you would follow these steps:

1. Click to position your cursor at the spot in the HTML code where you want the PayPal button to appear. Normally, you put it alongside or immediately under the product's description. Type the following tags to begin the form and identify your business. These tags must be present in every PayPal Buy Now form:

```
<form method="post" action="https://www.paypal.com/cgi-bin/webscr">
<input type="hidden" name="cmd" value="_xclick">
<input type="hidden" name="business" value="santa@northpole.com">
```

2. Having begun the form, you insert hidden fields to describe the product you have for sale. Type the following code, for example, to sell a book named "Toymaking" for $19.95:

```
<input type="hidden" name="item_name" value="Toymaking book">
<input type="hidden" name="amount" value="19.95">
```

To sell a different product, you change these value attributes. The rest of this code must always remain the same, including the names of these hidden fields.

3. Suppose you want to give the customer the option of getting an autographed copy of your book. When you ship the book, you will autograph it, if the user wants your autograph. To create such an option, type this code:

`on0` and `os0` are reserved names for the first of two possible options you can create. To create the other option, you use the names `on1` and `os1`.

```
<table><tr><td>
<input type="hidden" name="on0" value="Autographed?">Autographed?
<select name="os0"> <option value="Yes">Yes <option value="No">No
</select></td></tr></table>
```

4. The final step is to display the PayPal button and conclude the form tag. To do so, type this code:

```
<input type="image" src="http://www.paypal.com/en_US/i/btn/x-click-but01.gif"
 name="submit" alt="PayPal Buy Now Button">
</form>
```

The attribute `alt`, which stands for alternate, is an accessibility feature that specifies alternate text for this image. Chapter 9 is devoted to Web page accessibility.

Variable	Required?	Description
"post" action	Required	Must be set to "https://www.paypal.com/cgi-bin/webscr"
cmd	Required	Must be set to "_xclick"
business	Required	This is your PayPal ID or e-mail address where payments will be sent. This e-mail address must be confirmed and linked to your Verified Business or Premier PayPal account.
item_name	Optional	Description of item (maximum 127 characters). If omitted, the customer will see a field providing the option of entering an Item Name.
item_number	Optional	Pass-through variable for you to track payments. It will not be displayed to your customer but will be passed back to you at the completion of payment (maximum 127 characters). If omitted, no variable will be passed back.
amount	Optional	The price or amount of the purchase, not including shipping, handling, or tax. If omitted, the buyer will be able to adjust this value at the time of purchase.
return	Optional	An Internet URL where buyers will be returned after completing the payment, such as a Web page at your site thanking customers for their purchases. If the return is omitted, buyers will be taken to the PayPal site.
quantity	Optional	The quantity of items to be purchased. If omitted, this value will default to "1" and will not be shown in the payment flow.

TABLE 7-3 *Frequently Used PayPal Buy Now Button HTML and Hyperlink Variables* ■

Figure 7-5 shows the completed code from this exercise. Please remember that the PayPal site recommends that you always test each PayPal button you create to ensure that it works properly at your site. It is also possible that by the time you read this, some of the PayPal instructions may have changed. For the latest button-making instructions, follow this book's Web site link to PayPal's Buy Now Button Manual.

note *These are only a few of the possible form fields documented in PayPal's Buy Now Button Manual. To see the rest of the options that are available, follow this book's Web site link to PayPal's Buy Now Button Manual.*

Using the PayPal Button Factory

For users who are not programmatically inclined, PayPal has a button factory that generates automatically the HTML to paste onto your page to sell items at your Web site. To use the PayPal button factory, follow these steps:

1. Log on to your PayPal account at www.paypal.com and click the Merchant Tools tab. Under the Website Payments heading, follow the link to Buy Now Buttons to make the Button Factory appear.
2. The Button Factory prompts you to fill in some details about the product you want to sell. After you fill in these details, click the link labeled Create Button Now. In a text box titled "For Web pages," the button factory displays the HTML code to paste into your Web page.
3. Use your mouse to select all the HTML code in the text box, and then press CTRL-C to copy the code. Then click to position your cursor at the spot in your Notepad editor where you want the button, and press CTRL-V to paste the code. Normally, you paste this code alongside or immediately under the description of the product or service you are selling.

For more information about using the PayPal Buy Now Button on your Web site, follow this book's Web site link to PayPal's Buy Now Button Manual.

FIGURE 7-5 *The HTML code of a typical PayPal Buy Now button. Although this example is fully functional, it illustrates only a few of the PayPal fields you can put into an HTML form. To see the rest of the available form field options, follow this book's Web site link to PayPal's Buy Now Button Manual.* ■

Designing HTML Image Maps

An **image map** is an invisible layer of triggers placed over an image onscreen. The triggers can be rectangular, circular, or polygonal. When the user clicks inside one of the invisible triggers, the Web browser triggers the object of the link.

In earlier versions of HTML, using image maps required the use of a CGI call. When the user clicked a trigger in an image map, the browser sent the coordinates of the mouse click to a CGI program on a server, which told the browser what to do in response to the click. In the latest versions of HTML, it is possible to process image maps locally within the document instead of having to call on a CGI program for help in handling the mouse click.

Analyzing a Client-Side Image Map

One of the best ways of understanding image maps is to analyze a working example of an image map onscreen. Consider the example in Figure 7-6, which displays a piano keyboard on which the user is asked to click Middle C. If the user finds Middle C correctly, the user will be rewarded. If not, the user will be provided with an appropriate feedback message and given a chance to try again. To try this example yourself, go to this book's Web site and click the music keyboard icon.

Defining the Map and Area Tags

Figure 7-7 shows the HTML code that asks the question presented in Figure 7-6. The **<map>** and **</map>** tags demarcate the beginning and ending of the image map. Between them, you see the **<area>** tags that create the triggers. Inside each area tag is a *shape* attribute and a *coord* attribute that specifies the x,y coordinates of the links. These coordinates are pixel addresses inside the image. The top-left corner of an image is always pixel address 0,0 and is called the origin. The other addresses are all relative to the origin. The coord attribute has the syntax *coords= x1,y1,x2,y2*. The top-left corner of the area is x1,y1, and the bottom-right corner is x2,y2.

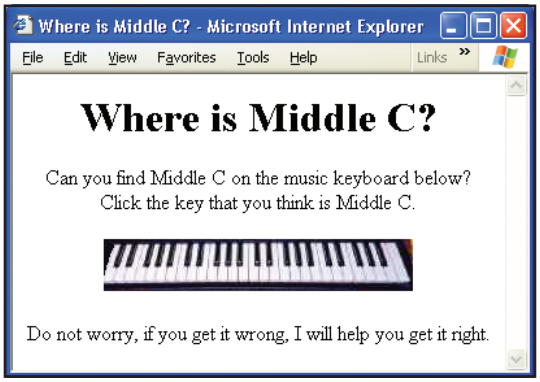


FIGURE 7-6 *This exercise asks the user to click Middle C on the music keyboard that is displayed onscreen. An invisible image map creates triggers for the correct answer and all the possible wrong answers. Depending on where the user clicks, these triggers bring up appropriate feedback screens, which are linked to the triggers in the image map. To try this example yourself, go to this book's Web site and click the music keyboard icon.* ■

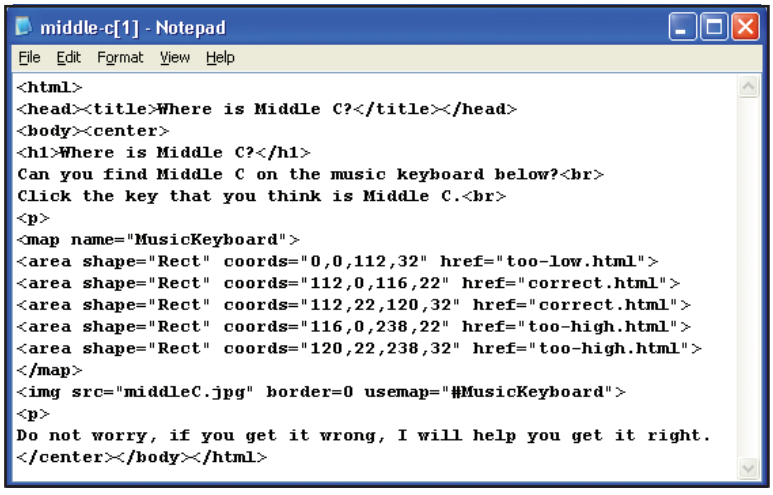


FIGURE 7-7 *The HTML code that presents the Middle C question.* ■

Visualizing the Coordinates

The Middle C image map contains five sets of coordinates. Figure 7-8 helps you visualize how these coordinates fit the too-low, too-high, and just-right regions of the music keyboard image.

The first set of coordinates has the HTML encoding of `coords=0,0,112,32`. The format of these coordinates is x1,y1,x2,y2. Therefore, they define an area that goes from 0,0 to 112,32, which is the portion of the keyboard below middle C. When the user clicks there, the too-low link gets triggered.

The second area has the coordinates 112,0,116,22, thereby specifying a rectangle from 112,0 to 116,22, which is where the top half of Middle C is in the image. The third area has the coordinates 112,22,120,32, which the bottom part of middle C comprises. The fourth and fifth areas define the "too high" regions to the right of Middle C.

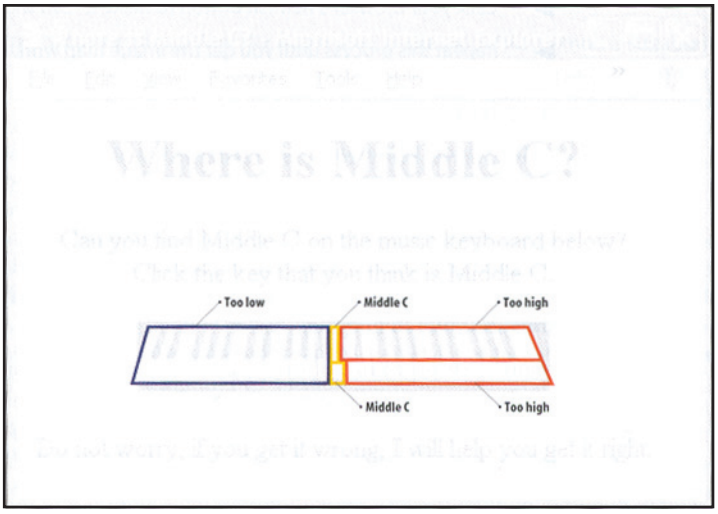


FIGURE 7-8 *Visualization of how the coordinates of the MusicKeyboard image map describe the too-low, just-right, and too-high regions of the music keyboard image.* ■

Applying the Usemap Attribute

Once you have defined an image map with the `<map>` tag, you use the *`usemap`* attribute to apply it to any image on the page. The `usemap` attribute modifies the `<img>` tag that puts the image onscreen. Notice that the image tag in Figure 7-7 has a `usemap` attribute telling your browser to use the MusicKeyboard map on the middleC.jpg image:

L 7-10
```
<img src="middleC.jpg" border=0 usemap="#MusicKeyboard">
```

The syntax of the `usemap` attribute is usemap=#*area_name* where *area_name* is the name you gave the image map in the `<map>` tag's `name` attribute. When the mouse clicks on one of the areas in the image map, the user follows the corresponding link.

Creating Nonrectangular Areas

In the Middle C example, the regions in the image map are all rectangular. Image maps can also contain circle areas. If you wanted a volleyball to be a trigger, for example, you could make a circle area to do that. The HTML syntax for a circle area is as follows:

L 7-11
```
<area shape="circle" coords="center-x,center-y,radius" href="your_link.html">
```

Replace `center-x`, `center-y`, and `radius` with integer values defining the center and size of your circle.

You can even create polygonal areas with any conceivable shape. On a U.S. map, if you wanted to make the state of Florida trigger something, you could define a polygon the shape of Florida. There is no limit to the number of triggers or their shape. You can literally create any kind of triggering situation you can imagine. The HTML syntax for a polygonal area is

L 7-12
```
<area shape="poly" coords="x1,y1,x2,y2,x3,y3,x4,y4" href="your_link.html">
```

Replace *x1,y1,x2,y2,x3,y3,x4,y4* with integer values defining the coordinates of each successive endpoint along the polygon. Specify one coordinate for each endpoint. Keep going if there are more than the three sides in this example. The last pair of coordinates should be the same as the first, to close the polygon.

Many Web page creation tools have image map assistants built in that can help you create image maps. Or you can use a tool such as Paint Shop Pro to figure out pixel locations. To learn more about image maps and tools for creating them, follow this book's Web site link to the image map tutorial.

Try This!

Roll Your Own Image Map

Because client-side image maps execute in the browser, you can create an image map without needing access to a CGI server. With all of the processing done on your computer, you do not even need to be connected to the Internet. All you need is a browser and a text editor, such as the Notepad. To roll your own image map, follow these steps:

1. *Obtain an image*. In Chapter 6, you learned how to capture and convert images into a format suitable for displaying on a Web page. Following that process, obtain an image on which you would like to create two or three rectangular triggers. Save the image in your *website* folder. Remember that the image must have a *.gif*, *.jpg*, or *.png* filename extension.
2. *Plot the coordinates*. Using your graphics editor, figure out the x,y coordinates of the upper-left and lower-right corners of each trigger you want to make. Most graphics editors will tell you the x,y coordinates as you mouse over the image. Paint Shop Pro, for example, displays the coordinates in the lower-left corner of the status bar as you mouse over the image. Remember that the upper-left corner of the image is coordinate 0,0.
3. *Code the map*. Use the Notepad to start a new file, type the following HTML code, and save the file in your *website* folder. Give the file an *.html* filename extension, such as **MyFirstImageMap.html**. When you type this code, substitute your actual coordinates and links for the placeholder values that are italicized in this code sample, which links to your résumé and home page by default:

```
<html>
<head>
  <title>My First Image Map</title>
</head>
<body>
<map name="MyFirstMap">
<area shape="Rect" coords="0,0,50,50" href="resume.html">
<area shape="Rect" coords="0,51,99,99" href="index.html">
</map>
</body>
</html>
```

Try This! *continued*

4. *Use the map*. After you create the HTML code for your image map, you can use it on any image on your Web page. In this example, you apply it to the image you selected in step 1. Click to position your cursor immediately prior to the `</body>` tag that ends the page you created in the previous step. Type the following code, replacing the italicized placeholder values with your actual image filename, width, and height attributes:

```
<img src="MyImage.jpg" border=0 usemap="#MyFirstMap" width="100" height="100">
```

5. *Test the map*. Save the file and open it with your browser. Mouse over the image and see if the cursor changes shape when you mouse over the hot spots. Click to try those links. If there are any problems, use the Notepad to make the necessary changes in your HTML, save the file again, and test it in your browser. Click your browser's Refresh button to make it read the new version of the file. Repeat this step until you get the image map working, and then congratulate yourself, because image maps enable you to create graphically rich user interfaces onscreen.

Creating HTML Frames

So far in this tutorial, you have created Web pages that appear individually when viewed with a browser onscreen. Because this is the simplest way to display a Web page, most Web sites present pages one at a time. In more complex designs, however, it is possible to divide the browser's display surface into multiple windows called *frames*. Each **frame** can display a different part of the same page or a totally different page. Through a technique called **frame targeting**, you can make clicks in one frame determine what gets displayed in another. Clicks on menu items in a left sidebar frame, for example, can make the chosen content appear in a main content frame.

This tutorial begins by having you create a simple frameset that will display different views of your Web page résumé. By trying out different layouts, you gain experience with the parameters you can use to split the screen into any frame design you can imagine. In practice, however, the most common use of frames is to create a narrow sidebar frame in which users can select menu items to display the desired document in a main content frame that appears alongside the menu. This tutorial concludes, therefore, by providing you with a step-by-step template you can use to set up this kind of sidebar frameset.

What Is a Frameset?

A **frameset** is an HTML Web page element that splits the browser window into two or more subdocuments called frames. You create a frameset with the **`<frameset>`** start and **`</frameset>`** stop tags. The start tag has `row` and `column` attributes that determine the layout. You specify the dimensions by pixels or percentages. Imagine that you want to create a

frameset that splits the browser window into two equal frames. Follow these steps:

1. Pull down the Notepad's File menu and choose New to create a new file. Type the following code, which creates the frameset. In this code, you will notice that the `<body>` start and `</body>` stop tags are surrounded by the `<noframes>` start and `</noframes>` stop tags. That is because the bodies of the frames will be in the HTML source documents that flow into the frames. Only users who do not have frames will see this particular body, which tells the users to update their browser:

```
<html>
<head>
<title>Frameset Example</title>
</head>
<frameset rows="50%,50%">
  <frame src="resume.html">
  <frame src="resume.html">
<noframes>
<body>
<p>
This page uses frames. Either update your browser, or
<a href="resume.html">view this without frames</a>.
</p>
</body>
</noframes>
</frameset>
</html>
```

2. Save the file in your *website* folder under the name *frameset.html*. Then open this file with your browser to see how it appears onscreen. If you typed the frameset code correctly, you will see that the browser divides the screen into the two frames created by your `<frameset>` tag. If you do not see these frames, go back to the previous step and correct any typographical errors in your code. As you work to troubleshoot any problems, remember to click the browser's Refresh button to make sure you are viewing the latest version of the file.

Figure 7-9 shows that the browser obeys your frameset tag by dividing the window into two equal frames. In the next part of this tutorial, you learn how to create other frame layouts. Read on.

Creating Frameset Layouts

With framesets, you can create any number of frames with any onscreen layout you can imagine. Learning how to do this is fun. That is why the following examples are designed for you to learn by doing. In each example, the `<frameset>` code appears first, followed by an illustration that diagrams its effect onscreen. If you want to make these effects hap-

pen on your own screen, simply modify the *frameset.html* example you created earlier in this chapter. The only code you need to change is the `<frameset>` code. All the rest of the *frameset.html* file remains the same.

Vertical Frames

Vertical frames divide the browser window into side-by-side columns onscreen. You create vertical frames via the `cols` attribute. You can easily modify the *frameset.html* example to display the frames vertically by typing **cols="50%,50%"** in place of the *rows* setting. Figure 7-10 shows that this modification creates two vertical frames that divide the screen into two equal columns.

To reinforce this concept, you should actually try this example via your Notepad and browser instead of just looking at the figure. Use the Notepad to open your *frameset.html* file and modify the frameset code to read as follows:

L 7-13

```
<frameset cols="50%,50%">
 <frame src="resume.html">
 <frame src="resume.html">
</frameset>
```

Save the file and open it with your browser. Click the browser's Refresh button to make sure you are viewing the latest version of the file. Use the scrollbars to view different parts of your résumé simultaneously.

Horizontal Frames

Horizontal frames divide the browser window into rows that appear in order from top to bottom onscreen. You create horizontal frames via the `rows` attribute. You can create as many rows as you

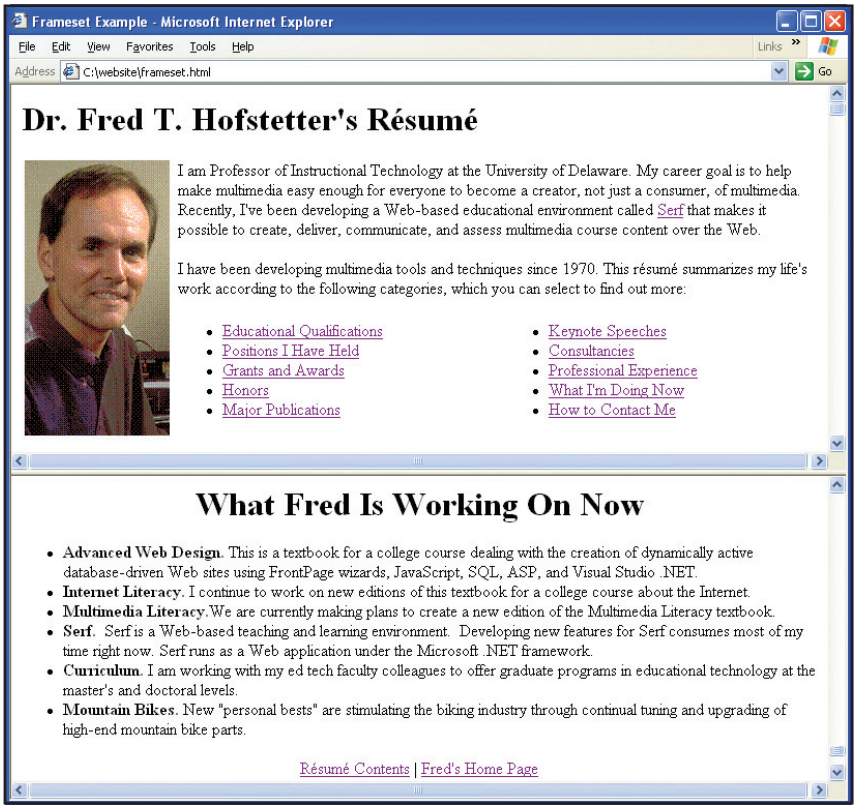


FIGURE 7-9 *The result of creating a frameset with rows="50%,50%", which divides the browser window into two equal horizontal frames. In this example, each frame is displaying the same document, which is a Web page résumé. Notice that each frame has a scrollbar you can use to scroll to different parts of the document.* ■

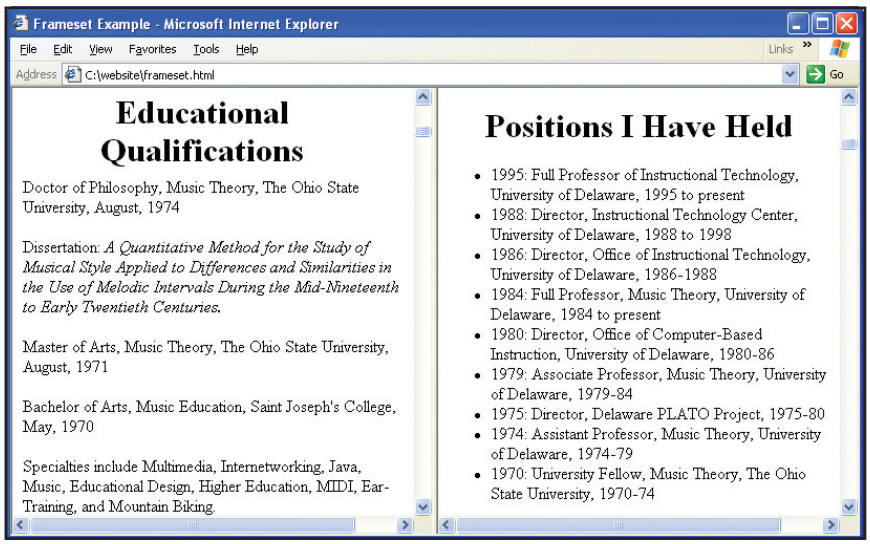


FIGURE 7-10 *You create vertical frames with the cols attribute. In this example, the attribute cols="50%,50%" is dividing the browser's window into two equal vertical frames.* ■

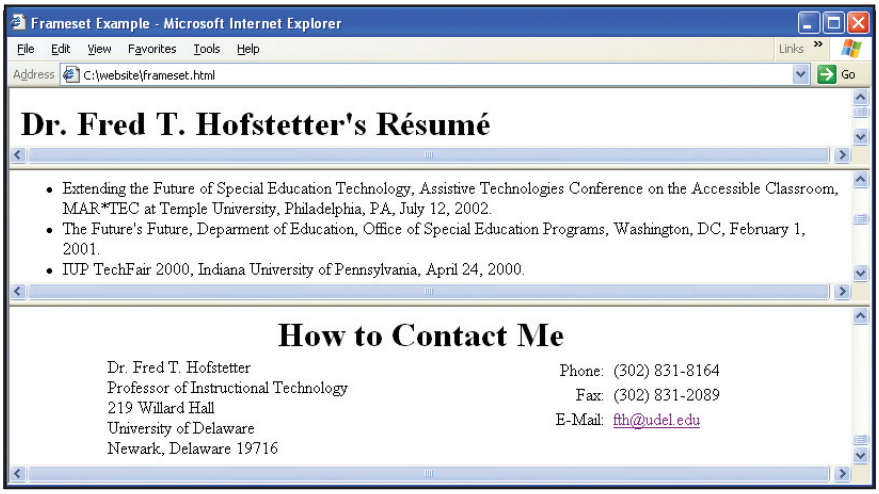


FIGURE 7-11 *You create horizontal frames with the rows attribute. In this example, the attribute rows="20%,30%,50%" creates three frames onscreen.* ■

want, but you should not go overboard. Imagine that you want three rows to be displayed onscreen. You want the top row to take up 20% of the screen, the middle row 30%, and the bottom row 50%. You can accomplish this with a `rows="20%,30%,50%"` attribute. Figure 7-11 shows the frames this creates in the browser window.

To reinforce this concept, you should actually try this example yourself by modifying the frameset section of your *frameset.html* file to read as follows:

```
<frameset rows="20%,30%,50%">
<frame src="resume.html">
<frame src="resume.html">
<frame src="resume.html">
</frameset>
```

Frameset Grids

A frameset grid is a layout in which there are both vertical and horizontal frames onscreen. You create a frameset grid by using both the `rows` and `cols` attributes in a frameset tag. Figure 7-12 shows an example that divides the screen into a grid of four frames. Here is the frameset code that created this example:

```
<frameset rows="50%,50%" cols="50%,50%">
<frame src="resume.html">
<frame src="resume.html">
<frame src="resume.html">
<frame src="resume.html">
</frameset>
```

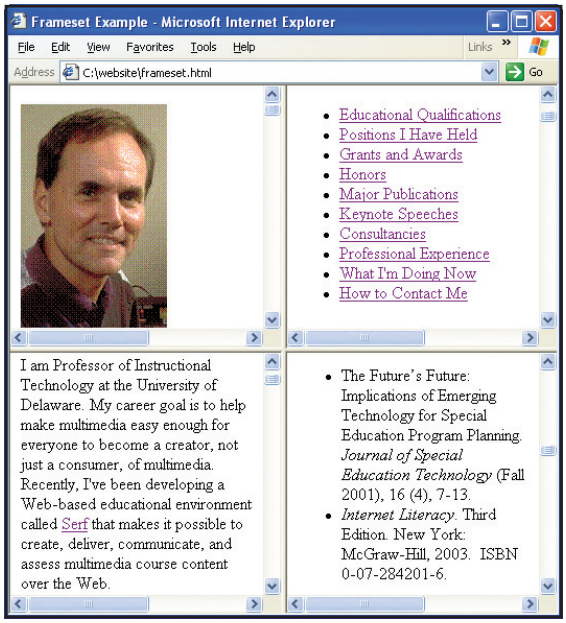


FIGURE 7-12 *You create a grid of both horizontal and vertical frames by using both the rows and cols attributes. In this example, the rows="50%,50%" and cols="50%,50%" attributes are creating a grid of four equal frames onscreen. You need to be careful, however, when creating frameset grids. Too many frames can make your page look like frames gone wild.* ■

Nested Framesets

As you just observed, creating a frameset that has both `rows` and `cols` attributes creates a grid of symmetrical frames onscreen. To divide one of the rows or columns, you must divide them all. In practice, you do not normally want all the frames to be divided. Instead, you would normally want to divide just one frame into subframes. To divide a single frame into subframes, you create a frameset inside that frame. A frameset that you create inside of another frame is called a **nested frameset**.

The following code creates a nested frameset. As illustrated in Figure 7-13, this code begins by creating a frameset that uses rows="20%,80%" to divide the screen into two horizontal frames. Then a nested frameset uses cols="35%,65%" to divide the larger of these two frames into two vertical frames:

The outer frameset creates two horizontal frames that occupy 20% and 80% of the window.

L 7-16

```
<frameset rows="20%,80%">
<frame src="resume.html">
  <frameset cols="35%,65%">
   <frame src="resume.html">
   <frame src="resume.html">
  </frameset>
</frameset>
```

The inner frameset creates two vertical frames that occupy 35% and 65% of the window.

Borderless Frames

Now that you know how to create a frameset with any layout that you can imagine, you may be wondering how to turn off the frame borders for situations in which you would like a seamless transition between the frames. In Figure 7-13, for example, the narrow frame at the top may be intended to hold a constant page heading or graphical banner for which the scrollbars have no meaning and may interfere with the visual effect you are trying

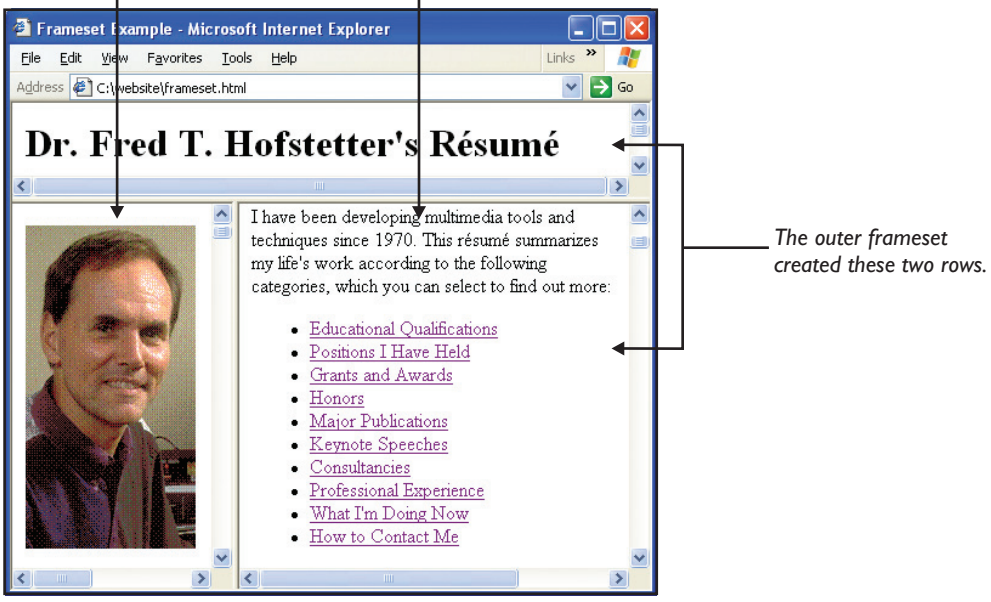


FIGURE 7-13 *A nested frameset is one that you create inside of another frameset. In this example, an outer frameset creates two rows onscreen that occupy 20% and 80% of the browser's window, respectively. Inside the 80% row, a nested frameset creates two columns that occupy 35% and 65% of the frame. By nesting framesets, you can create any layout you can imagine. In the next part of this tutorial, you learn how to turn the borders invisible for situations in which you do not want the frame's outline to show.* ■

to achieve onscreen. To remove the border around a frameset, you add the following attributes to the frameset tag:

L 7-17
```
frameborder="no" border="0" framespacing="0"
```

To remove the scrollbar from a frame, you add the following attribute to its frame tag:

L 7-18
```
scrolling="no"
```

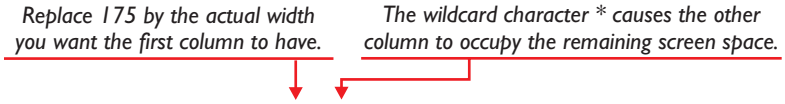
To prevent users from being able to resize a frame, you add the *noresize* attribute to the frame tag. To see all of these settings in action, you can modify your *frameset.html* code to remove the border and the scrollbar from the top row as follows:

L 7-19
```
<frameset rows="20%,80%" frameborder="no" border="0" framespacing="0">
<frame src="resume.html" scrolling="no" noresize>
  <frameset cols="35%,65%">
   <frame src="resume.html">
   <frame src="resume.html">
  </frameset>
</frameset>
```

Fixed-Size Frames

As you have undoubtedly observed by now, when you use percentages to set the dimensions in the `row` and `column` attributes of a frame tag, the frame automatically adjusts in size to maintain those percentages as the user resizes the browser window. Sometimes you may want to fix the size of a frame to a certain pixel width or height instead of letting the frame resize. If you are creating a left sidebar, for example, you may want to set it to a fixed pixel size and let the main content frame fill the rest of the screen.

Suppose you want a sidebar frame that has 175 pixels and a main content frame that occupies the rest of the browser window. The following frameset would create such a layout. Notice that the wildcard character * tells the browser to create a frame consisting of all the window space that remains after creating the 175 pixel sidebar:

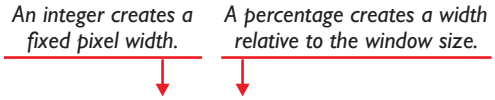


L 7-20
```
<frameset cols="175,*">
       <frame src="leftSidebar.html">
       <frame src="mainContent.html">
</frameset>
```

Wildcard Sized Frames

You can get pretty creative with the * wildcard character. The following frameset, for example, would create a layout that has three columns. The first column has a fixed width of 175 pixels. The third column occupies

10% of the window. The * wildcard character makes the middle column occupy the rest of the window:



L 7-21
```
<frameset cols="175,*,10%">
        <frame src="leftSidebar.html">
        <frame src="mainContent.html">
        <frame src="rightSidebar.html">
</frameset>
```

You can use the * wildcard character to size frames proportionally. The following frameset, for example, creates a layout in which the second column is three times as wide as each of the first and third columns:

L 7-22
```
<frameset cols="*,3*,*">
        <frame src="leftSidebar.html">
        <frame src="mainContent.html">
        <frame src="rightSidebar.html">
</frameset>
```

3 causes the middle frame to be three times as wide as the single * in the first and third frames.*

Frame Targeting

One of the most useful applications of a frameset is to establish a *targeting* relationship such that a mouse click in one frame can alter the display of a document in another frame. This is how you make hyperlinks in a left sidebar frame, for example, display their documents in the targeted main frame. Establishing such a targeting relationship is a two-step process. First, in the frameset, you use the `name` attribute to give the targeted frame a name. Second, in the document containing the hyperlinks, you use the `target` attribute to make the links come up in the targeted frame.

Naming the targeted frame in a frameset is the easy part. You do that simply by adding a `name` attribute to the targeted frame. Here is an example that names the targeted frame "mainContent" so items clicked in the left sidebar can target the main content frame:

L 7-23
```
<frameset cols="20%,80%">
 <frame src="sidebar.html">
 <frame src="resume.html" name="mainContent">
</frameset>
```

After you name the frame that the hyperlinks will target, you use the *target* tag to make the hyperlinks aim at the targeted frame. Here is an example, which you would put on the sidebar page:

L 7-24
```
<a href="resume.html#education" target="mainContent">
Educational Qualifications</a>
```

When clicked, such a hyperlink will bring up the educational qualifications section of your résumé in the mainContent frame, if it exists. If the mainContent frame does not exist, the browser will create a new window named "mainContent" to contain this content.

Using the Base Tag

You can avoid typing the `target` attribute in each hyperlink if you use the **`<base>`** tag to specify the default target of the sidebar links. The `<base>` tag goes into the `<head>` section of the document containing the hyperlinks. The format is:

L 7-25
```
<base target="your_target">
```

Replace *your_target* with the name of the targeted frame. By typing this single command into the `<head>` of the document, you make every hyperlink in the document inherit this target by default. To make every hyperlink in the document target by default the mainContent frame, for example, you would type the following tag:

L 7-26
```
<base target="mainContent">
```

You can override this default by hard-coding a different target on any hyperlink that you want pointing somewhere else.

The `<base>` tag also has a `URL` attribute in which you can specify the HTTP address of the targeted document. To make Santa Claus's résumé become the base document for the hyperlinks, for example, you could make the `<base>` tag read as follows:

L 7-27
```
<base url="http://www.northpole.com/santa/resume.html" target="mainContent">
```

In an anchor tag in that same document, this would make `href="#education"` point to http://www.northpole.com/santa/resume.html#education.

Try This!

Creating a Sidebar Frameset for Your Résumé

The best way to understand the concept of frame targeting is to work through an example. The *resume.html* file you created in the previous chapter provides an excellent subject, because your résumé contains a bulleted table of contents. Imagine using a frameset that divides your résumé into two vertical frames. On the left is a narrow sidebar consisting of your résumé's table of contents. On the right of the sidebar is a main content frame that displays the part of your résumé the user selects in the sidebar menu. The advantage of this two-sided approach is that the menu always remains onscreen in the sidebar. This enables users to select different parts of your résumé more quickly than when a sidebar is not present. To create a sidebar frameset for your résumé, follow these steps:

1. Pull down the Notepad's File menu and choose New to create a new file. Into this new file, type the code that is listed in Figure 7-14, and save it in your *website* folder as *sidebarframeset.html*. To save time, you can download this code by following this book's Web site link to Sidebar Frameset.
2. Pull down the Notepad's File menu and choose New to create a new file. Into this new file, type the code listed in Figure 7-15. The body of this file consists of the bulleted table of contents from my résumé. As instructed by the figure's callout, modify these items to fit your own résumé's contents. Then save the file in your *website* folder as *sidebar.html*.

Try This! continued

```
sidebarframeset.html - Notepad
File Edit Format View Help
<html>
<head>
<title>Sidebar Frameset</title>
</head>
<frameset cols="175,*">
  <frame src="sidebar.html" target="mainContent">
  <frame src="resume.html" name="mainContent">
<noframes>
  <body>
This page uses frames. Either update your browser, or
<a href="resume.html">view this without frames</a>.
  </body>
</noframes>
</frameset>
</html>
```

FIGURE 7-14 *Type this code to create the sidebar frameset used in the Try This! exercise. The target="mainContent" parameter makes the links clicked in the sidebar come up in the main content frame. Notice that the <noframes> section of this code provides users with an option to view the main content without frames. You can download this code by following this book's Web site link to Sidebar Frameset.* ■

This base tag makes the links in this file target the frame named mainContent.

The photo is optional. If you have a photo sized to fit inside the sidebar, use an <img> tag to display the image here.

```
sidebar.html - Notepad
File Edit Format View Help
<html>
<head>
<title>Resume Menu</title>
<base target="mainContent">
</head>
<body>
<img src="photo.gif">
<ul>
<li><a href="resume.html#qualifications">Education</a></li>
<li><a href="resume.html#positions">Positions</a></li>
<li><a href="resume.html#grants">Grants</a></li>
<li><a href="resume.html#honors">Honors</a></li>
<li><a href="resume.html#publications">Publications</a></li>
<li><a href="resume.html#keynotes">Speeches</a></li>
<li><a href="resume.html#consultancies">Consultancies</a></li>
<li><a href="resume.html#professional">Experience</a></li>
<li><a href="resume.html#current">Working On</a></li>
<li><a href="resume.html#contact">Contact Me</a></li>
</ul>
</body>
</html>
```

FIGURE 7-15 *This is the sidebar.html file, into which you type the links your Web page résumé's Table of Contents comprises. The links in this example are from my résumé. You need to replace these with the links you decided to put into your own Web page résumé.* ■

3. Use your browser to open the *sidebarframeset.html* file. As illustrated in Figure 7-16, you should see a frameset in which your *sidebar.html* file appears in the left frame and your *resume.html* file appears to its right. If this is not what you see onscreen, go back to step 1 and follow these instructions more carefully to troubleshoot any problems.

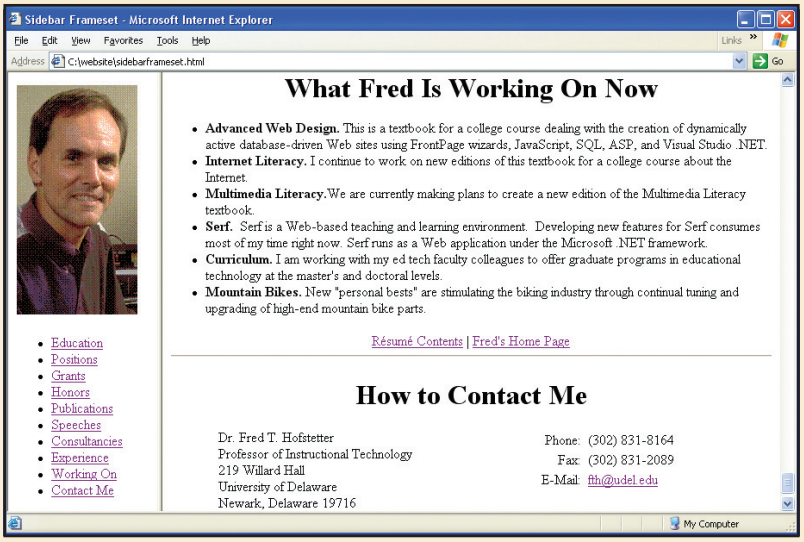


FIGURE 7-16 *When you open the sidebarframeset.html file with a browser, you get a frameset consisting of two frames. On the left, a sidebar displays a bulleted list of hyperlinks. When the user clicks a hyperlink, the requested information appears in the main content frame on the right. In this example, the user clicked the Working On link to display what I am working on now.* ■

4. To test the links in the sidebar, click each item in the left frame to see if it brings up the corresponding content in the main content frame on the right. If any of the links fail, use the Notepad to troubleshoot the problem by correcting the errant anchor tag in the *sidebar.html* file. If all of the links are failing, make sure you typed the <base> tag correctly in the <head> section of the *sidebar.html* file, as illustrated in Figure 7-15.
5. When you get this working, congratulate yourself heartily, because you have created a working frameset you can use as a model whenever you need to create a sidebar frame.

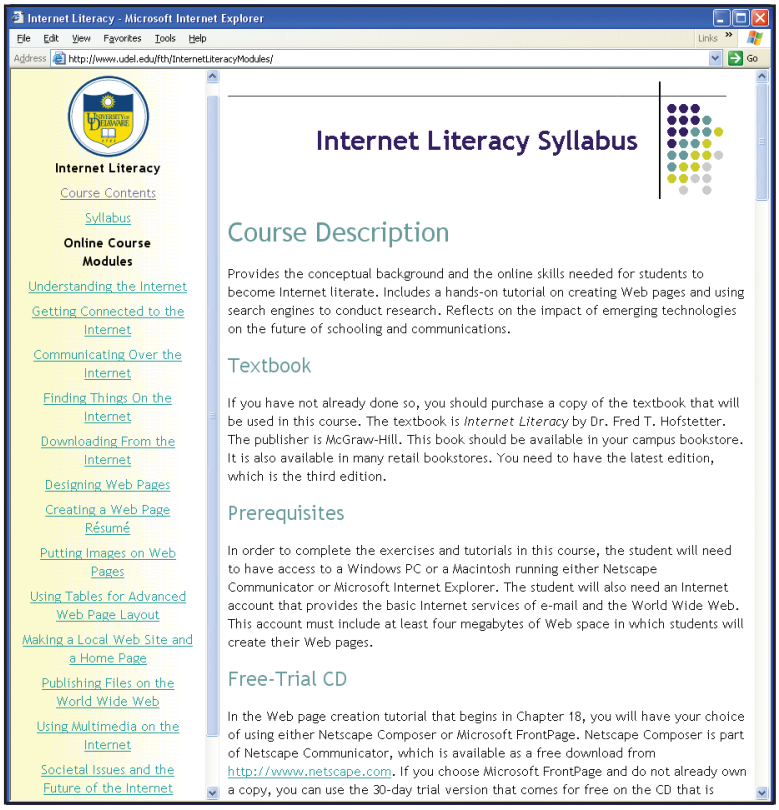


FIGURE 7-17 *The frameset version of the Internet Literacy course has a left sidebar frame that lets users click to bring up modules in the main content frame. Compare this to the version without frames in Figure 7-18. To try the frameset online, go to www.udel.edu/fth/InternetLiteracyModules.* ■

When Should You Use or Not Use a Frameset?

I am not a huge fan of framesets. Although frames provide an easy way to create sidebars, multiple windows can be confusing, especially for users with special needs. For this reason, I use framesets only when required to do so. In such cases, I design the content in such a way that users who do not want frames can navigate without them. Consider, for example, the frameset illustrated in Figure 7-17. This is a frameset version of my online course on Internet Literacy. This online course is published by PBS, which wanted a frameset version that could snap in to frames-based instructional management systems. I made PBS happy by creating the frameset version. At the same time, I designed the content in such a way that users who do not want frames can navigate without them. Figure 7-18 shows how the non-frames version provides navigation options that enable the user to move forward or backward and drill down or up through the course content without the aid of frames.

Still, there are applications in which framesets are cool. I like the frameset that PowerPoint creates when you choose the option to save a PowerPoint show as a Web page. The HTML Translator section of Chapter 5, for example, illustrates the PowerPoint frameset in Figure 5-3. There is no quicker way to create a frameset so rich in hypermedia and navigational functions.

Publishing a Web Site via FTP

To publish a file on the World Wide Web means to transfer the file into a folder on a Web server so other people around the world can access the file with a Web browser. Unless your computer happens to be a Web server where you could save the files directly, you need a way to transfer your files to the Web. This tutorial provides you with the knowledge and the tools needed to transfer files from your computer to a World Wide Web file server.

As you work through this tutorial, you publish your home page and your résumé on the Web. Then you can provide access to your Web pages by telling people what URL to go to. For example, suppose your Web server is www.northpole.com and your Web site is located on that server in a Web account named santa. Assume further that the default filename on your server is *index.html*. After you complete the exercises in this chapter, the URL of your home page will be http://www.northpole.com/~santa/index.html.

Because *index.html* is the default filename, you will be able to shorten the URL and tell users to go to http://www.northpole.com/~santa to see your home page.

If *index.html* is not the default filename on your Web server, you use your Web server's default filename instead. If you do not know what the default filename is on your Web server, you should check the guidelines at your Web hosting site or contact your server administrator to find out. If your Web server does not have a default filename, you can give your home page an intuitive name such as *home.html*.

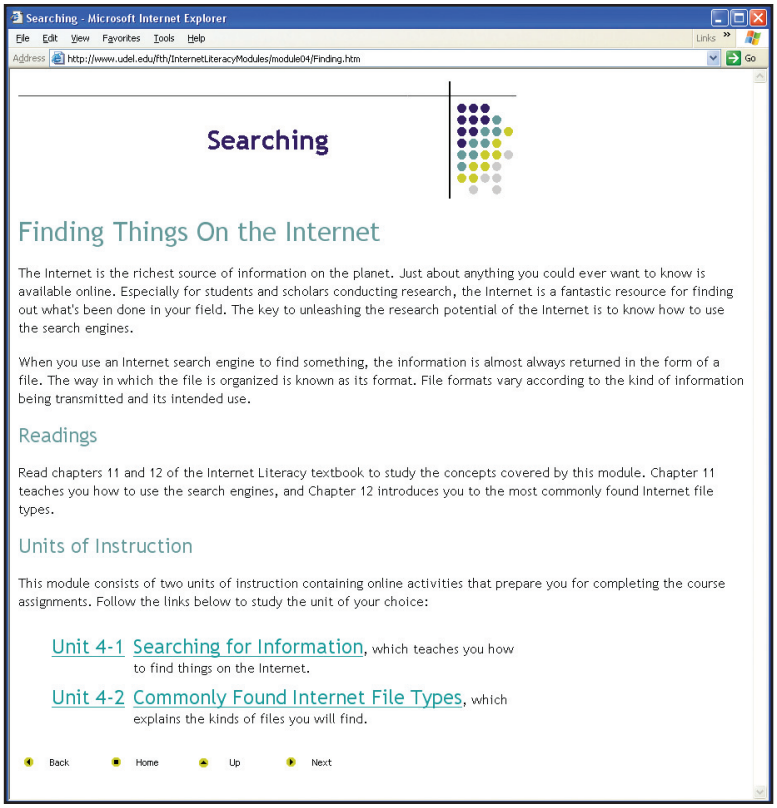


FIGURE 7-18 *The single-page version of the Internet Literacy course lets users navigate the main content minus the frameset illustrated in Figure 7-17. To surf the content without the frameset, go to www.udel.edu/fth/InternetLiteracyModules/mainContent.html.* ■

Getting Your Web Space

Before you can publish a file on the World Wide Web, you need to have some file space on the Web to hold your published files. There are three places where you can get file space on the Web. First, your ISP account probably includes a certain allotment of Web space. If you are not sure, check with your ISP. While you are checking, you might want to inquire as to what the limit is and how much it costs if you want to get more space. Second, your school or workplace may have a Web server on which you can obtain Web space. Check with your supervisor or IT staff to find out the policies for obtaining and using Web space at your school or workplace. Third, there are sites on the Web where you can get free Web space in return for registering your name at those sites. They give you the free space to keep you coming back to their sites, where you will see commercial advertising that pays for the free space. The free space may also cause ads to be placed on your Web pages, although some sites offer free Web space without putting ads on your Web pages.

Searching Google or Yahoo for the keywords "free FTP Web space" brings up many sites where you can get free FTP Web space.

Your Web Space Address

To transfer files to the Web, you need to know the name of your Web server and the path to your file space on that server. If you are using free Web space, you will be given the name of the server and the path to your file space when you register and get your free Web space. Make a note of this information, because it is very important. If you are using Web space from an ISP and you do not know the address of your Web space, contact your ISP and ask the question, "What is the FTP address of my Web space?" If Santa Claus had an AOL account named Santa, for example, and he asked AOL for the FTP address of his Web space, the answer would be something like **ftp://ftp.aol.com/santa**.

Choosing an FTP Client

Searching Google or Yahoo for the keywords "free ftp client" brings up many FTP clients that you can use to publish a Web. It is important for you to choose a client that has a good track record. This chapter's Web publishing tutorial teaches you the Core FTP Lite client that many students use because it is free for educational purposes and is easy to use. The Core FTP company also makes a professional edition called Core FTP Pro that is for commercial purposes. When this book went to press, a single-user Pro license cost $29.95. The following tutorial uses the free version called Core FTP Lite.

Installing the FTP Software

To download and install the free-trial version of the Core FTP Lite software, follow these steps:

1. Click the link at this book's Web site to download Core FTP Lite.
2. When the download page appears, click the link to download the free version of Core FTP Lite. When your browser asks what folder you want to download the file into, choose the folder you want to put it in. Normally, you use your *temp* folder.
3. When the file has finished downloading, click your computer's Start button, choose Run, and use the Run dialog's Browse control to locate the Core FTP Lite installation program. You will find this program in the folder where you saved the downloaded file in the previous step. Click the Run dialog's OK button to run the Core FTP Lite installation program.
4. The installation program guides you through the setup; follow the onscreen instructions.

How to Configure a New FTP Connection

The first time you use the Core FTP Lite program, you need to configure a new FTP connection. This is the connection through which the program will FTP the files that you want to publish to your Web site. To create a new FTP connection, follow these steps:

1. Double-click the Core FTP Lite icon on your Windows desktop to get Core FTP Lite running. You can also start Core FTP Lite by clicking the Windows Start button and choosing Programs | Core FTP | Core FTP.
2. The first time you run Core FTP Lite, it will ask if you want Core FTP to become your computer's default FTP program. I answered Yes to make Core FTP become the default FTP handler on my computer.
3. When the Core FTP window opens, it will display the Site Manager window, which you use to create a new FTP connection. Figure 7-19 shows the settings in the Site Manager window.

4. In the Site Name field, type the name of your site. When you make up this name, enter information that clearly identifies your site. Santa Claus, for example, could call his site Santa's North Pole Web Site.
5. In the Host/IP/URL field, type the address of your FTP server, such as ftp.northpole.com.
6. If the Anonymous box is checked, uncheck it.
7. In the User Name field, type the user ID by which you are known on your Web server; this will probably be the first part of your e-mail address, up to but not including the @ sign.
8. In the Password field, type your password. If you are the only person using this computer, you can let Core FTP remember your password. If other people use this computer, however, you should check the box titled *Don't save password*. When this box is checked, you have to type your password each time you log on to your FTP site.
9. Leave the rest of the settings in the Site Manager window alone for now. You learn about the advanced settings in Chapter 13, which covers Internet security.

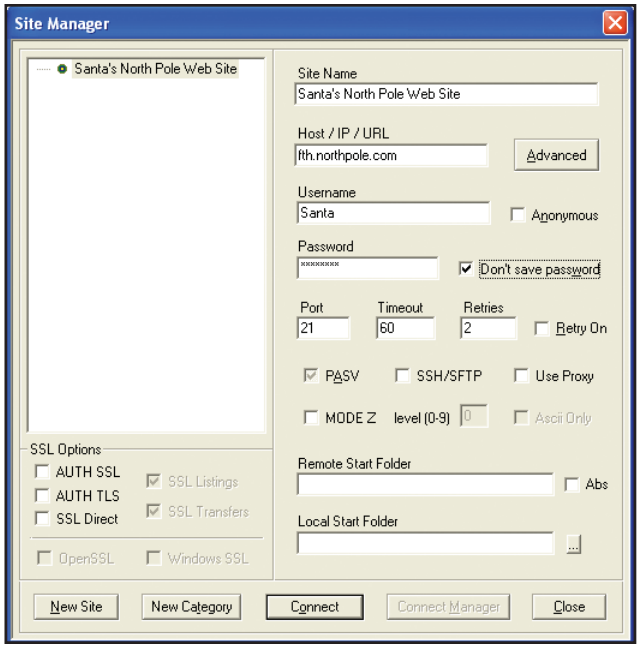


FIGURE 7-19 *When you run Core FTP Lite for the first time, you use the Site Manager to configure your FTP connection. Illustrated here are the settings Santa Claus entered to create his FTP connection.* ■

How to FTP Files to the Web

Figure 7-20 shows that the Core FTP Lite program has graphical controls that make it easy to FTP a file to your Web site. For example, suppose you want to FTP your home page and your résumé from your computer to your World Wide Web account. Follow these steps:

1. If you do not already have Core FTP Lite running, click the Windows Start button and choose Programs | Core FTP | Core FTP. When the Site Manager window appears, select the site to which you want to connect. Santa Claus, for example, would choose Santa's North Pole Web Site.

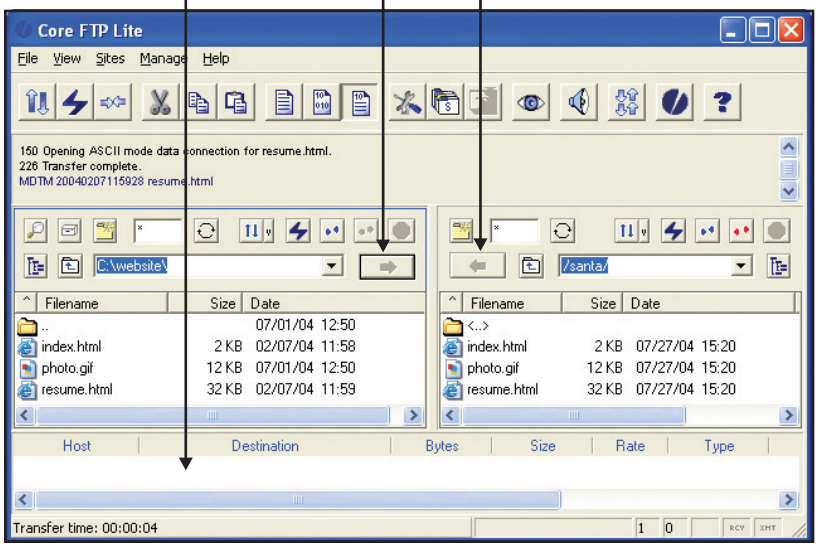


FIGURE 7-20 *The Core FTP Lite program displays folder listings for your local computer and the remote site.* ■

2. At the bottom of the Site Manager window, click the Connect button. Wait a few seconds while Core FTP makes the connection.

3. The left side of the Core FTP window is the My Computer side, and the right side is the remote computer side. On the left side of the Core FTP window, browse to the folder in which the file you want to transfer resides; in this example, browse to the *website* folder on your hard drive. See Figure 7-20.
4. On the right side of the Core FTP window, browse to the folder in which you want to transfer the files; in this example, that will be the main folder of your World Wide Web account.
5. To transfer your résumé, click once on *resume.html* on the left side of the Core FTP window; then click the → button to transfer the file to the Web. After the transfer completes, you will see your *resume.html* file listed on the right side of the Core FTP window in your World Wide Web folder.
6. To transfer your home page, click once on *index.html* on the left side of the Core FTP window; then click the → button to transfer the file to the Web. After the transfer completes, you will see your *index.html* file listed on the right site side of the Core FTP window in your World Wide Web folder.
7. To transfer your images, click once on the name of an image on the left side of the Core FTP window; then click the → button to transfer the file to the Web. After the transfer completes, you will see your image file listed on the right side of the Core FTP window in your World Wide Web folder.
8. Repeat step 7 for each image you want to publish on the Web. If you click one file and then SHIFT-click another file, you can select multiple files to transfer all at once. To add a single file to a group of selected files, hold down the CTRL key and click the filename once. You can also transfer files by clicking and dragging them from the My Computer side to the remote-site side of the Core FTP window.

How to Delete and Rename Files at a Web Site

Your FTP software provides a way to delete files you no longer want on the Web. It also lets you rename files. To delete and rename files at a Web site, follow these steps:

1. If you do not already have Core FTP Lite running, click the Windows Start button and choose Programs | Core FTP | Core FTP. In the Site Manager window, select the site to which you want to connect and click the Connect button.
2. Click once on the name of the file you want to delete or rename on the remote-site side of the Core FTP window in your World Wide Web site.
3. To delete the file, click the Delete button. Core FTP will ask if you really want to delete it. Click the Yes button if you really want to.

4. To rename the file, click it once to select the name you want to change. Click the name a second time to enter text-editing mode. Type the new filename, or modify the existing name, and then press ENTER.

Coping with Case-Sensitive File Servers

Remember that many World Wide Web servers are case sensitive. If your Web server is case sensitive, you need to make sure that the filenames you FTP to the server match the case you gave them in your HTML source code. For example, the UNIX operating system is case sensitive. On a UNIX-based server, if an image is named *PORTRAIT.GIF* and your HTML file attempts to access it as *portrait.gif*, you will get a "File Not Found" error. Folder names are also case sensitive; make sure the case of your folders on the Web matches the case you gave them in your HTML source code.

To cope with case-sensitive file servers, always keep the names of your files and folders in all uppercase or all lowercase. Most people use all lowercase, which is what I recommend you do. You should also avoid typing spaces or special symbols in your filenames and folder names.

Correcting Problems and Updating Published Web Pages

Sometimes you will need to correct a problem or update information on a Web page that you have published to your Web site. A link might have gone out of date, for example, and you need to update it. Maybe you have received an award, and you want to put it on your résumé page. The way to correct a problem or update information on a published Web page is to open the page with your text editor, correct the problem, save the page, and republish it to your Web site.

After you republish the page, be sure to test it with your browser to make sure the changes work correctly. Because your browser will have cached the previous version of the file, you need to click your browser's Reload or Refresh button to make your browser read the new version of the file. To make your browser refresh everything, including the graphics as well as the text, hold down the Shift key while you click the browser's Reload or Refresh button.

Relative vs. Absolute Links

Links on the Web can be relative or absolute. A relative link means that a file has been specified without its complete URL. The browser will look for the file in folders related to the current Web page's folder; hence the term *relative*. An absolute link means that the complete URL has been specified.

Suppose Santa Claus has a folder at his Web site called *wishlist*. In the *wishlist* folder is a file called *danny.html* that contains the list of presents Danny wants for Christmas. On Santa's home page, Santa can link to Danny's wish list either as a relative link or as an absolute link. The relative link would be wishlist/danny.html. The absolute link would be http://www.northpole.com/~santa/wishlist/danny.html.

Because relative links make it easy to work with a Web site on your computer's hard drive, Web creation tools such as Netscape Composer, Microsoft FrontPage, and Dreamweaver use relative links when you create links to files relative to the page you are creating. In addition to making it easy to publish the site from your PC to the Web, relative links make it easier to move the site from one Web space to another, should you ever decide to move the site. For the relative links to work when you move or transfer the files to the Web, you must maintain a good directory structure on your PC and at your Web site.

Maintaining a Good Directory Structure

You need to be careful how you create folders and subfolders when you make a local web that you plan to mount on the World Wide Web. Because the links you make to local files are made relative to those files, the directory structure of the local web must be exactly the same as you intend to have out on the World Wide Web. Otherwise, the links will fail when you transfer your local web to the World Wide Web.

Suppose you have lots of HTML files, pictures, sounds, and movies that you plan to mount on the World Wide Web. You should keep them organized in a neat directory structure like the one illustrated in Figure 7-21.

If your files are scattered across multiple folders and multiple drives on your PC, it will be time consuming and tedious to re-create that same directory structure on your World Wide Web server. Troubleshooting problems that occur on Web sites that are not well organized is also difficult. One of the most common causes of links not working, for example, is when you move a file into a different folder after you have already created a Web page that links to it. You can avoid this kind of problem by adopting a good directory structure and adhering to it.

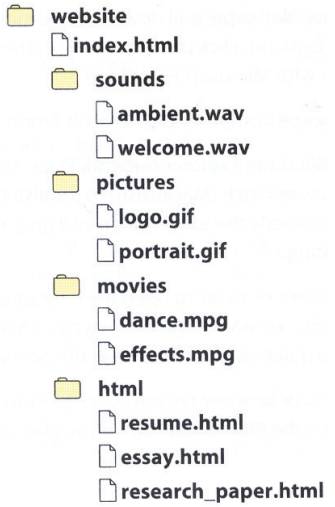


FIGURE 7-21 *Maintaining a good directory structure is important in managing the files at a Web site. Because the links you make to local files are made relative to those files, the directory structure of the local web must be exactly the same as you intend to have out on the World Wide Web. If you keep the files in subfolders of the local web folder you plan to publish, relative links to those files will work fine out on the World Wide Web.* ■

How to Create New Folders on the Web

As the number of files at your Web site increases, you may choose to create folders to help keep your site organized. For example, if you create a series of Web pages related to your work, you might create a folder called *work* to keep them in. The directory structure you create on the Web must mirror the structure of the *website* folder on your PC. If you create a *work* folder at your Web site, then you must also create a *work* subfolder in the *website* folder on your PC. To create a folder at your Web site, follow these steps:

1. If you do not already have Core FTP Lite running, click the Windows Start button and choose Programs | Core FTP | Core FTP. In the Site Manager window, select the site to which you want to connect and click the Connect button.
2. On the remote-site side of the Core FTP window, make sure your current directory is the one in which you want to create a new folder. If it's not, double-click on a directory name to select it, or click the Up Directory button to move back a level of directory structure.

3. Click the Make Directory button to open the Make Directory dialog.

4. Enter the name of the folder you want to create and click OK.
5. Wait for a second or two while the new folder gets created. When Core FTP refreshes the directory listing, the new folder will appear in it.
6. If you want to enter the new folder, double-click its icon.

note *When you name the folder, avoid special characters and spaces. Instead of a space, you can type a dash or an underscore to represent the space.*

Setting the File Permission Attributes

After you FTP your files to the Web, if people are not able to access your files in the manner you want, you may need to set the **file permission attributes**, which are settings that determine who is allowed to read and execute your files. You will probably want to set these attributes to let anyone in the world read and execute your files but allow only you to modify or delete them. Core FTP makes it easy to change the permission attributes. To change the permission attributes of a file with Core FTP, follow these steps:

1. If you do not already have Core FTP Lite running, click the Windows Start button and choose Programs | Core FTP | Core FTP. In the Site Manager window, select the site to which you want to connect and click the Connect button.
2. Right-click the file or folder whose attributes you want to change.
3. When the menu pops up, choose Properties.
4. The File Properties dialog appears as illustrated in Figure 7-22.
5. Check the boxes to let Owner, Group, and Other read your files, but allow only the Owner to write them. (The owner is you!)
6. Click OK to close the dialog and make the changes take effect.

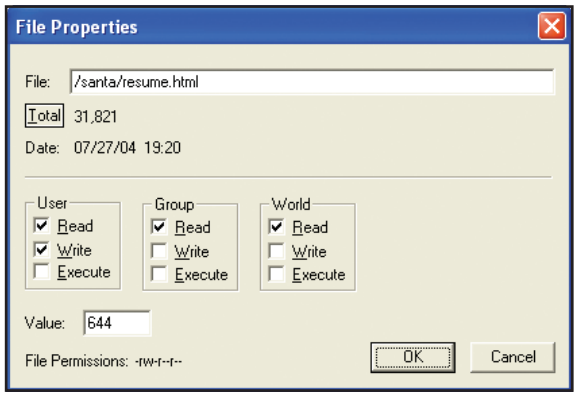


FIGURE 7-22 *The Core FTP Properties dialog lets you change the file permissions on UNIX hosts.* ■

Chapter 7 Review

Chapter Summary

After reading this chapter and completing the step-by-step tutorials and Try This! exercises, you should understand the following facts about the Internet:

Creating HTML Forms

- A form is an HTML Web page element that prompts the user to fill in one or more blanks and/or select items from a menu of possible choices, after which the user clicks a button to submit the filled-out form.
- The `<form>` start and `</form>` stop tags mark the beginning and ending of an HTML form. Between these tags go the controls that prompt the user for specific information, along with any onscreen instructions you may wish to provide.
- A prompt is an onscreen instruction that asks the user to provide some information, either by typing a response into a text field or by choosing something from a selection of possible choices.
- In HTML, you use the `<input>` tag to create the form controls that receive input from the user. The `<input>` tag has a `type` attribute that specifies the kind of control you want.
- To create a text field, you set the `<input>` tag's type to *text*. The `<input>` tag also has a `name` attribute that you use to specify the name of the control and a *`size`* attribute that specifies how many characters wide the field will be onscreen. There is also a *`maxlength`* attribute that specifies the maximum number of characters the user can type.
- A radio button is a form control that displays a small round button with which the user can select one, but not more than one, item from a list of possible choices. You create a radio button by setting an `<input>` tag's `type` attribute to `radio`. You use the `name` attribute to assign a name to the radio button group, which includes any other radio button of the same name within the form.
- To create a check box, you make an `<input>` tag of type check box.
- To create a selection menu, you use the `<select>` start and `</select>` stop tags. Between these tags, you use the `<option>` tag to specify the menu options from which the user can make a selection. To permit multiple selections, you add the `multiple` attribute to the select tag.
- To create a text area, you use the `<textarea>` start and `</textarea>` stop tags. To set the width, you use the `cols` attribute, which tells the browser how many characters wide to make the text box. To set the height of the text area, you use the `rows` attribute.
- A password field is a text box that works like a text field except that when the user types the entry, the browser displays asterisks instead of revealing the sensitive information onscreen. You create a password field by creating an `<input>` field that has its `type` attribute set to password.
- You make a Submit button by creating an `<input>` tag that has a `type` attribute set to submit. By default, the value the browser prints inside the button is the same as its type. You can change this value via the `value` attribute.
- You make a Reset button by creating an `<input>` tag that has a `type` attribute equal to `reset`.
- There are two HTTP methods that the `<form>` tag can use to submit the form's data when the user clicks the Submit button: method="get" and method="post". The GET method puts the form data in a query string that gets appended to the URL specified in the form tag's `action` attribute. The POST method sends the form data to the process identified in the form tag's `action` attribute. This process is normally a server-side script that reads and processes the form data, saves appropriate information in a server-side database, and returns an HTML response to the user.

Making a PayPal Buy Now Button

- PayPal uses two special form techniques to power its button. First, PayPal uses hidden fields to identify the business that is selling the product, the name of the product being sold, and the selling price. Second, PayPal substitutes the PayPal button image for the Submit button that normally appears on an HTML form.
- A hidden field is an HTML form control that creates a variable name and a corresponding value that are not displayed onscreen but are posted along with the rest of the form data when the user clicks the button to submit the form. You create a hidden field by setting an `<input>` tag's `type` attribute to `hidden`. In the `name` and `value` attributes, you type the variable name and its corresponding value.
- To make a different graphic substitute for the default Submit button in a Web form, you use an image button, an HTML form element you create with an `<input>` tag of type image. You set the tag's `name` attribute to `submit`, which makes the image button post the form data when clicked.
- When clicked, the PayPal Buy Now button posts HTML form data to the PayPal e-commerce server. In addition to containing information about the product being purchased, the form data identifies the business that is making the sale.
- For users who are not programmatically inclined, PayPal has a button factory that generates automatically the HTML to paste onto your page to sell items at your Web site.

Designing HTML Image Maps

- An image map is an invisible layer of triggers placed over an image on the screen. The triggers can be rectangular, circular, or polygonal. When the user clicks inside one of the invisible triggers, the Web browser triggers the object of the link.
- In earlier versions of HTML, using image maps required the use of a CGI call. When the user clicked a trigger in an image map, the browser sent the coordinates of the mouse click to a CGI program on a server, which told the browser what to do in response to the click. In the latest versions of HTML, it is possible to process image maps locally within the document instead of having to call on a CGI program for help in handling the mouse click.
- The `<map>` and `</map>` tags demarcate the beginning and ending of the image map. Between them, you see the `<area>` tags that create the triggers. Inside each area tag is a `shape` attribute and a `coord` attribute that specifies the x,y coordinates of the links. These coordinates are pixel addresses inside the image. The top-left corner of an image is always pixel address 0,0 and is called the origin. The other addresses are all relative to the origin. The `coord` attribute has the syntax *coords*= *x1,y1,x2,y2*. The top-left corner of the area is x1,y1, and the bottom-right corner is x2,y2.
- Once you have defined an image map with the `<map>` tag, you use the `usemap` attribute to apply it to any image on the page. The `usemap` attribute modifies the `<img>` tag that puts the image onscreen. The syntax of the `usemap` attribute is `usemap=#`*`area_name`* where *area_ name* is the name you gave the image map in the `<map>` tag's `name` attribute. When the mouse clicks on one of the areas in the image map, the user follows the corresponding link.
- Many Web page creation tools have image map assistants built in that can help you create image maps. Or you can use a tool such as Paint Shop Pro to figure out pixel locations.
- Because client-side image maps execute in the browser, you can create an image map without needing access to a CGI server. With all of the processing done on your computer, you do not even need to be connected to the Internet. All you need is a browser and a text editor, such as the Notepad.

Creating HTML Frames

- A frameset is an HTML Web page element that splits the browser window into two or more subdocuments called frames. You create a frameset with the `<frameset>` start and `</frameset>` stop tags. The start tag has `row` and `column` attributes that determine the layout.
- Vertical frames divide the browser window into side-by-side columns onscreen. You create vertical frames via the `cols` attribute of the `<frameset>` tag.
- Horizontal frames divide the browser window into rows that appear in order from top to bottom onscreen. You create horizontal frames via the `rows` attribute of the `<frameset>` tag.

- A frameset grid is a layout in which there are both vertical and horizontal frames onscreen. You create a frameset grid by using both the `rows` and `cols` attributes in a frameset tag.
- To divide one of the rows or columns in a frameset grid, you must divide them all. In practice, you do not normally want all the frames to be divided. Instead, you would normally want to divide just one frame into subframes. To divide a single frame into subframes, you create a frameset inside that frame. A frameset that you create inside of another frame is called a nested frameset.
- To remove the border around a frameset, you add the following attributes to the frameset tag:

```
frameborder="no" border="0" framespacing="0"
```

- To remove the scrollbar from a frame, you add the following attribute to its frame tag:

```
scrolling="no"
```

- To prevent users from being able to resize a frame, you add the `noresize` attribute to the frame tag.
- You can use the * wildcard character to size frames proportionally. The following frameset, for example, creates a layout in which the second column is three times as wide as the first and third columns:

```
<frameset rows="*,3*,*">
```

- One of the most useful applications of a frameset is to establish a targeting relationship such that a mouse click in one frame can alter the display of a document in another frame. This is how you make hyperlinks in a left sidebar frame, for example, display their documents in the targeted main frame. Establishing such a targeting relationship is a two-step process. First, in the frameset, you use the `name` attribute to give the targeted frame a name. Second, in the document containing the hyperlinks, you use the `target` attribute to make the links come up in the targeted frame.
- You can save a lot of time typing if you use the `<base>` tag to specify the target of the sidebar links. The `<base>` tag goes into the `<head>` section of the document containing the hyperlinks. The format is:

```
<base target="your_target">
```

Publishing a Web Site via FTP

- The first time you use your FTP software, you will need to configure a new connection for your Web site. The new connection configuration identifies the domain name of your Web server and your user ID on that server.
- In addition to enabling you to publish files to your Web site, the FTP software lets you delete files you no longer want on the Web. It also lets you rename files.
- Many World Wide Web servers are case sensitive. If your Web server is case sensitive, you need to make sure the filenames you FTP to the server match the case you gave them in your HTML source code.
- Links on the Web can be relative or absolute. A relative link means a file has been specified without its complete URL. The browser will look for the file in folders related to the current Web page's folder; hence the term *relative*. An absolute link means that the complete URL has been specified.
- You need to be careful how you create folders and subfolders when you make a local web that you plan to mount on the World Wide Web. Because the links you make to local files are made relative to those files, the directory structure of the local web must be exactly the same as you intend to have out on the World Wide Web. Otherwise, the links will fail when you transfer your local web to the World Wide Web.
- After you FTP your files to the Web, if people are not able to access your files in the manner you want, you may need to set the file permission attributes, which are settings that determine who is allowed to read and execute your files.

Key Terms

check box *(8)*
common gateway interface (CGI) *(8)*
file permission attributes *(35)*
form *(2)*
frame *(19)*
frame targeting *(19)*
frameset *(19)*
GET method *(8)*
hidden field *(12)*
image button *(13)*
image map *(16)*
<input> tag *(4)*
maxlength attribute *(4)*
name attribute *(4)*
nested frameset *(22)*
<option> tag *(10)*
password field *(11)*
POST method *(8)*
prompt *(3)*
radio button *(5)*
Reset button *(6)*
<select> tag *(10)*
size attribute *(4)*
Submit button *(6)*

Key Terms Quiz

1. A(n) _____ is an HTML Web page element that prompts the user to fill in one or more blanks and/or select items from a menu of possible choices, after which the user clicks a button to submit the response.
2. A(n) _____ field is a text box that works like a text field except that when the user types the entry, the browser displays asterisks instead of revealing the sensitive information onscreen.
3. A(n) _____ button is a form control that displays a small round button that enables the user to select one, but not more than one, item from a list of possible choices.
4. A(n) _____ is a form control that displays a small box that the user can check to select one or more items onscreen.
5. The _____ causes an HTML form to put its data in a query string that gets appended to the URL specified in the form tag's `action` attribute.
6. The _____ causes an HTML form to send the form data to the process identified in the form tag's `action` attribute.
7. A(n) _____ field is an HTML form control that creates a variable name and a corresponding value that are not displayed onscreen but are posted along with the rest of the form data when the user clicks the button to submit the form.
8. To make a different graphic substitute for the default Submit button in a Web form, you use a(n) _____ button, which is an HTML form element you create with an `<input>` tag of type _____.
9. A(n) _____ is an invisible layer of triggers placed over an image on the screen. The triggers can be rectangular, circular, or polygonal. When the user clicks inside one of the invisible triggers, the Web browser triggers the object of the link.
10. A(n) _____ is an HTML Web page element that splits the browser window into two or more subdocuments called _____.

Multiple-Choice Quiz

1. A form control that presents the user with a small box that the user can click to turn an option on or off is called a
   a. check box
   b. radio button
   c. multiple item selection menu
   d. single item selection menu
2. To create a text field, you set the `<input>` tag's type to
   a. check box
   b. password
   c. radio
   d. text

3. Which attribute sets the width of a text area?
   a. `cols`
   b. `height`
   c. `rows`
   d. `width`
4. Which attribute assigns a name to an `<input>` control?
   a. `label`
   b. `name`
   c. `size`
   d. `value`
5. In an `<input type="submit">` tag, which attribute changes the default word *submit* that the browser prints inside the form's Submit button?
   a. `label`
   b. `name`
   c. `size`
   d. `value`
6. Which tag creates a trigger that specifies the x,y coordinates of the links in an image map?
   a. `<area>`
   b. `<coord>`
   c. `<map>`
   d. `<usemap>`
7. Which tag specifies the code that will execute in a frames disabled browser?
   a. `<cancelframes></cancelframes>`
   b. `<framesdisabled></framesdisabled>`
   c. `<noframes></noframes>`
   d. `<noframeset></noframeset>`
8. Which frameset divides the window vertically into two equal halves with no other divisions or subdivisions?
   a. `<frameset rows="50%,50%">`
   b. `<frameset cols="50%,50%">`
   c. `<frameset rows="50%,50% cols= "50%,50%">`
   d. `<frameset rows="50%,*">`
9. Which tag creates a frameset grid?
   a. `<frameset rows="50%,50%">`
   b. `<frameset cols="50%,50%">`
   c. `<frameset rows="50%,50% cols= "50%,50%">`
   d. `<frameset rows="50%,*">`
10. Which frameset tag creates a layout in which the second row is three times as wide as each of the first and third rows?
    a. `<frameset cols="*,3*,*">`
    b. `<frameset rows="*,3*,*">`
    c. `<frameset cols="*,2*,*">`
    d. `<frameset rows="*,2*,*">`

Essay Quiz

1. The PayPal form uses an image button to replace an HTML form's default Submit button with PayPal's customized PayPal button, which contains the PayPal branding. Think of a situation in which you could create a custom button to use on forms in your school or workplace Web site. What graphical branding elements would you include in such a button, and what would you want to make happen when the user clicks it?
2. In this chapter, you learned how to make hidden fields containing form data that does not appear onscreen. Can hidden fields be used to contain confidential information that you wouldn't want the user to see? Before you answer this question, pull down your browser's View menu and choose Source to see how users can reveal the source code of any form onscreen.
3. Give an example of a situation in which you could use an image map on your current school or workplace Web site. Describe the graphic that the map would comprise, and explain what would happen when users click the map's hyperlink areas.
4. Figures 7-17 and 7-18 display frameset and frameless versions of an online course, respectively. Visit these live by going to www.udel.edu/fth/InternetLiteracyModules for the frameset and www.udel.edu/

fth/InternetLiteracyModules/mainContent.htm for the frameless version. If you have any trouble getting them onscreen, be aware that www.udel.edu is a case-sensitive Web server, so you must type the lowercase and uppercase letters exactly where they appear in these folder names and filenames. After you get the course onscreen, try navigating via the sidebar in the frameset version, and also try clicking the navigational icons at the bottom of each screen in the frameless version. Compare the navigational impression that each version gives you. Then write a brief essay in which you express your personal position on framesets. In what kinds of situations would you want to use frames, and when would you recommend against them?

Lab Projects

• Lab Project 7-1: Person-to-Person Payment Processing

PayPal presents an interesting alternative for a small business that is considering how to power its e-commerce activities. In this chapter, for example, you have learned how to create Web forms that present users with onscreen controls for selecting things and making choices. By using a PayPal button to submit those choices to CGI scripts running on the PayPal server, it would seem as though small businesses could avoid the overhead of paying for more complex e-commerce hosting from an instant storefront provider such as Amazon, Yahoo, eBay, or bCentral. Imagine that your employer has asked you to evaluate these alternatives and recommend whether your small business should use PayPal Buy Now buttons to sell products directly from your Web site. Use your word processor to write a report in which you describe how PayPal works, compare it to the leading instant storefront alternatives, and recommend whether to use PayPal at your site. In creating this recommendation, address the following issues:

- **Functionality** Earlier in this chapter, Table 7-3 listed the frequently used PayPal Buy Now Button HTML and hyperlink variables. Go to the PayPal site's Merchant Tools tab, and look over the complete list of hyperlink variables that the PayPal Buy Now button supports. Do these variables cover the kinds of options you will need on your company's e-commerce pages?
- **Cost** Consider the costs involved in adding PayPal Buy Now buttons at your company Web site as compared to paying for an instant storefront solution. Be sure to include all costs in this comparison, such as fixed monthly fees and per-purchase percentages that you must pay for different kinds of payment processing.
- **Time** Creating your own storefront may take more time than using an Instant Storefront solution. Consider whether you have that kind of time. Also consider whether the PayPal tools, such as the Buy Now Button Factory, can speed the process.
- **Opportunity** Designing your own e-commerce pages may provide opportunities that you would not have at an instant storefront. Consider the options available at popular instant storefronts such as Amazon, Yahoo, eBay, or bCentral. Do you need features that are not supported by these instant storefronts?

If your instructor has asked you to hand in this assignment, make sure you put your name at the top of your essay; then copy it onto a disk or follow the other instructions you may have been given for submitting this assignment.

• Lab Project 7-2: Creating a Graphical Front End

If you think about what you actually see when you go to a Web site, what appears onscreen is either textual or graphical. Web sites that depend mainly on text, such as keyword search engines, typically have text-based user interfaces. Other sites get more creative and provide you with a visual image to click. You can see an example of a graphical user interface at the Library of Congress Web site at www.loc.gov. Imagine that your employer has asked you to look into the possibility of creating a graphical front end for your school or company Web site. In developing your recommendation, consider these alternatives:

- One way to create a graphical front end is to use an image map. This method is suitable if you have a picture or a graphic that illustrates or contains the objects you want users to click. I dream of creating a virtual tour of Fort McHenry, for example, which is located in Baltimore and chronicles the writing of the Star Spangled Banner. You could snap a wide-angled picture of the fort, put it on a Web page, and use an image map to create triggers on the doors and windows. To look into a window, the user would click it. To enter a room, the user would click its door.
- Another way of creating a graphical front end is to use tables to position individual graphics onscreen where you want them. To link the graphics to their intended targets, you surround each image by anchor tags that use the `HREF` attribute to specify the hyperlinks. In the previous chapter, Figure 6-22 analyzes the design of a Web page that uses tables to create this kind of layout.
- If your workplace has content that lends itself to the use of image maps, use your word processor to write a brief essay about that. Describe what the images would depict, tell where the triggers would be, and explain what will happen when the user clicks them.

If your instructor has asked you to hand in this assignment, make sure you put your name at the top of your essay; then copy it onto a disk or follow the other instructions you may have been given for submitting this assignment.

Main text file is embedded
because of difficulty w/ keyword references.
PLS. always export the text after done with the chapter.
(File - Export Text ...)

PLEASE PLACE THIS NOTE AT END OF CHAPTER,
AFTER BUILDING AS A REMINDER
DURING CORRECTIONS