

WEB PAGE DESIGN AND CREATION

B

Most sites on the Web consist of a series of pages. Some of these pages are “static,” meaning they are created ahead of time (like a Microsoft Word document or a text file) and stored at the site. Anyone who accesses a static Web page is going to see the same thing. Other pages delivered by a Web site are “dynamic.” This means that the page content is created by a computer program that is run every time the page is accessed. Thus, the contents can vary from one user to the next or from one time to the next.

Regardless of whether a Web page is static or dynamic, the language used to create the page is the same. This language is called the **hypertext markup language**, or **HTML** for short. This appendix discusses HTML basics and examines how static Web pages are created. Appendix C examines various scripting languages and programming interfaces that are used to create dynamic pages and to add higher levels of interactivity than that provided by standard HTML.

HTML provides the means to control both the contents and appearance of a page. If a site consists of a few pages, then standard HTML will suffice. For larger sites, maintenance is simplified if the contents of the pages can be treated separately from their appearance. This is where newer standards such as cascading style sheets (CSS) and the extensible markup language (XML) come into play. Both of these standards are also briefly explored in this appendix.

hypertext markup language (HTML)

The language used to create Web pages.

HTML

When Tim Berners-Lee first conceived the World Wide Web in 1989, he was searching for a formal language that could be used to create and link text documents in a distributed network. A colleague, Anders Berglund, advised him to use a “SGML-like” syntax. At the time, SGML (Standard Graphic Markup Language) was a well-established but highly complicated markup language used for managing complex documents. Berners-Lee knew that SGML was too complex for the average researcher to use. What he took from SGML was its use of “markup” tags. The end result was a highly simplified markup language that he called the hypertext markup language.

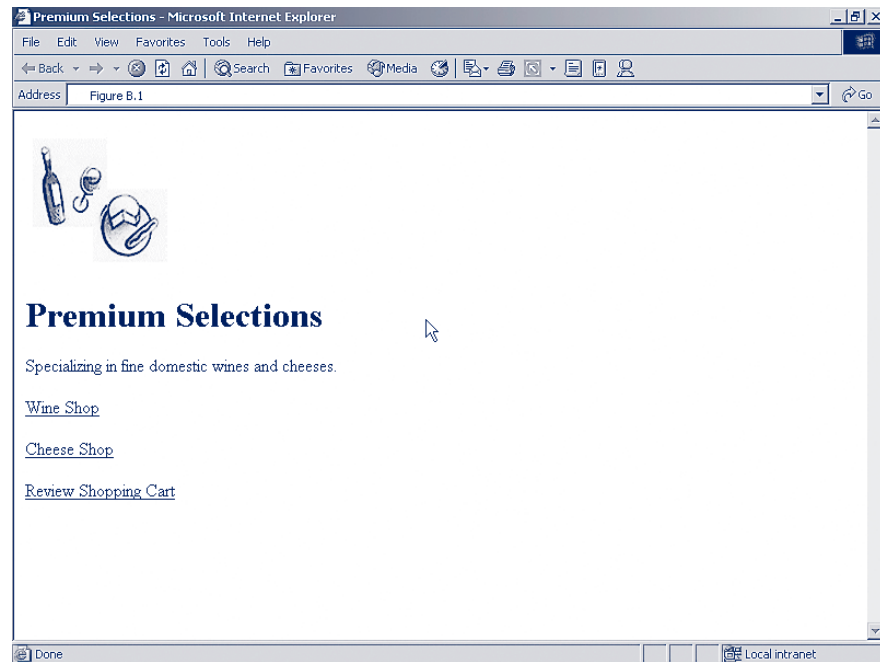
A SIMPLE EXAMPLE

Virtually every page on the Web has an underlying text (ASCII) file containing markup tags describing the structure and content of the page. When you view a particular page, you are seeing your browser’s rendering or interpretation of those tags. For instance, consider the sample Web page shown in Exhibit B.1 (next page). This is the home page (starting page) of a Web site (premiumselect.com) for a hypothetical merchant called Premium Selections. Throughout this appendix, this simple site will serve to illustrate various features of HTML and Web page construction. This particular home page links readers to three other static pages—one containing selections from the wine shop, another containing selections from the cheese shop, and a third that enables consumers to review the wines and cheeses they have placed in their electronic shopping cart and to purchase those selections.

The HTML text file underlying this home page is shown in Exhibit B.2 (next page). Like other static Web pages, this page consists of a series of tags. Most, but not all, tags come in pairs. For example, in Exhibit B.2 the `<H1> . . . </H1>` tags indicate that the enclosed text is to be treated as a “number 1 headline.” If you want to see the HTML tags underlying any particular page on the Web, access the page, click the “View” menu at the top of the browser, then select either “Source” in Microsoft’s Internet Explorer or “Page Source” in Netscape’s Navigator. A window will appear containing the HTML code for the page.

Before we look in detail at the syntax and use of specific HTML tags, let’s briefly review those shown in Exhibit B.2.

1. The `<HTML> . . . </HTML>` tags that surround the other tags on the page basically tell the browser that this is a Web page.
2. The head (`<HEAD> . . . </HEAD>`) is the first component of the page. The head contains a title (`<TITLE>title text</TITLE>`). The text of the title is displayed on the title bar at the top of the browser. In this case, the words “Premium Selections” will appear on the title bar.
3. The second major component of the page is its body (`<BODY> . . . </BODY>`). In this example, the body tag specifies various color parameters for the page. Included among these parameters are the

EXHIBIT B.1 Home Page for Premium Selections**EXHIBIT B.2 Sample Web Page Code**

```

<HTML>

<HEAD>
<TITLE>Premium Selections</TITLE>
</HEAD>

<BODY BGCOLOR="white" TEXT="black" LINK="black">

<P>
<IMG SRC="www.premiumselect.com/companylogo.gif">
</P>

<H1>Premium Selections</H1>

Specializing in fine domestic wines and cheeses.

<P>
<A HREF="www.premiumselect.com/wines.htm">Wine Shop</A>
</P>

<P>
<A HREF="www.premiumselect.com/cheeses.htm">Cheese Shop</A>
</P>

<P>
<A HREF="www.premiumselect.com/shoppingcart.exe">Review Shopping Cart</A>
</P>

</BODY>
</HTML>

```

page's background color (BGCOLOR="white"), the color of its text (TEXT="black"), and the color of the hypertext links (LINK="black") on the page. The body of the page is where you find the real contents—in this case, a company logo, some headline text, some descriptive text, and three hypertext links pointing to other pages at the site.

4. At the top of the page is an image of the company's logo. The image is specified by an `` tag. Like the body tag, the image tag usually contains a series of parameters delimiting various features of the image. In this case, there is only one parameter—the "source" of the image or, more specifically, the name of the image file (SRC= "www.premiumselect.com/companylogo.gif").
5. The logo is followed by a headline specified by a headline tag (`<H1>headline text</H1>`). Headlines come in various sizes, from H1 to H6. The number determines the size and style of the font used to display the headline text. The smaller the number is, the larger the font. The browser determines the specific font that is used with a particular number.
6. After the headlines comes simple descriptive text ("Specializing in fine domestic wines and cheeses"). As the text does not have a tag, it will be displayed as is (again, with the font determined by the browser).
7. Next is a series of three hypertext links each specified with an "anchor" tag (`<A>some text or an image`). Each of the anchor tags has a hypertext reference (HREF) that designates the page that will be returned when a user clicks the anchor. For instance, when the user clicks on the first anchor (`Wine Shop`), the page "wines.htm" will be retrieved. Anchors appear on a page in the form of text or an image. In this example, the anchors are text. How is the text of an anchor distinguished from regular text? There are two ways. First, the text of an anchor is underlined and often has a different color than regular text. Second, when the mouse cursor is over an anchor, the shape of the cursor usually changes from an arrow to a hand.
8. Finally, there are a series of paragraph tags (`<P> . . . </P>`) surrounding the image and anchor tags. These paragraph tags simply serve to add line spacing around the image and to anchor the text. If they were not there, everything on the page would run together.

At this point, you might want to experiment a little bit with this example. To do so, activate a simple text editor (such as Microsoft's Notepad). Type the text that appears in Exhibit B.2. However, leave out the `` tag and the paragraph tags surrounding the image. Name the file something like "myfile.htm" and save it to a subdirectory on your machine (Note: If you are using Notepad, the first time you save the file you will have to save it as a File Type of "All Files" rather than as a File Type of "Text Document"). After you save the file, do not close the text editor. Open your Web browser. Once it is open, click on the "File" menu at the top of the browser. Then select "Open" from the File menu. A file selection dialog box will appear. Find the file you saved and open it. If you have not made any mistakes, you should see something similar to Exhibit B.1, but without the logo. We are now ready for some experimentation. Go back to the text editor and try some of the changes in the following list:

1. Change BGCOLOR from "white" to "blue," the TEXT from "black" to "white," and LINK from "black" to "yellow." Save the file but do not close the editor. Return to your browser and click on the "View" menu and select "Refresh." The colors on the page should change.
2. Change the headline tags from a "number 1 headline" (`<H1> . . . </H1>`) to a "number 2 headline" (`<H2> . . . </H2>`). Save the file but do not close the editor. Return to your browser and click on the "View" menu and select "Refresh." The font size of the headline should change.
3. Remove the paragraph tags surrounding the anchor tags. Save the file but do not close the editor. Return to your browser and click on the "View" menu and select "Refresh." All of the anchor tags should appear on a single line.

In the remainder of this appendix you will be exposed to a variety of other tags that can be used to create both simple and complex pages.

OVERVIEW OF HTML TAGS

The original version of HTML (version 1.0) consisted of a small set of tags that made it possible to create primitive Web pages consisting of basic text and images. Compared to the features offered by existing word processors (e.g., Microsoft Word), the tags in HTML 1.0 made it difficult, if not impossible, to:

- ▶ Control font sizes and styles
- ▶ Place text and images at specific points on a page
- ▶ Display data and text in tables
- ▶ Create forms for entering data that could be sent to a Web server for processing

The fact that the initial version of HTML lacked these capabilities did little to deter the growth of the Web. In fact, the combination of HTML 1.0 and the early Web browsers, such as Mosaic, resulted in a graphical user interface (GUI) that spurred tremendous growth of the Web. The simple standards made it easy for Web authors to create pages and for Web readers to access those pages.

With growth came increased demands from Web authors for new tags that addressed the limitations in HTML 1.0. From 1994 to the present, the HTML standard, which is now controlled by the World Wide Web Consortium (W3C at w3.org), has undergone a wide variety of changes. The current standard is HTML 4.01. The W3C has reformulated HTML 4.01 in XML (see the discussion on XML later in the appendix) and has produced a recommended standard called XHTML 1.0, which is not widely used at the moment.

In this short appendix there is no way to do justice to the myriad of tags in HTML 4.01 and the variety of methods and techniques used to create the sophisticated pages found across the Web. If you are interested in understanding the current HTML standard, you should look at Castro (2002). A number of Web sites offer tutorials demonstrating the standard (e.g., w3schools.com). If you are interested in Web site design, then a book like Willard (2001) will fit the bill. Because it is possible to list the HTML source behind any Web page, another way to learn about design ideas and the use of various tags is to browse the Web and view the source code for pages that look interesting. If you decide to use someone else's source on your own pages, you will need to be careful about what you use. Certain images, text, and general designs are trademarked and copyright protected.

Although HTML 4.01 makes it possible to create very sophisticated Web pages, it is important to note that many of the most successful sites rely on a handful of tags and very simple designs. In this way, the pages can be quickly downloaded over even the slowest modems and viewed in virtually any Web browser. Exhibit B.3 lists some of the basic tags used with many of today's sites. For example, the pages on Yahoo! are constructed (almost) solely from the tags in this table. Rather than detailing each of the individual tags in Exhibit B.3, in the discussion that follows, we look at the use of images, tables, frames, and forms.

LINKING PAGES WITH ANCHOR TAGS

Anchor tags provide the hypertext links that enable users to “surf” the Web from one site or page to another. On a Web page, an anchor appears in the form of either underlined text or an image (sometimes denoted with a border around it). In either case, when the mouse cursor moves over an anchor, the cursor changes shape, indicating that a click will retrieve the linked page. The syntax of an anchor tag is `anchor text or image`, where “URL” represents the Web address of a linked Web page or another location on the same Web page. For example, in Exhibit B.2 the anchor tag

```
<A HREF=“www.premiumselect.com/wines.htm”>Wine Shop</A>
```

is displayed on the page as the underlined text “Wine Shop.” When this anchor is clicked, a request will be sent to the Web server “www.premiumselect.com” to retrieve the HTML page “wines.htm.”

Anchors can also be used to tell a Web server to run a program. At an e-commerce site, the program might be used to collect marketing data from a potential consumer and store it in a database for later use, display a particular catalog page depending on a consumer's query, or process a consumer's purchase order. Usually, a program will require some input information in order to run. For instance, to process a purchase order, the program will need the buyer's name, address, credit card number, items and quantities being purchased, and so on. This information can be specified with the anchor tag or collected from a Web form. Once a program is run, the program will dynamically produce a response in the form of an HTML page that will be returned by the Web server.

In order to run a program, the syntax of the anchor needs to look like: `anchor text or image`. For example, in Exhibit B.2, when the `Review Shopping Cart` anchor is clicked, the “www.premiumselect.com” Web server will run a program called “shoppingcart.exe” that will show a potential buyer the purchases they have selected up to that point in time.

ADDING IMAGES TO A PAGE

One of the major attractions of the Web is the integration of text and images. The Web would be a rather unexciting place if it were simply “hypertext” rather than “hypermedia.” In an HTML page, images serve a variety of functions, such as:

- ▶ Embellishing the aesthetics of a page
- ▶ Enhancing the information and data contained in the page

EXHIBIT B.3 Basic HTML Tags	
Type	HTML Tags
Text formatting	<code>bold</code> <code>emphasis</code> <code><H#>header1</H#></code> where # is 1 to 6 <code> text font</code>
Positions, paragraphs, spacing	<code><CENTER>centered content</CENTER></code> <code>
 line break</code> <code><P>paragraph</P></code> <code><DIV>divided content</DIV></code> <code>span of content</code>
Tables	<code><TABLE>table content </TABLE></code> <code><TR>table row</TR></code> <code><TH>table header</TH></code> <code><TD>table data</TD></code>
Lists	<code>unordered list</code> <code>ordered list</code> <code>list item</code>
Form	<code><FORM>form content</FORM></code> <code><INPUT> input area </INPUT></code> <code><TEXT> text input box </TEXT></code> <code><TEXT AREA> multiline text input </TEXT AREA></code> <code><SELECT>list of selections</SELECT></code>
Document	<code><HTML>html document</HTML></code> <code><HEAD>doc head</HEAD></code> <code><BODY>doc body</BODY></code> <code><TITLE>doc title</TITLE></code>
Graphics	<code> graphic image</code> <code><HR> horizontal rule/line</code>
Linking and anchor	<code><A HREF>anchor content</code>

- Serving as buttons or icons linking one page to another
- Focusing or attracting attention to particular areas of a page (e.g., an animated advertisement)

As noted in our earlier example, images are incorporated in a page with the `` tag whose syntax is:

```
<IMG SRC= "image file" image parameters>
```

where the “image file” is the name of the image file and “image parameters” represents a list of potential parameters specifying features such as the image’s `WIDTH`, `HEIGHT`, `ALIGNMENT`, and so on.

A wide variety of image formats are supported on the Web. The two most popular formats are GIF (.gif files) and JPEG (.jpg files). JPEG is used when higher quality images are required. A variant of the GIF format is the animated GIF. An animated GIF file contains a series of pictures or graphics that are displayed in a designated order at a specified timing (almost like the “cells” in an animated cartoon). Animated GIFs have become the standard image format for the banner ads that appear on many Web pages.

To use an image as a hypertext button or icon to link one page to another, the image tag must be embedded within an anchor tag. As an example, consider Exhibit B.4 (next page) and the associated HTML file shown in Exhibit B.5 on the next page. Again, there are three anchors on the page. Each of the anchors is denoted with image and text. For example, the first anchor, which links to the Wine Shop, has an image `` followed by some text (“Wine

EXHIBIT B.4 Images as Anchors for Hypertext Linking

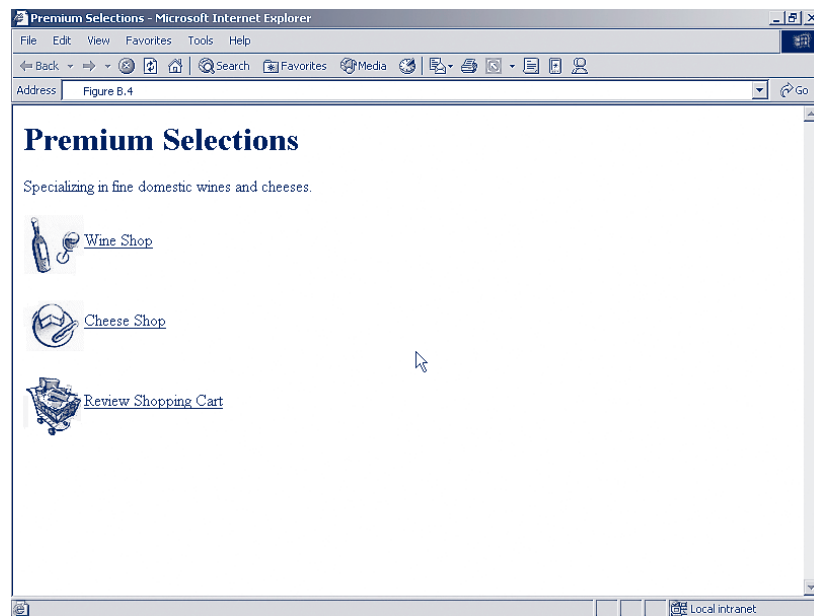


EXHIBIT B.5 HTML Tags for Image Anchor

```
<HTML>

<HEAD>
<TITLE>Premium Selections</TITLE>
</HEAD>

<BODY BGCOLOR="white" TEXT="black" LINK="black">

<H1>Premium Selections</H1>

Specializing in fine domestic wines and cheeses.

<P>
<A HREF="wines.htm"><IMG SRC="wines.gif" BORDER=0 ALIGN=middle>Wine Shop</A>
</P>

<P>
<A HREF="cheeses.htm"><IMG SRC="cheeses.gif" BORDER=0 ALIGN=middle>Cheese Shop</A>
</P>

<P>
<A HREF="shoppingcart.exe"><IMG SRC="shopcart.gif" BORDER=0 ALIGN=middle>Review Shopping Cart</A>
</P>

</BODY>
</HTML>
```

Shop”). In this case, the image has two additional parameters. The first says that the image should be displayed without a border (BORDER=0). The other option is BORDER=1. The second parameter (ALIGN=MIDDLE) states that the middle of the image should be lined up with the text. Some other options are ALIGN=TOP or ALIGN=BOTTOM.

DISPLAYING DATA AND CONTROLLING PAGE LAYOUT WITH TABLES

In HTML, tables have two purposes—to display numbers and text in tabular form and to control the layout of various sections and components on a page. Exhibit B.6 outlines the basic structure of an HTML table.

To understand the rudiments of HTML tables (see Exhibit B.6), you need to know that HTML tables (<TABLE> ... </TABLE>):

- ▶ Consist of three basic elements: rows (<TR> ... </TR>), headers (<TH> ... </TH>), and cells (<TD> ... </TD>)
- ▶ Are laid out row-by-row, starting with the row of column headers at the top of a table
- ▶ Have headers and cells that can contain virtually anything, including text, images, anchors, other tables, or some combination of these
- ▶ Can be displayed with or without grid lines, which in a table are called “borders” and are specified as being “on” <TABLE BORDER=1> or “off” <TABLE BORDER=0>

If you think of a page as a table of cells where the borders between the cells are turned off, then you begin to see how tables can be used to control the layout of a page by simply putting text and images within particular cells of the table.

Tables can become very complicated. Tables can contain other tables, and rows and columns can span one another. In this discussion we will stick with the basics. To illustrate the basics, first examine the page shown in Exhibit B.7 (next page). This is the “Wine Shop” page for the Premium Selections site. The associated HTML file is shown in Exhibit B.8 (next page). This page does not use tables. Instead, a series of paragraph tags (<P> ... </P>) and break tags (
) are interspersed throughout the page to control its layout. Besides the fact that the page looks a little boring, it is also difficult to follow the information. If there were more than a handful of wines (certainly the case for a real online wine store such as the Virtual Vineyard) it would be very difficult to understand the information.

Look what happens when the same information is placed in a table. The results are shown in Exhibit B.9 (page 9) with the underlying HTML in Exhibit B.10 (page 9).

As Exhibit B.10 shows, the table is specified between the <TABLE BORDER=1 CELLPADDING=5> and </TABLE> tags. The table is displayed with grid lines because BORDER=1. The headers and cells of the table have a little extra space because CELLPADDING=5 has been added. The first row of the table consists of the column headers (i.e., Name, Description, etc.). Next comes the row of data for the Merlot wine. This is followed by the row of data for the Chardonnay wine.

Even though HTML 4.01 provides specific mechanisms for placing components at particular locations on a page, HTML tables still remain the most popular method for controlling the placement of various objects, including images, on a Web page.

EXHIBIT B.6 Elements of an HTML Table

Table begin <TABLE>

Table Row Begin <TR>	Column Header <TH>...</TH>	Column Header <TH>...</TH>	...	Column Header <TH>...</TH>	Table Row End </TR>
Table Row Begin <TR>	Row Header <TH>...</TH>	Cell <TD>...</TD>	...	Cell <TD>...</TD>	Table Row End </TR>
...
Table Row <TR>	Row Header <TH>...</TH>	Cell <TD>...</TD>	...	Cell <TD>...</TD>	Table Row End </TR>

Table End </TABLE>

EXHIBIT B.7 Displaying Text and Data Without HTML Tables

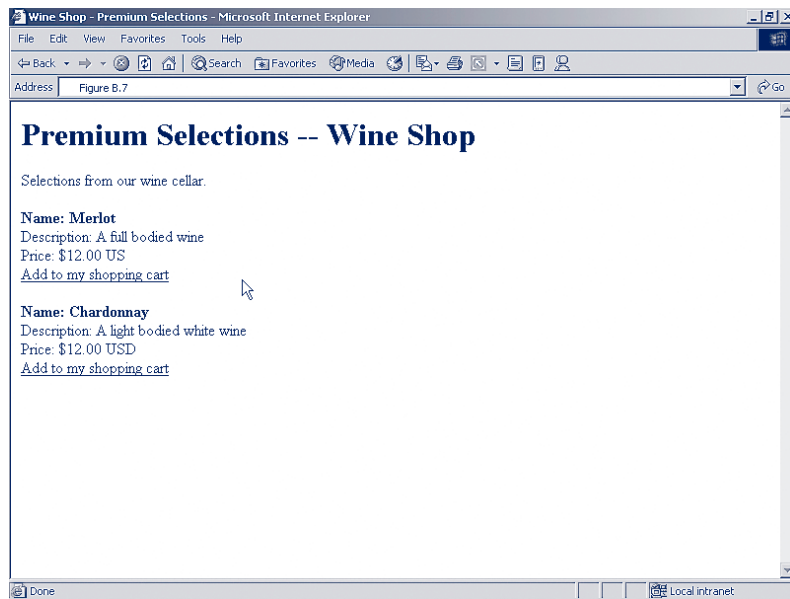
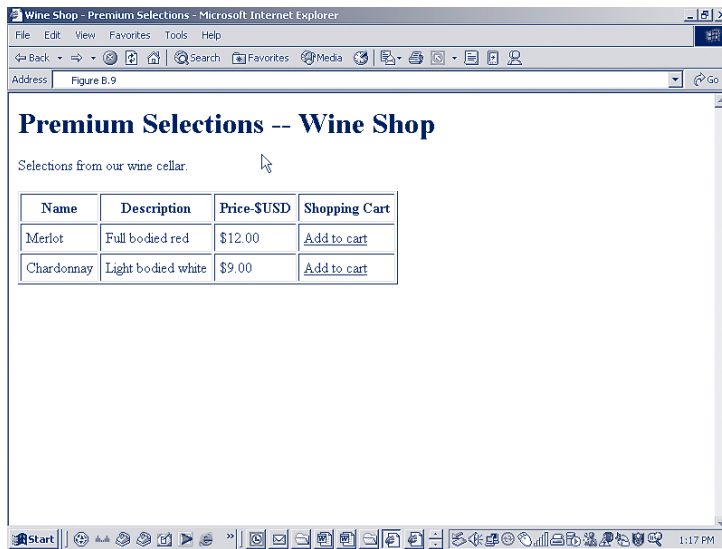


EXHIBIT B.8 HTML Tags for Data and Text Display

```
<HTML>
<HEAD>
<TITLE>Wine Shop - Premium Selections</TITLE>
</HEAD>
<BODY BGCOLOR="white" TEXT="black" LINK="black">
<H1>Premium Selections - Wine Shop</H1>
<P>Selections from our wine cellar</P>
<B>Name: Merlot</B>
<BR>
Description: A full bodied wine
<BR>
Price: $12.00 US
<BR>
<A HREF="www.premiumselect.com/shoppingcart.exe?wine=merlot">Add to my shopping cart</A>
<BR>
<P>
<B>Name: Chardonnay</B>
<BR>
Description: A light bodied white wine
<BR>
Price: $12.00 US
<BR>
<A HREF="www.premiumselect.com/shoppingcart.exe?wine=chardonnay">Add to my shopping cart</A>
<BR>
</BODY>
</HTML>
```


EXHIBIT B.9 Web Page with HTML Table**EXHIBIT B.10 HTML Tags for Tabular Display of Text and Data**

```

<HTML>
<HEAD>
<TITLE>Wine Shop - Premium Selections</TITLE>
</HEAD>
<BODY BGCOLOR="white" TEXT="black" LINK="black">
<H1>Premium Selections - Wine Shop</H1>
<P>Selections from our wine cellar</P>
<TABLE BORDER=1 CELLPADDING=5>
<TR>
<TH>Name</TH>
<TH>Description</TH>
<TH>Price-$US</TH>
<TH>Shopping Cart</TH>
</TR>
<TR>
<TD>Merlot</TD>
<TD>Full bodied red</TD>
<TD>$12.00</TD>
<TD><A HREF="www.premiumselect.com/shoppingcart.exe?wine=merlot">Add to cart</A></TD>
</TR>
<TR>
<TD>Chardonnay</TD>
<TD>Light bodied white</TD>
<TD>$9.00</TD>
<TD><A HREF="www.premiumselect.com/shoppingcart.exe?wine=chardonnay">Add to cart</A></TD>
</TR>
</TABLE>
</BODY>
</HTML>

```

CONTROLLING NAVIGATION WITH FRAMES

Frames divide an HTML page into a series of well-defined, independent segments, much like the panes in a window. On many Web pages with frames, one of the frames (typically on the left side or top of the page) is used as a menu or table of contents for navigating from one page in the Web site to the next. Another frame displays the contents of the page selected from the menu.

As an example, consider the page shown in Exhibit B.11 and the accompanying code shown in Exhibit B.12. When the page displayed in Exhibit B.11 is first opened, an “index” frame is displayed on the left and the “view” frame is displayed on the right. The content of the “index” frame comes from a page labeled “index.htm,” whereas the content of the “view” frame comes from a page labeled “start.htm.” When a user clicks on one of the selections in the “index” frame (say “Wine Shop”), the associated Web page (here “wines.htm”) is displayed in the “view” frame.

Some of the more interesting pages on the Web use frames. Unfortunately, if you try to view the HTML source for these pages, you will encounter a problem. The only thing that will be displayed is the HTML code used to specify the frames, not their content. For instance, if you tried to use your browser to look at the source for the page in Exhibit B.11, you would see the source in Exhibit B.12. You would not see the source for either “index.htm” or “start.htm,” which are the pages that are actually shown in the frames.

In the past, the use of frames was controversial because they were not supported by all browsers. Today, this is no longer the case. They are used extensively through the Web.

EXHIBIT B.11 Web Page with Frames

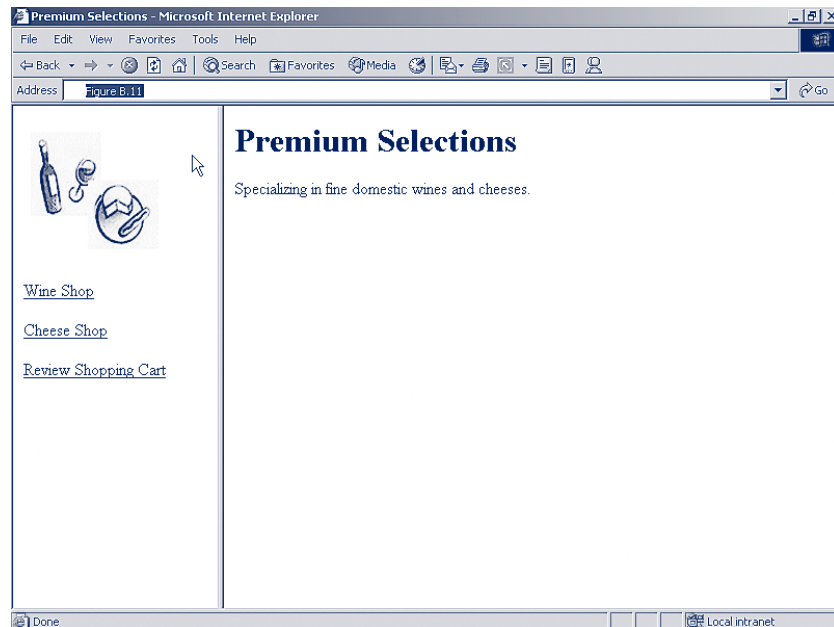


EXHIBIT B.12 HTML Tags for Frames

```
<HTML>
<HEAD>
<TITLE>Premium Selections</TITLE>
</HEAD>
<FRAMESET COLS="25%,*">
  <FRAME SRC="index.htm" NAME=index>
  <FRAME SRC="start.htm" NAME=view>
</FRAMESET>
</HTML>
```

SENDING INFORMATION TO THE WEB: URLS VERSUS FORMS

How does a Web browser send information to a Web server? One way is by attaching “key-value” pairs to the end of a URL in an anchor tag. For example, reconsider the listing in Exhibit B.9. Here, the user can add a wine to their shopping cart by clicking on the words “Add to Cart.” How does the Web server at the Web site “www.premiumselect.com” know that the user has selected the Merlot or Chardonnay? Look carefully at the underlying anchor tags:

```
<A HREF="www.premiumselect.com/shoppingcart.exe?wine=merlot">Add to Cart</A>
<A HREF="www.premiumselect.com/shoppingcart.exe?wine=chardonnay">Add to Cart</A>
```

When the user clicks on either of these anchors, a request is sent to the Web server to run a program called “shoppingcart.exe.” A key-value pair (i.e., wine=merlot or wine=chardonnay) is attached to the request after the “?” mark so that the program knows the type of wine to add to the shopping cart.

Passing parameters as key-value pairs attached to a URL works fine when there is a small set of parameters and the specific values for those parameters are known in advance. However, most of the time there are a number of parameters and values that are not known in advance. For instance, in our example, suppose we want to give the user the option to specify the number of bottles to add to the cart. It is not possible to know the number ahead of time. This is where HTML forms come into play. The page shown in Exhibit B.13 provides users with a form that allows them to designate the number of bottles of merlot they want to purchase. The HTML underlying this form is displayed in Exhibit B.14 (next page). The various elements in the form are contained between the beginning <FORM> tag and the associated ending </FORM> tag. This form has three elements. First is the “INPUT TYPE=TEXT” fields where the user types in the number of bottles of each type of wine to be purchased. Next, there is a “SUBMIT” button (i.e., INPUT TYPE=SUBMIT). If the user types in “2” for the number of bottles of merlot, when the user clicks the submit button the key-value pairs “quantity_merlot=2” and “quantity_chardonnay=0” will be sent to the “www.premiumselect.com” server, which is designated in the “ACTION” field of the <FORM> tag.

Again, there are a wide variety of form elements, including radio buttons, check boxes, selection lists, text input fields, submit buttons, regular buttons, images, text areas, password input boxes, and reset buttons. A detailed discussion of these elements is beyond the scope of this appendix. Readers are referred to Castro (2002) for a complete discussion.

EXHIBIT B.13 Web Page with Forms

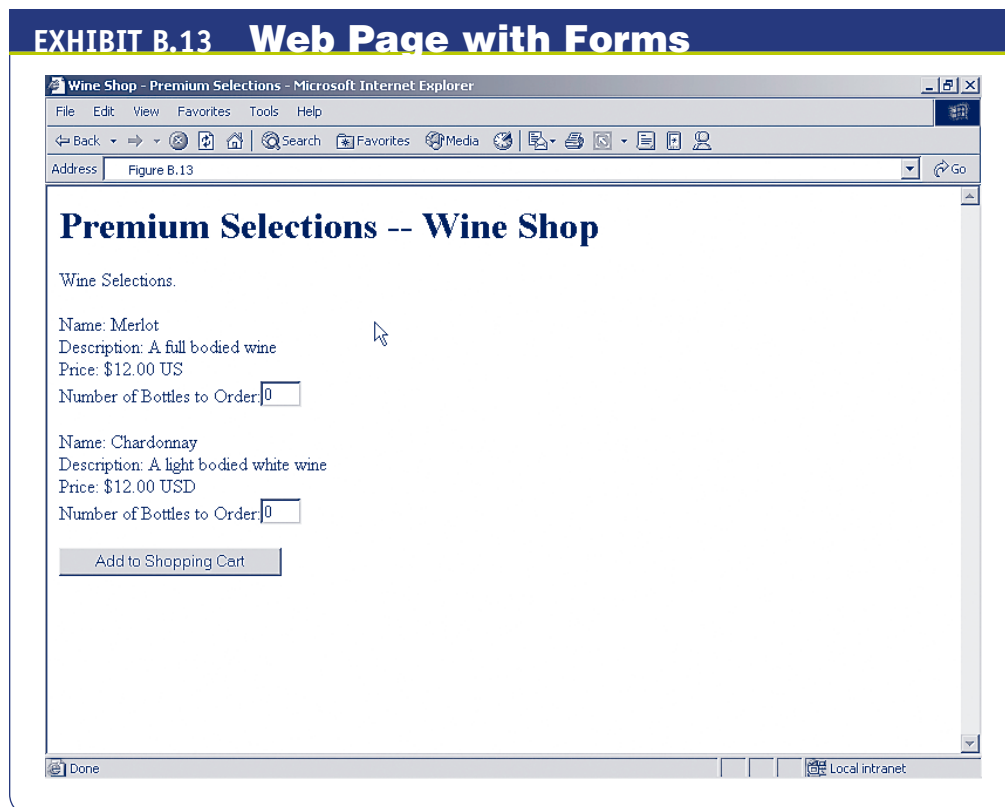


EXHIBIT B.14 Tags for Forms

```

<HTML>

<HEAD>
<TITLE>Wine Shop - Premium Selections</TITLE>
</HEAD>

<BODY BGCOLOR="white" TEXT="black" LINK="black">

<H1>Premium Selections—Wine Shop</H1>

<P>Wine Selections.</P>

<FORM ACTION="www.premium.select" METHOD=POST>

Name: Merlot<BR>
Description: A full bodied wine<BR>
Price: $12.00 US<BR>
Number of Bottles to Order:
<INPUT TYPE=TEXT SIZE=2 NAME=quantity_merlot VALUE="0">
<P>

Name: Chardonnay<BR>
Description: A light bodied white wine<BR>
Price: $12.00 US<BR>
Number of Bottles to Order:
<INPUT TYPE=TEXT SIZE=2 NAME=quantity_chardonnay VALUE="0">
<P>

<INPUT TYPE=SUBMIT VALUE="Add to Shopping Cart">
</P>

</FORM>

</BODY>
</HTML>

```

HTML EDITORS

With the earlier versions of HTML, it was easy to create Web pages with a basic text editor (e.g., Microsoft's Notepad) by manually typing in the content as well as the appropriate HTML tags. Although many Web designers still rely on these simple tools, a larger number employ specialized HTML editors to create pages.

The editors are of two sorts. First are the specialized text editors. Among other features, these editors provide toolbars that enable easy insertion of common HTML tags. This overcomes the problem of remembering the names and syntax associated with the various tags, but it still requires manual entry. Included among this class of editors are the following commercial products:

- ▶ Hotdog Professional (sausagetools.com)
- ▶ UltraEdit (ultraedit.com)
- ▶ EditPlus (editplus.com)

The second type of editors are the Web authoring tools that support WYSIWIG ("what you see is what you get") visual design and layout of Web pages. Once a page is arranged, the underlying HTML is automatically generated. In addition to visual layout, these programs also support:

- ▶ Editing of the underlying HTML source code generated from the visual layout
- ▶ Creating and maintaining an entire site in addition to individual pages
- ▶ Editing and creating of images
- ▶ Querying and updating databases

Among the better known authoring tools are:

- ▶ Microsoft FrontPage (microsoft.com/frontpage)
- ▶ Adobe GoLive (adobe.com/golive/main.html)
- ▶ Macromedia Dreamweaver MX (macromedia.com/software/dreamweaver)

Although the authoring tools make it easier to create aesthetically pleasing pages, designers still need to understand HTML in order to modify and maintain existing pages, especially given the differences that still exist between the standards supported by the two main Web browsers—Netscape and Internet Explorer.

CASCADING STYLE SHEETS

Regardless of their function, most e-commerce sites contain hundreds, if not thousands, of pages. It usually takes a group of people to create and maintain these pages. Even with a group of people, crafting these pages by hand is an impossible task. Take the Web site, amazon.com, as an example. Amazon.com sells thousands of items—books, CDs, electronic devices, and so on. As you move through the site, the pages displaying the merchandise are basically the same. They have a common structure and appearance. If Amazon.com only sold a small number of products, then they might be able to manually create their pages ahead of time. However, they are unable to do so. For this reason, the contents of their pages—item names, descriptions, prices, customer comments, and so on—are stored in a database, the structure and appearance of the pages are controlled by a set of template pages, and the pages are generated automatically by a computer program or server-side script. (You will learn more about scripting in Appendix C.)

It is somewhat difficult to describe the exact structure of a template. Basically, you can think of a template as a Web page with a set of placeholders for the content. At the Amazon.com Web site, when a consumer moves to a page displaying a particular product, the page is dynamically produced by a computer program (or server-side script) that retrieves the product information from the product database, accesses the appropriate template for that page, and fills in the place holders with the appropriate content. The same strategy is used at most of the large sites on the Web.

In order to use this strategy, the contents of a page must be separated from its structure and appearance. Once this is done, templates can be used to create the look and feel of the pages. In setting up the templates, the people responsible for creating and maintaining the templates have to decide on a common look and feel for the site. More specifically, they have to decide on the appearance of the elements making up the pages—the headers, tables, form elements, buttons, paragraphs, and so on. In the vernacular of document processing, they have to decide on the style of the elements, including such features as colors, fonts, alignment, and the like.

It is possible to control the style of various elements using standard HTML, but doing so presents some problems. First, the function, structure, and appearance associated with many of the HTML tags are all intertwined. For example, header tags (e.g., <H1>) take a certain font size. The specific size that the headers take will depend on the different browsers and on the preferences set by individual users. Second, only a few HTML tags can be used to directly control the appearance of various elements in a page. One of these is the tag. This tag can be used to set the relative size, family, and color of various text elements in a page. For example, consider the following: Special Value Sales. This will display the words "Special Value Sales" in a bold Arial font. The words will be colored red, and the size of the text will be slightly larger than normal (default is SIZE=3). Imagine that you had to insert tags throughout a page to control the look of various elements. For instance, suppose you wanted all paragraphs to have a particular face and size. Not only would this be tedious, but it also would be error prone, especially given the number of pages and the number of people working on those pages.

Cascading style sheets (CSSs) are a well-supported and easy-to-use standard designed to address problems of this sort. CSSs are text files or special text in a HTML file that specify styles for various elements in an HTML document. A style is a set of formatting characteristics that can be applied to any element so that its appearance can be controlled and easily changed. When a style is applied to an element, a whole group of formats are applied at once. For example, styles can be used to control:

- ▶ The alignment, character formatting, line spacing, and borders of a paragraph
- ▶ The font, size, and formatting (e.g., bold and italics) of selected text
- ▶ The border, shading, alignment, and fonts in a table
- ▶ The alignment, numbered or bulleted characters, and fonts in a list

CSSs serve to:

- ▶ Simplify the creation and maintenance of pages by eliminating the need to add detailed tags and attributes throughout a page and by making it easier to modify the look of particular elements by simply changing styles

cascading style sheet (CSS)

A standard that uses text files to specify formatting characteristics (styles) for various elements in an HTML document, allowing styles to be controlled and easily changed.

- Enhance the richness of the design and layout to Web pages by offering enhanced positioning and control over the elements of an HTML document
- Ensure consistency across the pages in a site, especially when there are multiple people involved in the creation and maintenance of the pages

As an example of the way that styles might be used, consider Exhibit B.15, which defines a style for the body and “number 1 headers” for the home page of the Premium Selections site. Styles are usually defined in the “head” of a page and are denoted by the `<STYLE> . . . </STYLE>` tags. In this case, two styles are defined. Each style can be thought of as an instruction or rule that has two parts: a selector (or HTML element) and a set of properties. In this instance, the first style states that the “BACKGROUND” color of the “BODY” will be white. The second says that all H1 headlines will use a 24-point font (FONT-SIZE: 24) and will be italicized (FONT-WEIGHT: ITALIC). The page that results from these styles is essentially the same as the one shown in Exhibit B.1.

In practice, styles are usually placed in a separate file that is linked to a page rather than physically embedding the styles within a page. In this way, a set of styles can be applied to more than one page or document, much like the style templates shared among Microsoft Word documents. For instance, in our earlier example, we might have removed the styles specified between the `<STYLE> </STYLE>` tags and placed them in a separate file called “mystyles.css.” The name of the file could be anything as long as it has a .css extension. In place of the `<STYLE> </STYLE>` tags, we would insert the following tag:

```
<LINK REL=stylesheet TYPE="text/css" HREF="mystyles.css">
```

We could do this for any number of pages, ensuring that the backgrounds and H1 headers would be treated the same across the pages. If we decided, for example, to change the color of the background or the size of the H1 fonts, then all we would have to do is change the settings in the “mystyles.css” file.

EXHIBIT B.15 HTML Tags for Embedded Styles

```
<HTML>

<HEAD>

<TITLE>Premium Selections</TITLE>

<STYLE>

  BODY {BACKGROUND:white}
  H1 {FONT-SIZE:24 pt; FONT-WEIGHT: italic}
</STYLE>

</HEAD>

<BODY>

  <P>
  <IMG SRC="premium.gif">
  </P>

  <H1>Premium Selections</H1>
  Specializing in fine domestic wines and cheeses.

  <P>
  <A HREF="wines.htm">Wine Shop</A>
  </P>

  <P>
  <A HREF="cheeses.htm">Cheese Shop</A>
  </P>

  <P>
  <A HREF="shoppingcart.exe">Review Shopping Cart</A>
  </P>

</BODY>
</HTML>
```

Like the HTML standards, the standards for CSS are fairly involved and are controlled by the W3C (see w3.org/style/css). Meyer (2002) offers both a beginner's introduction to and an advanced discussion of CSS. A number of tutorials are also available on the Web, including the introduction found at westciv.com.au/style_master/academy/css_tutorial/index.html.

EXTENSIBLE MARKUP LANGUAGE

Style sheets simplify the task of creating and maintaining a consistent look and feel across a Web site. This still leaves the issue of content management. HTML documents do not have structured content. The content is simply interspersed among the tags in the page. This makes it difficult to create and maintain the content. It also makes it difficult for other applications, such as software agents, to use the content. To understand these difficulties, look at the code displayed in Exhibit B.16. (Remember that Web designers and software robots work with code, not the actual display.) Now, answer a couple of questions. First, "What do these figures represent?" You probably guessed stock prices. Second, "What was the closing price for Microsoft stock (MSFT) on this particular date?" You can certainly determine this by looking at the tags and content, but it requires effort. It would be even harder to write a program to extract the prices associated with the stocks contained in the table. The problem is that HTML mixes the content with the display format and provides few clues to the meaning of the content.

Look at the code in Exhibit B.17 (next page) and try to answer these same questions. It is a pretty easy task to find the answers from this code. It would also be a relatively easy task to write a program that understood the data. The code in Exhibit B.17 is written in XML, which is short for **extensible markup**

extensible markup language (XML)

An open standard for defining data elements. Like HTML, it uses a set of tags along with content (or values), but unlike HTML, the tags are defined by the Web designer (there are no fixed tags).

EXHIBIT B.16 Mystery Table

```
<HTML>
<HEAD>
<TITLE>Sample Table</TITLE>
</HEAD>
<BODY>

<H1>Prices</H1>

<TABLE>
<TR>
<TH>Ticker</TH>
<TH>Hi</TH>
<TH>Low</TH>
<TH>Close</TH>
<TH>Volume</TH>
</TR>
<TR>
<TD>MSFT</TD>
<TD>50</TD>
<TD>48</TD>
<TD>49</TD>
<TD>92500000</TD>
</TR>
<TR>
<TD>XYZ</TD>
<TD>80</TD>
<TD>78</TD>
<TD>78</TD>
<TD>100000</TD>
</TR>
</TABLE>

</BODY>
</HTML>
```


EXHIBIT B.17 XML Document of Information from Mystery Table

```
<?xml version="1.0" ?>
<STOCKS>
  <STOCK>
    <TICKER>MSFT</TICKER>
    <HI>51</HI>
    <LOW>48</LOW>
    <CLOSE>50</CLOSE>
    <VOLUME>92500000</VOLUME>
  </STOCK>
  <STOCK>
    <TICKER>XYZ</TICKER>
    <HI>80</HI>
    <LOW>78</LOW>
    <CLOSE>78</CLOSE>
    <VOLUME>100000</VOLUME>
  </STOCK>
</STOCKS>
```

document type definition (DTD)

In XML, a file that defines the tags that are allowed and the manner in which they can be used; basically, a set of grammar rules for the tags in a particular document.

language. The XML standards are controlled by the W3C. Like HTML, an XML document is a text file with a set of tags and content (or values). Unlike HTML, the tags are not fixed. This means that the Web designer or author is free to create their own tags; this is why the language is called extensible. For any given page, the tags that are allowed and the manner in which they can be used are defined in a file called the **document type definition (DTD)**. A DTD is basically a set of grammar rules that dictate the makeup of the tags in a particular document. DTDs are optional and are used primarily for determining the validity of an XML document. (If you are interested in acquiring a deeper understanding of DTDs, see Ray 2001.)

The tags in an XML document define the structure of the content. The tags in Exhibit B.17 indicate that this is a STOCK REPORT. Each STOCK has a TICKER, HI, LOW, CLOSE, and VOLUME. The structure is arbitrary; no standard defines specifically how stocks are to be reported in an XML document. However, several domains are working to develop a set of standard tags. At last count, the W3C listed about 200 such efforts, including the following:

- ▶ **Open Financial Exchange (OFX).** An XML specification proposed by Microsoft, Intuit, and Checkfree for exchanging financial data among financial institutions, businesses, and consumers via the Internet (ofx.net).
- ▶ **Open Trading Protocol (OTP).** An XML standard developed by 30 member companies for handling remote electronic purchases regardless of the payment mechanism—SET, digital cash, e-checks, and debit cards (ietf.org/html.charters/trade-charter.html).
- ▶ **Extensible Business Reporting Language (XBRL).** An XML specification developed by an international consortium of over 85 businesses for the preparation and exchange of business reports, including regulatory filings such as annual and quarterly financial statements, general ledger information, and audit schedules (xbrl.org).
- ▶ **Electronic Business Extensible Markup Language (ebXML).** A modular suite of XML specifications sponsored by UN/CEFACT and OASIS, enabling enterprises of any size and in any location to conduct business over the Internet (ebxml.org).

An extensive description of these and other efforts can be found in Oasis (2003).

Although XML has the same roots as HTML and is basically a child of the Web, an XML document says nothing about how the structure and content are to be displayed. In fact, XML is device independent. It can be displayed on any device, including browsers, printers, PDAs, and cell phones.

The contents of an XML document can be displayed in a Web browser in a number of ways. As originally envisioned, a special type of style sheet called an **extensible style language (XSL)** was to be used in defining how the contents of an XML document were to be displayed. That vision has been sidetracked. Currently, version 1.0 of the XSL standard is still under review by the W3C. Because it has taken so long to agree on the standard, the W3C has divided it into two components: **XSLT** (for **XSL Transformations**) and **XSL-FO** (for **XSL Formatting Objects**). XSLT has been formalized, whereas

extensible style language (XSL)

A special type of style sheet that was to be used in defining how the contents of an XML document were to be displayed; this standard is still under review.

XSLT (XSL Transformations)

A component of XSL that has been formalized.

XSL-FO (XSL Formatting Objects)

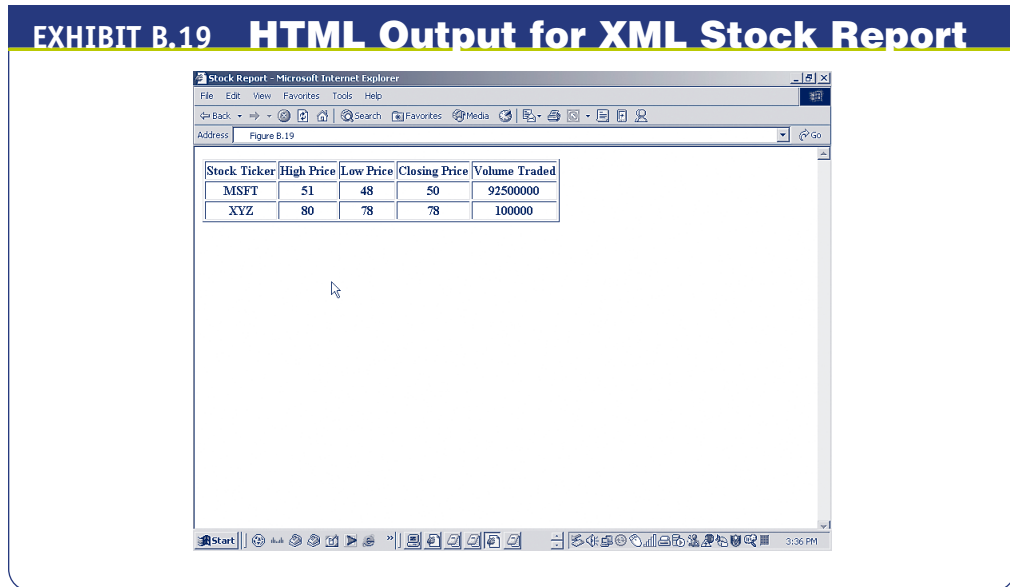
A component of XSL that has not yet been formalized.

XSL-FO has not. This means that in order to use an XSL file to display an XML document in something like a browser, one has to specify a series of XSL Transformations that detail how the various XML structures will appear in the browser. Because of the technical nature of XSLT, no attempt will be made to illustrate this method. Instead, you are referred to Castro (2002) or Mangano (2002).

In the case of Microsoft's IE browser (version 5.0 and above), there is a simpler method. With IE 5.0, an HTML page can be used to display the XML structure and values. This method is illustrated in Exhibit B.18. As the figure shows, the XML structures and values are placed between <XML> </XML> tags. The content between the <XML> </XML> tags is also given an ID (i.e., "xmlData"). The ID is then referenced as a data source (DATASRC) by the HTML TABLE used to display the XML contents. tags are employed to associate particular (DATAFLD) elements in the XML structure (e.g., TICKER) with particular cells in the table. The resulting HTML page is shown in Exhibit B.19 (next page).

EXHIBIT B.18 Sample HTML Page for Displaying XML Stock Report

```
<HTML>
<HEAD>
<TITLE>Stock Report</TITLE>
<XML ID="xmlDoc">
<STOCKS>
<STOCK>
<TICKER>MSFT</TICKER>
<HI>51</HI>
<LOW>48</LOW>
<CLOSE>50</CLOSE>
<VOLUME>92500000</VOLUME>
</STOCK>
<STOCK>
<TICKER>XYZ</TICKER>
<HI>80</HI>
<LOW>78</LOW>
<CLOSE>78</CLOSE>
<VOLUME>100000</VOLUME>
</STOCK>
</STOCKS>
</XML>
<BODY BGCOLOR="#FFFFFF">
<TABLE BORDER="1" DATASRC="#xmlDoc">
<THEAD>
<TR>
<TH>Stock Ticker</TH>
<TH>High Price</TH>
<TH>Low Price</TH>
<TH>Closing Price</TH>
<TH>Volume Traded</TH>
</TR>
</THEAD>
<TR>
<TH><SPAN DATAFLD="TICKER"></SPAN></TH>
<TH><SPAN DATAFLD="HI"></TH>
<TH><SPAN DATAFLD="LOW"></TH>
<TH><SPAN DATAFLD="CLOSE"></TH>
<TH><SPAN DATAFLD="VOLUME"></TH>
</TR>
</TABLE>
</BODY>
</HTML>
```

EXHIBIT B.19 HTML Output for XML Stock Report**XHTML**

An intermediary language developed by the W3C to ease the transition from HTML to XML by applying the rigors of XML's syntax rules to HTML's tags.

The W3C has developed an intermediary language—**XHTML**—to ease the transition from HTML to XML (Castro 2002). Essentially, XHTML applies the rigors of XML's syntax rules to HTML's tags (w3.org/Markup). For example, in XML all the tags appear as pairs. In HTML, some tags do not require pairs (e.g., <P>), but in XHTML they do (e.g., <P> requires a closing </P>). The uptake of XHTML has been slow, in part because the browsers still support HTML and are likely to do so for the next few years. Until that time, it makes more sense to use XML to manage content and to convert it to HTML for display either with XSLT rules or <XML> </XML> tags rather than developing pages in XHTML.

KEY TERMS

Cascading style sheet (CSS)	13	Hypertext markup language (HTML)	1	XSLT (XSL Transformations)	16
Document type definition (DTD)	16	XSL-FO (XSL Formatting Objects)	16	XHTML	18
Extensible markup language (XML)	15				
Extensible style language (XSL)	16				

REFERENCES

- Castro, E. *HTML for the World Wide Web with XHTML and CSS: Visual QuickStart Guide*, 5th ed. Berkeley, CA: Peachpit Press, 2002.
- Mangano, S. *XSLT Cookbook*. Sebastopol, CA: O'Reilly & Associates, 2002.
- Meyer, E. *Eric Meyer on CSS: Mastering the Language of Web Design*. Indianapolis, IN: New Riders Publishing, 2002.
- Oasis. "The Cover Pages." xml.coverpages.org/AboutCoverPages.html, 2003 (accessed July 2003).
- Ray, E. *Learning XML*. Sebastopol, CA: O'Reilly & Associates, 2001.
- Willard, W. *HTML: A Beginner's Guide*. New York: McGraw-Hill, 2001.

ADDITIONAL READINGS

- Zeldman, J. *Designing with Web Standards*. Indianapolis, IN: New Riders Publishing, 2003.