Introduction to HTML

Hypertext Markup Language, or HTML, is a markup language that enables you to structure and display content such as text, images, and links in Web pages. HTML is a very fast and efficient method of describing and formatting information that can be easily exchanged with other people. Because HTML was created to be a human-readable programming language, it is relatively easy to learn and does not require any special applications to create: you can design HTML pages in simple text editors, such as Notepad in Windows.

Web Servers and Clients

The World Wide Web consists of computers called Web servers that store Web pages, making them available to other Web servers as well as Web clients. *Web client* is a term used to describe an application, such as a Web browser, that is capable of viewing Web pages. The Web pages are made of plain-text files that contain HTML code and links to other content, such as images, music, and movies. Web pages are stored (or *hosted*) on Web servers. When a user enters the address of a page into his or her browser, the HTML file stored at that address transfers the content of the Web page over the Internet to the user's computer. The user's Web browser processes the HTML code and displays the Web page according to instructions found within the HTML code.

Standards

Although HTML was initially created by a few select organizations, the increasing popularity of the World Wide Web and HTML made it necessary to create an additional organization to ensure that HTML was written according to a standard; this would make HTML easier to implement, and help guarantee that a Web page would look much the same in all Web browsers. The World Wide Web Consortium (referred to as the *W3C* for short) is the organization that today oversees the development and standards of many Web-related technologies, including HTML: essentially the W3C decides what code will comprise the HTML specification. You can find more information about the W3C, and HTML, at the World Wide Web Consortium's Web site at www.w3c.com.

Elements

Web pages created using HTML consist of elements. Elements are the content items such as text, images, and movies that make up a Web page. The job of HTML is to tell Web browsers how to display these elements to the user. Text headings, a table containing lists of information, and an image banner are all examples of elements. Elements can contain other elements; for example, a table may contain elements such as text headings or images.

HTML Tags

HTML tags create the elements that comprise a Web page. For example, the `<p>` tag denotes text as a paragraph and the `<img>` tag indicates an image. Most HTML tags consist of two parts: an opening tag and a closing tag. You create both opening and closing tags in the same way: Each tag starts with the symbol for less than (`<`) and ends with the symbol for greater than (`>`). The name of the tag goes between the two brackets. In the closing tag, you precede the name of the tag with a forward slash (`/`). To create an HTML tag, then, you simply place the name of the tag in both the start and the end tag, and then specify the content of the element between the tags. For example, the name of the paragraph tag is simply `p`. Therefore, the opening tag of the paragraph tag is `<p>` and the closing tag is `</p>`. The text that will comprise the paragraph must be placed within the opening and starting HTML tags, as in `<p>This is some text</p>`. Some HTML tags do not require a closing tag. For example, the horizontal rule tag `<hr>` does not require a closing tag because it does not enclose any information: it simply generates a horizontal line on the Web page.

Tag Attributes

You can enhance HTML tags with the use of attributes. Attributes define additional parameters and characteristics for the HTML element. Although an HTML tag can create a paragraph of text, you can add attributes to the paragraph tag instructing the Web browser how to display that text. For example, an attribute can indicate how to align the text within that paragraph. Apart from straightforward characteristics such as setting the alignment and the color of an element, you can also use attributes to apply programming code such as JavaScript, or formatting information using Cascading Style Sheets, to the element. Insert attributes into the opening tag of the element; the attribute name is followed by the equal sign (`=`) and then the value of the attribute, which is enclosed in quotation marks. For example, to instruct the browser to center a paragraph on a Web page, add the `align` attribute with the value `center` to the `<p>` tag, as in `<p align="center">`. You do not need to make any other changes to the content of the element or the closing tag.

The Future of HTML

The latest version of HTML (HTML 4.01) is also the last version of the language. XHTML and Cascading Style Sheets (CSS) are intended to replace HTML. XHTML and CSS separate Web page content from any formatting instructions, making that content easier to manage. Although HTML contains code that describes and formats information, XHTML describes the structure of the content, and Cascading Style Sheets formats the information. This creates smaller XHTML documents that are easier for people to read, and faster for clients such as Web browsers to load and interpret (or parse). Although HTML will be replaced eventually by XHTML and CSS, HTML is still available and likely to remain on the World Wide Web for many years to come.

Introduction to XHTML

Extensible Hypertext Markup Language, or XHTML, is similar to HTML. If you know how to create Web pages using HTML, then you already know most of what you need to know about creating Web pages using XHTML. Although HTML can define both the structure and the appearance of a Web page, XHTML defines the structure of a Web page while relying on other technologies, such as Cascading Style Sheets (CSS), to specify the formatting information. XHTML is a markup language like HTML, but was made to conform to the XML standard.

XML

Extensible Markup Language, or XML, is a markup language that, like HTML, is used to create documents. Many applications and devices use the XML standard to exchange information, and sometimes between quite dissimilar objects — such as a database application and an application running on a mobile phone. XML is a widely used industry standard. XHTML was an attempt to create a language for constructing Web pages that conforms to the standards and principles of XML. You can think of XHTML as an attempt to rewrite the HTML language using XML.

Differences Between HTML and XHTML

There are a number of differences between HTML and XHTML, the most notable being that, unlike HTML, XHTML requires all tags to have a closing tag. Although most HTML tags have a closing tag, some do not, such as the break tag `<br>`. In XHTML these tags must have a closing tag. If the tag does not enclose any content, such as the `<br>` tag, you can add a forward slash (/) preceded by a space to the opening tag instead of using two tags, as in `<br />`. The tag `<br />` is the correct XHTML version of the HTML `<br>` tag. Although a number of major differences exist between HTML and XHTML, somebody learning XHTML who is already familiar with HTML would immediately notice only a small number of differences. One example is that all tag names are lowercase in XHTML. So, the HTML tag `<BODY>` is only valid in XHTML as `<body>`. All attribute names in XHTML must also be lowercase, and all of the values within tags must be enclosed in quotation marks.

XHTML Standards

Although XHTML is a single language, it consists of two major standards: XHTML Strict and XHTML Transitional. XHTML Strict is just that: it requires XHTML code to strictly follow the rules of the XML standard. XHTML Transitional is not as strict as XHTML Strict; XHTML Transitional was deliberately made to be less strict to help bridge the gap between the loose, more forgiving HTML standard of yesterday and the stricter, less forgiving XHTML standard of today. XHTML Transitional was made to be just that — transitory — so it is not a good idea to standardize on XHTML Transitional. XHTML Transitional is more like a short-term solution for quickly re-creating existing HTML pages in XHTML until an XHTML Strict version can be made.

Benefits of XHTML

XHTML allows developers who are specialized in creating content to focus solely on what they are good at: creating and compiling content. By using another technology, such as Cascading Style Sheets (CSS), to format the information, experts such as graphic designers and layout artists can focus solely on the appearance of information without getting bogged down in the content and meaning of the information. Separating information from its formatting also allows that information to be more easily accessed on dissimilar devices such as handheld computers, mobile phones, and television set top boxes. For example, you can create a Web page containing a company directory of personnel, and display the same Web page on a computer monitor, a handheld computer, or mobile phone, completely formatted for printing, without having to create multiple Web pages. Although the formatting instructions for the computers and the printer would be different, the underlying information would remain the same.

XHTML is far stricter in its syntax that HTML, which leads to less errors in your code and makes your Web pages accessible to different Web browsers and other Web-based tools, such as search engines. This strictness is necessary to standardize Web pages across the Internet, but it can result in a learning curve for people used to the more forgiving HTML standard.

XHTML retains the best ideas and features of HTML, but has been created so that XHTML can be improved in the future with minimal impact on existing Web pages. Unlike HTML, XHTML is extensible, meaning that it can easily be added to, and that changes to XHTML will not have to wait for Web browser manufacturers to implement the new features of XHTML before the benefits of these features are realized.

Disadvantages of XHTML

Because XHTML is stricter than HTML and requires the use of other technologies such as Cascading Style Sheets (CSS) to create Web pages, creating Web pages using XHTML will initially take more time. XHTML is also less forgiving than HTML when a Web page's code contains errors.

Many of today's Web developers use specialized tools to create Web pages. Any of the tools that are more than a couple of years old may not be able to create valid XHTML Web pages without upgrading or replacing the software.

Because XHTML uses CSS to format Web page elements, users must also learn how to implement CSS if they want their Web pages to resemble those created with just HTML.

When to Use XHTML

Generally, if you are creating Web pages that will exist on the Internet for a long period, you should use XHTML to ensure that your Web pages will be compatible with Web browsers of the future. If you are simply creating a list of this week's

fixtures for your local sporting club's Web page and do not need the information to be accessible for very long, you can safely use HTML instead of XHTML to structure and format your Web page.

Introduction to Cascading Style Sheets (CSS)

Cascading Style Sheets, or CSS, is a markup language used to specify instructions for displaying XHTML and HTML elements. Unlike HTML and XHTML, CSS is used exclusively for the formatting and display of information. Page background colors, font colors, and colors of hyperlinks on a Web page are the kinds of formatting that CSS controls. For example, you can use CSS to change the color of a paragraph of text and to describe where on the Web page that text is to be placed.

Style Sheet Properties

Style sheets are comprised of style sheet *rules* that apply to the elements of a Web page. Each style sheet rule consists of a *selector* and a *declaration block*. The selector indicates to which part of the Web page the style sheet will apply. For example, if the selector name in the style sheet rules is `p`, the style sheet rule will apply to all `<p>` tags found within the Web page. The declaration block consists of one or more *declarations* enclosed in curly brackets ({}). The declaration consists of a style sheet *property* and its value. A colon (:) separates the style sheet property name and the value of the style sheet property. Semicolons (;) separate declarations from each other. You must become familiar with only a few essential properties to effectively use style sheets. For example, the `color` property defines the color of an element. A simple style sheet rule that defines the `color` property of a paragraph of text is `p {color: blue}`.

Internal, External, and Local Style Sheets

There are a number of places where you can define the style sheets for your Web pages. The code for *internal* (or *embedded*) style sheets is stored within the code of the Web page that uses the style sheet. The Web browser does not display the style sheet itself; it applies the style sheet information to any elements within the Web page that have been instructed to use that information. *External* style sheets are style sheets that are saved in a file separate from the Web page code. The Web browser processes the Web page code first, sees from instructions in the Web page code that it needs to access the external style sheet file, and then retrieves and processes the information in the style sheet file. The major benefit of external style sheets is that you can use a single external style sheet to format multiple pages on your Web site. *Local* (or *inline*) style sheets consist of style sheet code that is applied to single elements within a Web page. Local style code is actually embedded within the individual tags of the Web page elements to which they apply.

Cascading Style Sheets

Because there are multiple ways of applying style sheet information to an element, a single element within a Web page can use style sheet information from external, internal, and local style sheets. If all of the style sheet properties being applied are different, then all the style sheet properties will be *cascaded* into one set of style sheet instructions and applied to the single element. This allows the Web developer great flexibility when creating style sheets for Web pages that have many different formatting requirements. For example, if a certain paragraph on a Web page sees a conflicting rule in external, internal, and local style sheets, the three style sheets are applied in a cascading fashion — one after the other. The local style sheet, which is applied last, prevails over the external and internal style sheets and determines the formatting of the element in question. So, if an external style sheet specifies that text on the Web page should be black, an internal style sheet specifies that text should be dark blue, and the local style sheet specifies that text should be white, the text will be white. This cascading effectively enables you to set one global style sheet for the entire Web page and then specify individual exceptions to this rule using internal or local style sheets.

Advantages of CSS

There are many benefits to using CSS to format the information on your Web pages. Style sheets give you enormous control over the appearance of information on your Web pages. You can use style sheets to precisely position elements on a Web page and to apply characteristics such as borders, colors, and backgrounds to individual elements. Using CSS to format the information on your Web pages can save you a large amount of work. You can create one style sheet that will determine the type of formatting used on all the Web pages in a Web site. For

example, if you have a Web site that contains hundreds or even thousands of Web pages, you can use a single style sheet to define the background color of all those Web pages instead of coding those formatting instructions into each individual page. Using a single CSS file also saves time when you want to make changes to Web pages that use the CSS file; for example, by changing a few lines of code within a single external style sheet, you can change the background color of all Web pages that use that style sheet.

Disadvantages of CSS

There are some disadvantages to CSS. When using style sheets, you have to be more organized and keep up-to-date information about your Web site. Because style sheets often involve the creation of separate files, you must be more careful when modifying, adding, or removing files from your Web server. You must also keep track of what styles are applied to which Web pages so that you do not inadvertently create style sheet rules that conflict, which can result in Web pages that do not appear as intended. Using external style sheets requires a Web browser to download an additional file from the Web server before a Web page displays, which can add precious seconds to loading times. Using local style sheet rules, while

quick and easy to implement, are more difficult to maintain, particularly if you have a large number of them. Many older Web browsers do not support CSS; and even if they do, they often support only a portion of the available CSS properties, which can lead to inconsistent results when your Web pages are viewed on these older browsers.

The benefits of using style sheets far outweigh the disadvantages. If you are contemplating creating a large amount of Web pages or intend to create Web pages in the future, you should take the time to learn the benefits that style sheets offer in creating Web pages.

Web Browsers

Web browsers are applications that display Web pages you create with HTML, XHTML, and CSS. Many different browsers are available, and most are free of charge. Almost all PC operating systems include a Web browser by default, and most people use the browser that came with their PC.

Compatibility

One of the major reasons to create standards for HTML, XHTML, and CSS is so that different Web browsers will display Web pages in the same way. As long as a Web browser can identify and render a Web page using the version of HTML, XHTML, and CSS that the Web page utilizes, the Web page will display to the user as the page's author intended.

Current Standards

XHTML, XHTML, and CSS use numbering schemes to identify their versions to Web browsers. Although new versions of these technologies will be available in the future as the technologies are improved and enhanced, any Web pages created with these versions should be compatible with current Web browsers.

Major Web Browsers

Starting with Netscape Navigator in the early- to mid-1990s, many Web browsers have come and gone. Today, only a few browsers remain viable. This section names and briefly discusses three of the more popular browsers.

Firefox

Firefox is a new, free Web browser that was designed to be fast, secure, and simple to use. Firefox includes powerful features such as tabbed browsing, multiple home pages, and an integrated search feature that lets you search major Web search engines such as Google, Amazon.com, and eBay. You can download Firefox from www.getfirefox.com.

Internet Explorer

Because of the popularity of the Microsoft Windows operating system, it should be no surprise that the most popular Web browser is currently Microsoft's Internet Explorer. Although IE is included with Windows, you can also download it from www.microsoft.com/ie.

Netscape Navigator

Netscape Navigator was one of the first commercial Web browsers available. Earlier versions of Netscape (versions 4 and prior), though still used by some people, do not support many modern Web standards. It is therefore a good idea to test all Web pages you design in both newer versions of Netscape (such as Netscape 7.2) and older versions (such as Netscape Communicator 4.8). Netscape is available free at www.netscape.com.

Versions

A few years ago, Web browser manufacturers would introduce new versions of their browsers on a frequent basis. Because of the standardization in HTML, XHTML, and CSS, Web browsers today typically offer a major upgrade once every year or two. You should always make sure that your Web pages are viewable with the most popular Web browsers at the current time.

Enhancements

In the early days of the Web, browser manufacturers would introduce enhancements to their Web browsers that only applied to Web pages viewed with that manufacturer's browser. This practice made writing Web pages difficult because designers had to figure out how to handle users who did not have the browser for which a certain Web site was optimized. In many cases, the users would receive a message saying the site was not supported by their browser. This was more of a concern in the past, before the standardization of HTML, XHTML, and CSS. This should not happen in the future, as most Web browser manufacturers now support most of the latest Web standards.

Creation Tools

Before you can create Web pages using HTML, XHTML, and CSS, you will need some sort of creation tools. Because these Web-based technologies store their information in plain-text files, it is possible to create Web pages and many related files with a simple text editor. Most of the coding examples in this book were created with a simple text editor that comes with the operating system. If you will be creating complex Web pages or need to create and maintain a large amount of Web pages, you may want to consider using an application made specifically for creating Web pages and style sheets. Using dedicated creation tools will reduce the amount of errors in your code and speed up the time required to create Web pages. Most Web page creation tools include not just the features required to create Web pages, but also other features such as spell checkers, file transfer utilities, and code formatting features that make it easier to maintain, manage, and deploy Web pages.

Types of Creation Tools

Text Editors

There are many different types of text editors available, ranging in features from the very sophisticated to the very simple. Almost all operating systems include their own text editor that you can use to create Web pages (Windows comes with Notepad, for example). Many experienced Web development professionals use text editors to create some or all of their Web pages. Most Web servers also have text editors available, enabling you to quickly update your Web pages from where they reside on the server.

HTML Editors

HTML editors make it easier for you to create Web pages using HTML and XHTML. Standard features include quick insertion of popular tags, automatic checking that you are using the appropriate tags, and utilities that format your code with spaces and indents for easier readability. If you intend to create a large number of Web pages, you should consider using an HTML editor.

WYSIWYG Editors

WYSIWYG is an abbreviation for "what you see is what you get." WYSIWYG editors are Web page creation tools that enable you to create Web pages without having to learn any of the underlying language (such as HTML). Information is placed on a Web page in the same manner that you would place information in a word processing document. You can type text and paste images right into the Web page; the application will then generate the required HTML, XHTML, and style sheet code. WYSIWYG editors are quite suitable for users who do not want to learn HTML, XHTML, and CSS coding and only need to create a small number of Web pages.

Creation Tools

1st Page 2000

1st Page 2000 is a professional HTML editing application that works just as well for people learning to create Web pages as it does for experienced professionals. It is available on the World Wide Web at www.eversoft.com.

HomeSite

HomeSite is a comprehensive HTML, XHTML, and style sheet editor available from Macromedia. HomeSite has been widely used by Web development professionals for many years. More information about HomeSite is available on the World Wide Web at www.macromedia.com.

TopStyle

TopStyle is a robust Web page editor that excels in the creation and maintenance of style sheets. You can use TopStyle by itself to create Web pages, or integrate it with your other Web page development applications to create and maintain style sheets. For more information about TopStyle, visit www.bradsoft.com.

UltraEdit

UltraEdit is a superb text editor used by many Web developers who prefer working with raw HTML, XHTML, and style sheet code. Although it is a text editor, UltraEdit includes many features that make working with Web pages easier. For more information about UltraEdit, visit www.ultraedit.com.