In this chapter

- What you need to know about HTML
- More about creating and coding style sheets
- Web scripting languages primer
- A few words about XML

23

Knowing the Code

Itching to take your site to a whole new level? Find yourself eager to add some neat touches that other sites offer but you don't know how? Wonder what a script is but don't want to learn programming to write one?

FrontPage does so much for you, yet it doesn't offer complete soup-to-nuts functionality. Some highly desirable features require you to reach past FrontPage to grab tools and tricks along with trial-and-error efforts to achieve the results you want.

What may surprise you in your pursuit of an even better site is that beyond the Web buzzwords are advanced options that you don't need to be a developer or a genius to incorporate. Many people have blazed the electronic frontier ahead of you and now offer helpful paths for you to follow their steps in the form of tutorials, HTML and scripts you can add to your pages, as well as examples of how XML can let you wed data to your site in a format that can be easily exchanged with others.

Why It's Important to Understand HTML

Probably one of the reasons you purchased FrontPage 2003 is because it's advertised as a product that will help you produce a great-looking, professional-quality Web site—without requiring you to know a lick of HTML. Indeed, you can use FrontPage to hammer out a first site—as you've done—without necessarily knowing much beneath the surface of the product's tools and features.

Ah, but there is so much more that is possible when you peer beneath the hood and begin to work at a code level. Doing so enables you to tweak your pages and design with better precision, overcome some of FrontPage's limitations and failings, and add scripts that will afford your site better functionality and up-to-the-minute features. In effect, you're only limited to what FrontPage offers if you choose to be. Otherwise, the design world is your virtual oyster with that precious pearl within.

One of FrontPage's great features is that it enables you to use Code view (discussed in Chapter 11, "Using Code in FrontPage") to look at the code for pages you've already built, and compare the code with the actual pages in Page Preview view. In this way, you can discern what each section of the HTML code does (not to mention scripts and XML). Then, through trial and error, you can make it do something else.

Although this chapter can't begin to teach you all you eventually may want to know about HTML, scripting, and XML, you can get a running start here. Then when you're ready to learn more, you can explore the incredible wealth of material about them on the Web.

HTML Basics

Hypertext Markup Language, or *HTML*, uses specific tags to identify particular elements in a Web page. Every Web page requires a certain fundamental set of HTML tags to be recognized as a Web page and to load in a browser. For example, Figure 23.1 shows a very simple page in HTML with these essential tags.

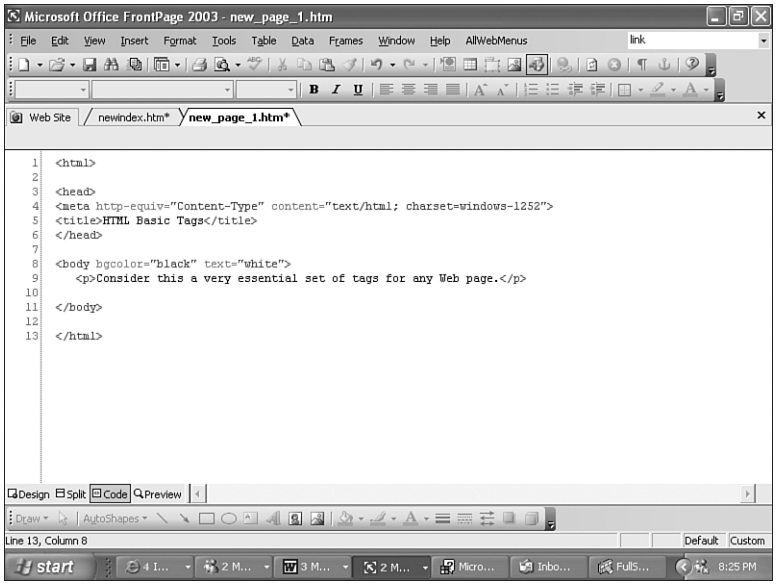


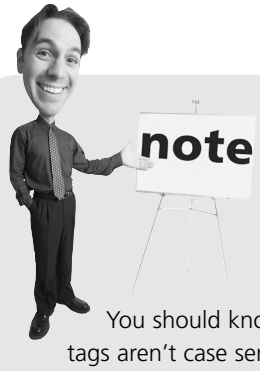
FIGURE 23.1
These are the page tag essentials.

Such tags appear with opening and closing angle brackets and usually appear in pairs, where there is an opening tag, such as <TITLE>, and a closing tag, such as </TITLE>. That backslash before the element (here, a page title) designates it as a closing tag. Tags typically surround text to provide formatting and placement information and to identify the presence of a hyperlink or of a file (graphic, video, or sound file) that needs to be loaded with the page.

Tags are also used to identify a specific area of a page. Look at the figure again, and you'll see that the main text of the page appears between paragraph tags— and .

Within the tag, in addition to the page element the tag identifies, an attribute may appear that specifies something about how the page element should look or behave. In the example you saw in Figure 23.1, the <BODY> tag includes an attribute for both a background color (bgcolor) of black and a text color (text) of white.

Table 23.1 shows some of the most common and useful HTML tags and attributes, along with an understanding of what each specifies.



You should know that tags aren't case sensitive, so typing <STYLE> will produce the same result as typing <style>.

Table 23.1 HTML Tags and Attributes

Tag	Function
<BODY> </BODY>	Identifies the visible content of the page.
<HEAD> </HEAD>	Used to specify the title of the page (<TITLE>) and any additional document-identifying information.
<HTML> </HTML>	Anything contained within these tags is HTML.
<STYLE> </STYLE>	Delineates particular style information for the document.
<TITLE> </TITLE>	Specifies the actual title of the page and appears between the <HEAD> and </HEAD> tags.
Attribute	**Function**
ALINK	Used to specify the color to be used for active hyperlinks, such as ALINK="red".
BACKGROUND	Identifies any image to be used as a background for the page, such as BACKGROUND="news.gif".
BGCOLOR	Specifies a background color for the page, such as BGCOLOR="green".
LINK	Identifies the color to use for unclicked hyperlinks, such as LINK="purple".
TEXT	Allows you to define a color for the associated text, such as TEXT="white".
VLINK	Lets you specify a color for a link that has been clicked, such as VLINK="blue".

There are specific text and formatting tags as well. Table 23.2 shows a few of these.

Table 23.2 Specific Text and Formatting Tags

Tag	Function
	Any text between these tags appears in boldface type.
 	Indicates a line break; no closing tag is used.
 	Specifies a size for the font to be used, where *xx* is the size of the font, such as 10 pt or 20 pt.
	Designates a level-one heading (the largest).
	Designates a level-six heading (the smallest).
	Any text between these tags is italicized.
	Anything appearing between these tags displays as a bulleted list.
	Any text between these tags appears as a separate paragraph.
	Sets the alignment for a paragraph, where *xx* could be left, right, or center.

Tag	Function
<table> </table>	Establishes a table on the page.
	Anything between these tags appears as an unnumbered list.

The tags shown in Table 23.3 help you specify information related to the links used.

Table 23.3 Hyperlink and Email Tags

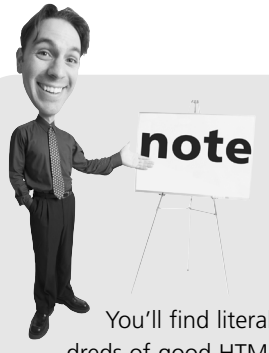
Tag	Function
 	Creates a hyperlink where the URL is a specific page or site address.
 	Creates a link to a specified email address.

Finally, Table 23.4 provides some tags that help you delineate files to be loaded as part of the page.

Table 23.4 Image-related Tags

Tag	Function
<img src="filename">	Specifies an image to load, where the filename is the full name of the image.
<img src="filename" align=*xx*>	As above, but indicates how the image should be aligned (left, right, center, top, and so on).

These are just a few of the many different HTML tags and their attribute variations that can be included on a Web page. When you learn how to use Code view in the next chapter, you can use some of these basic tags to better understand how FrontPage has helped you create your pages. You can then learn more advanced HTML as you further develop your site.



You'll find literally hundreds of good HTML tutorials and articles on the Web by using a search engine such as Google. Two strong ones are the WebMonkey's HTML cheat sheet at `http://hotwired.lycos.com/webmonkey/reference/html_cheatsheet/` and Dave Raggett's Introduction to HTML at `http://www.w3.org/MarkUp/Guide/`.

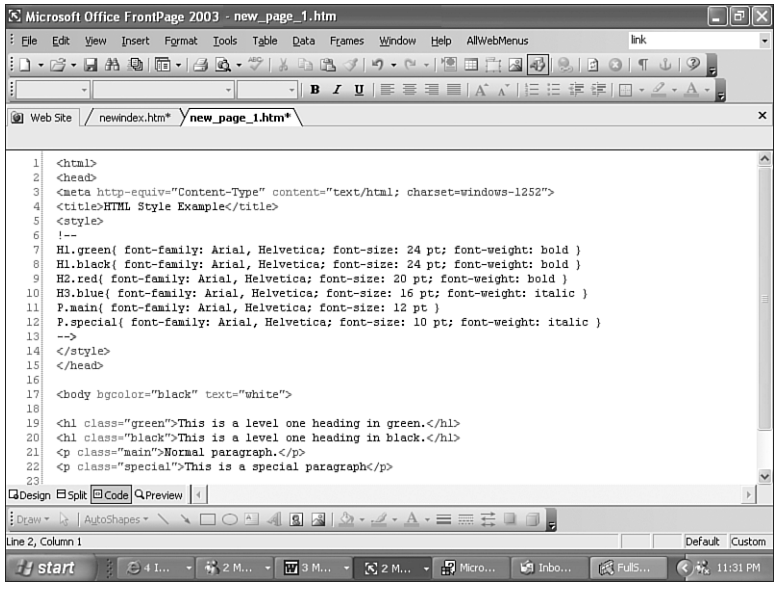
Coding Style Sheets

Chapter 13, "Using Styles and Cascading Style Sheets," offered an introduction to the topic of Web style sheets and their role in helping you control how your pages look when viewed in a Web browser.

To give a short recap, style information embedded into a Web page always exists between <STYLE> and </STYLE> tags and between !-- and --> tags. All of this, in turn, appears between the <HEAD> and </HEAD> tags. Please note that the lack of brackets around the !-- and --> is not a fluke; they are supposed to appear just like that.

To see an example of styles embedded into HTML, look at Figure 23.2. Here, two different types of H1 headings and paragraphs are delineated. When you want to choose a specific version of each, you do so using the <h1 class="color"> or <p class="type"> tags to do so.

FIGURE 23.2
An example of an embedded style sheet available within the HTML code of a FrontPage Web page.



For information about coding external style sheets, refer to the example provided in Chapter 13.

Coding Colors

One thing you'll see as you begin to examine HTML code from your pages as well as pages in other Web sites is that colors can be identified in different ways. The two most common formats are as follows:

- The name of the color, in quotation marks
- The hexadecimal number (called hex code) for the color, preceded by the # sign

Of the two, hex code is usually preferred for two reasons: because it's in a language the Web browser can understand, and because it gives you more control over your color choices. For example, think of how many different shades of blue exist. You can specify blue in your HTML, but which shade of blue will you get? Will it be dark, light, or medium?

With hex code, different shades have different numbers assigned to them. When you want to use a light blue, you might specify a hex code of #66CCFF; for medium, you might type #3333FF; and for dark, #000066. For example, to set a Web page body's background color to dark blue and the text to a very light blue, the code would look like this:

```
<body bgcolor=#000066 text=#66CCFF>
```

You can find numerous color charts with their associated hex codes on the Web. A popular one can be found at `http://hotwired.lycos.com/webmonkey/reference/color_codes/`, as shown in Figure 23.3.

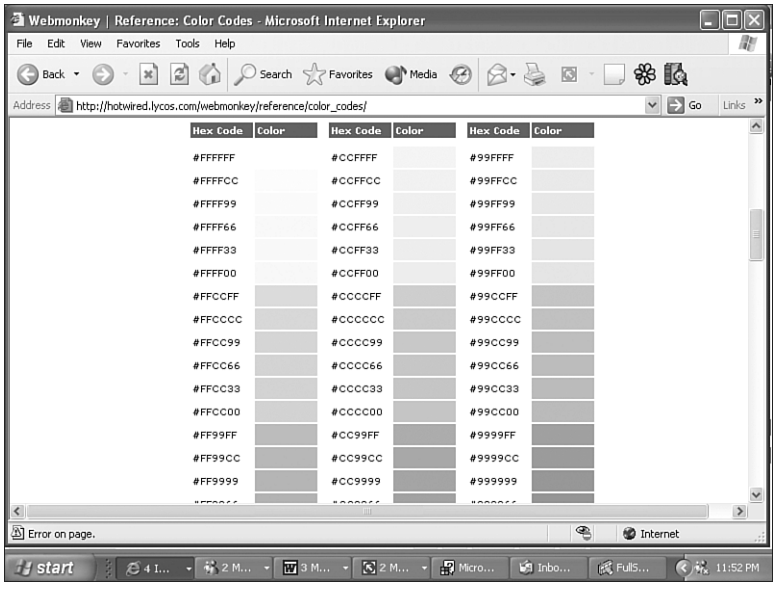


FIGURE 23.3
Webmonkey's color code chart is just one example of hundreds of hex code charts available for reference on the Web.

Scripting Languages

Most of the features FrontPage helps you add enable you to create a static Web site. A static site is one where pages get created and updated only through the direct

efforts of the person responsible for the site. Little is automated. This is different from large professional sites, where much of the generation of new Web pages is done through automated processes.

Another difference between the big boys and an entry-level FrontPage Web site is that there is little ability for the visitor to interact with the latter. Sure, FrontPage offers discussion Webs so that people can post messages, but it offers the visitor no immediate feedback.

Scripting is commonly the way a site grows from a static, noninteractive entity to one where new pages can be generated on the fly, often by a custom request by a visitor. For example, scripting might help the visitor access a database through a Web page and receive custom views based upon the kind of information he or she is looking for. More and more, visitors are coming to expect this kind of service in a Web site, particularly one that looks like a professional entity.

Such scripting is done not through HTML alone (HTML is considered a display language rather than a programming language) but through scripting languages, some of which are far easier for mere mortals like us to learn than others. Some of the most frequently used scripting languages include the following:

- **Common Gateway Interface (CGI) scripting**—The original Web scripting language; this one is typically used for working with Web forms.
- **JavaScript**—One of the most widely used scripting languages. JavaScript is suited to a number of different purposes, particularly for page elements such as page menus and fancy graphics.
- **Perl**—Widely used, particularly on the Unix (non-Microsoft) platform, for handling administrative tasks and database-to-site functions.
- **PHP**—An embedded scripting language used extensively on sites that interact with databases and data servers.

Some of these scripting languages, like JavaScript and PHP, are referred to as embedded programming languages because you can actually add the code for them to an HTML document such as a Web page; others, like Perl, are not embedded but are used through the Web server itself to output an HTML document. Most scripts, however, act between a Web site's code-inclusive page and a Web server in producing results.

Just two primary scripting languages are directly supported within FrontPage: VBScript and JScript. VBScript often only works when run within the Microsoft Internet Explorer browser, so people running Netscape, Opera, or another browser version may experience problems. This means the script may not work at all. JScript, on the other hand, is a variation of JavaScript and is better supported for cross-browser use.

Downloading Scripts

Dozens if not hundreds of high-traffic developer-oriented Web sites make scripts available for you to either copy to your page(s) directly or to download and then copy. A short list will be provided for you in a moment, but you can find many more by looking through a Web search engine or stopping by some of the Web design sites that recommend script sites in their links pages.

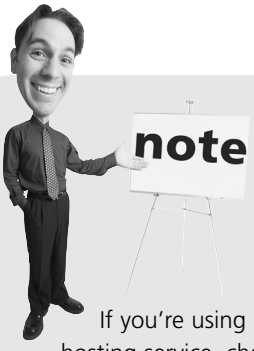
Some of these sites request a modest fee upfront for their scripts, while many allow you to download and try them for free—although they may set certain conditions for their use. For example, some sites prohibit you from taking their script apart to make your own version or from removing their name(s) from the script. Alternatively, you may have to pay a fee if you plan to use their script in a commercial, for-profit site. For this reason, you should always check the information accompanying the script or any document on the site that spells out how these scripts are meant to be used.

Before you start downloading, there are some other details to attend to:

- View complete information about what the script is supposed to do.
- Read any warnings about possible problems that may arise when using the script.
- Check to see if the script specifies that it only runs on a particular type of server. Not all scripts and scripting languages run on servers that work with FrontPage Webs.
- Determine whether the site appears reputable. A few scripting sites offer scripts that say they do one thing, when they may actually do something harmful either to your site or its visitors; if you see hacking-related material on the site, you may want to go elsewhere.
- Determine whether the site is recommended by other Web designers and developers.
- If the site has a message board or comment area, check to see what other people say about the script you want to use.
- Find out whether the site offers a URL where you can go to see the script in action.
- Read instructions on how to use the script. (See the section "Adding Scripts to Your Web" later in this chapter for more information.)

The following is a short list of some of the sites where you can find code and scripts to download. As stated before, conduct a Web search for a much broader list of possibilities.

- CGIScript.net
- Hotscripts.com
- Java-scripts.net
- Javascript.com
- Javascript.Internet.com
- ScriptSearch.com



If you're using a Web-hosting service, check its site. Many offer sample scripts that can be adapted to your site free of charge, while others may have specific rules about the type and nature of scripts you run on your site. Still others may require you to notify them before you use something like a CGI script so they can enable this capability for you from their end.

Adding Scripts to Your Web

One thing you really need to look for when downloading scripts from the Web is a list of instructions, often available alongside the description of what the script does on the site where you got the script. Many of the best sites offer rather detailed steps for getting it to work when you adapt it to your page; sometimes, these instructions will specify how to do it within FrontPage itself.

Not all scripts will be ready to use as is. You may have to modify certain parameters within the script to have it run properly on your site or work with your settings. For example, if the script is supposed to deliver some information to you via email from visitors accessing the feature on your site, you probably need to plug in your email address.

If you don't understand what you need to do to apply a script to your site, see if you can find an email address or Web message board where you can ask questions before you try to run the script. People willing to share their scripts are often happy to answer a few quick questions, although a free script doesn't always translate into hours and hours of free technical support from the developer.

Let's look at an example taken from a Web site with a sample script for creating a pop-up message on a page. Figure 23.4 shows how the JScript was added within the HTML between the <HEAD> and </HEAD> tags. Most scripts start with the <script language="language"> tag, which specifies which language is being used for the script, and then end with the closing </SCRIPT> tag.

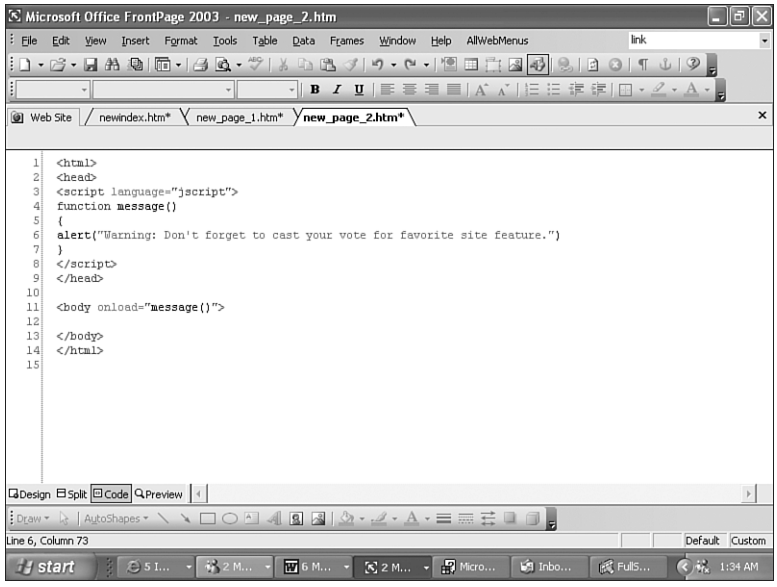


FIGURE 23.4
The short JScript code added to create a pop-up message in FrontPage Code view.

XML

Extensible Markup Language, or XML, is frequently touted as the universal data standard of the Web because its format is such that information contained within its files can be easily shared between different individuals, groups, and large organizations regardless of what kind of platforms they run. Increasingly, you see sites both large and small adopting XML as a way to exchange information both with their users and with other sites.

An XML document can seem both easy to understand as well as to create because it uses a system of tags not unlike HTML. There's a difference, however; the person designing the XML file can come up with his or her own system of tags, using his or her own wording. As long as the document meets certain criteria for a well-formed XML document (that is, the elements used are consistent throughout, all tags that are open are then closed, all values listed are contained within quotation marks, and such), the data that such a document contains can be used as a data source (not unlike a text database) by the creator as well as by others. Such data and documents must be validated, usually through the presence of special files called XML schema built into the XML document, as well as by a Document Type Definition or DTD, which serves as a master control list of document type information.

It's the data source element that makes XML compelling for a Web. For example, you could create a whole database of information reasonably quickly in XML and then allow your site visitors to access information contained within that data source via your Web pages.

FrontPage enables you to create an XML document right from a blank page and save it in .xml format. In Figure 23.5, you see an XML document designed to store two customer records (so far) in XML format in a manner that conforms to the standards set forth for XML creation.

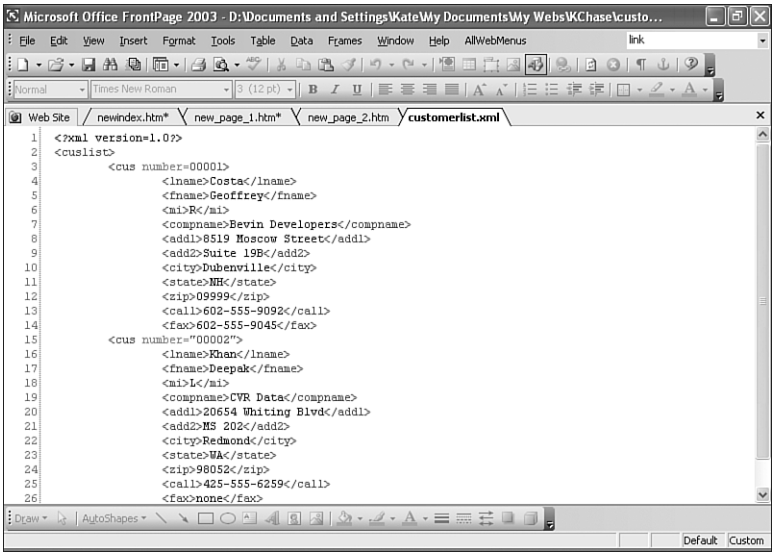


FIGURE 23.5
The two customer entries for the XML data file can probably be understood pretty quickly with a little analysis; this is what makes XML easy to create and use.

Unfortunately, this section can only scratch the surface of a rather large and important topic. Both within FrontPage's Help feature and on the Web, you can find far more details about XML and what you can do with it. Two good places to start are `http://www.softwareag.com/xml/about/starters.htm` and `http://www.w3schools.com/xml/default.asp`.

The Absolute Minimum

If the look of your Web pages when seen in Code view was a mystery before, you now have some of the keys to understanding that special Web language called HTML. After reading this chapter, you can understand what HTML does and why it's important for you to know of the basics, appreciate how tags often both open and close, and code some basic HTML formatting such as creating a bulleted list or providing a hyperlink to another page. You also can identify the most common scripting languages used to add new functionality to your site as well as understand how to download them and get instructions for using them. Lastly, you saw how to define XML, why it can be helpful, and where you can go to learn more.

With this solid introduction to some weighty subjects now digested, you're ready to embrace some of the many features possible for a site that FrontPage doesn't offer directly.