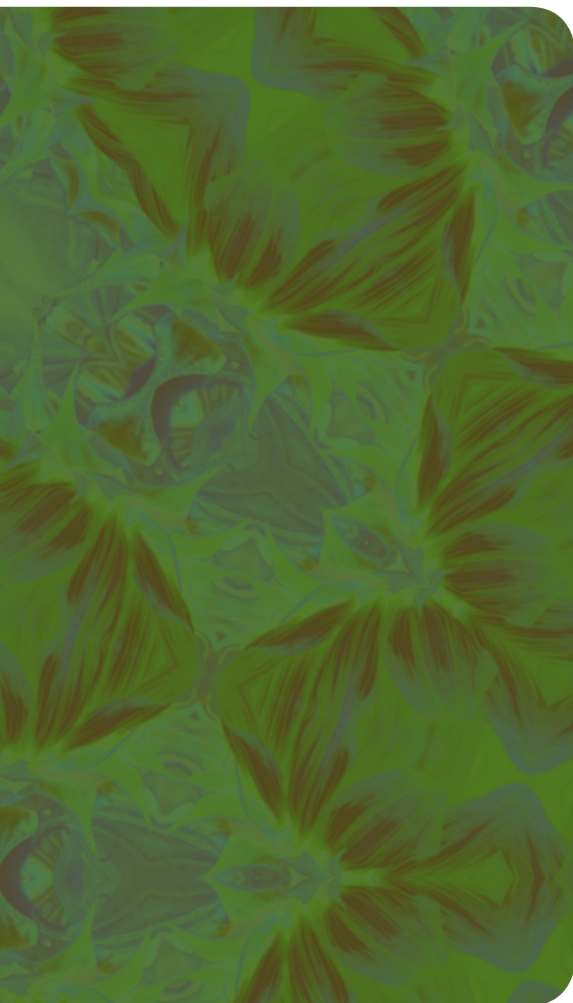


HTML, XHTML, and XML

3rd Edition

Patrick Carey



COMPREHENSIVE

**New Perspectives on HTML, XHTML, and XML,
3rd Edition—Comprehensive**

Vice President, Publisher: Nicole Jones Pinard
Executive Editor: Marie L. Lee
Associate Acquisitions Editor: Brandi Shailer
Senior Product Manager: Kathy Finnegan
Associate Product Manager: Leigh Robbins
Editorial Assistant: Julia Leroux-Lindsey
Director of Marketing: Cheryl Costantini
Marketing Manager: Ryan DeGrote
Marketing Coordinator: Kristen Panciocco
Developmental Editors: Mary Kemper, Robin M. Romer
Senior Content Project Manager: Jennifer Goguen McGrail
Composition: GEX Publishing Services
Text Designer: Steve Deschene
Art Director: Marissa Falco
Cover Designer: Elizabeth Paquin
Cover Art: Bill Brown
Copyeditor: Suzanne Huizenga
Proofreader: Kathy Orrino
Indexer: Alexandra Nickerson

© 2010 Course Technology, Cengage Learning

ALL RIGHTS RESERVED. No part of this work covered by the copyright herein may be reproduced, transmitted, stored or used in any form or by any means graphic, electronic, or mechanical, including but not limited to photocopying, recording, scanning, digitizing, taping, Web distribution, information networks, or information storage and retrieval systems, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without the prior written permission of the publisher.

For product information and technology assistance, contact us at
Cengage Learning Customer & Sales Support, 1-800-354-9706
For permission to use material from this text or product, submit all
requests online at cengage.com/permissions
Further permissions questions can be emailed to
permissionrequest@cengage.com

Some of the product names and company names used in this book have been used for identification purposes only and may be trademarks or registered trademarks of their respective manufacturers and sellers.

Microsoft and the Office logo are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Course Technology, Cengage Learning is an independent entity from the Microsoft Corporation, and not affiliated with Microsoft in any manner.

Disclaimer: Any fictional data related to persons or companies or URLs used throughout this book is intended for instructional purposes only. At the time this book was printed, any such data was fictional and not belonging to any real persons or companies.

ISBN-13: 978-0-495-80640-0

ISBN-10: 0-495-80640-4

Course Technology
20 Channel Center Street
Boston, Massachusetts 02210
USA

Cengage Learning is a leading provider of customized learning solutions with office locations around the globe, including Singapore, the United Kingdom, Australia, Mexico, Brazil, and Japan. Locate your local office at:
international.cengage.com/region

Cengage Learning products are represented in Canada by Nelson Education, Ltd.

To learn more about Course Technology, visit www.cengage.com/coursetechnology
To learn more about Cengage Learning, visit www.cengage.com

Purchase any of our products at your local college store or at our preferred online store www.ichapters.com

Objectives

Session 1.1

- Learn the history of the Web and HTML
- Describe HTML standards and specifications
- Understand HTML elements and markup tags
- Create the basic structure of an HTML file
- Insert an HTML comment

Session 1.2

- Work with block-level elements
- Create ordered, unordered, and definition lists
- Work with inline elements
- Understand the div and span elements

Session 1.3

- Add attributes to HTML elements
- Format page content using the style attribute
- Mark empty elements with one-sided tags
- Add an inline image to a Web page
- Work with character sets and codes

Developing a Web Page

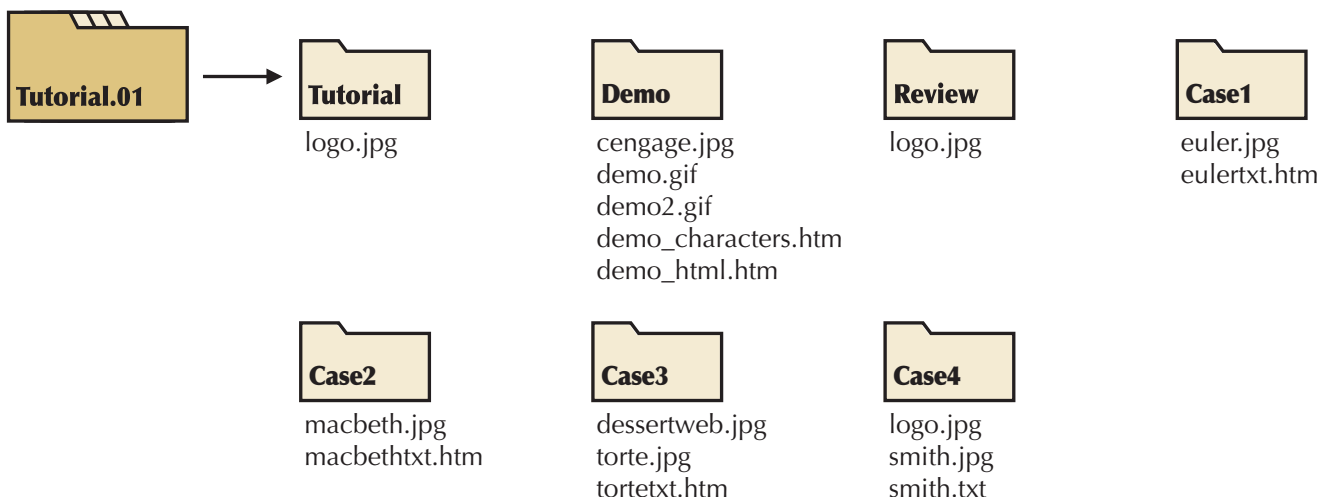
Creating a Product Page for a Startup Company

Case | Dave's Devil Sticks

Dave Vinet is a machinist in Auburn, Maine. In his spare time, Dave builds and juggles devil sticks—juggling props used in circuses and by street performers. In recent years, he has made customized sticks for his friends and colleagues. Encouraged by their enthusiasm for his work, Dave has decided to start a business called Dave's Devil Sticks. So far his customers have come through word of mouth; now Dave wants to advertise his business on the Web. To do that, Dave needs to create a Web page that describes his company and its products. He has the text describing his company in a flyer that he hands out at juggling conventions. He has also contacted a graphic artist to design a logo. He wants to use this material in his Web page.

He has come to you for help in designing a Web page and writing the code. He wants the Web page to contain the same information and graphics contained in his flyer. To create Dave's Web page, you'll have to learn how to work with HTML, the markup language used to create documents on the World Wide Web.

Starting Data Files



Session 1.1

Exploring the History of the World Wide Web

Before you start creating a Web page for Dave, it's helpful to first look at the history of the Web and how HTML was developed. You'll start by reviewing networks.

Networks

A **network** is a structure that links several points called **nodes** allowing for the sharing of information and services. For computer networks, each node is a device such as a computer or a printer or a scanner, capable of sending and receiving data electronically over the network. A computer node is also called a **host** to distinguish it from other node devices.

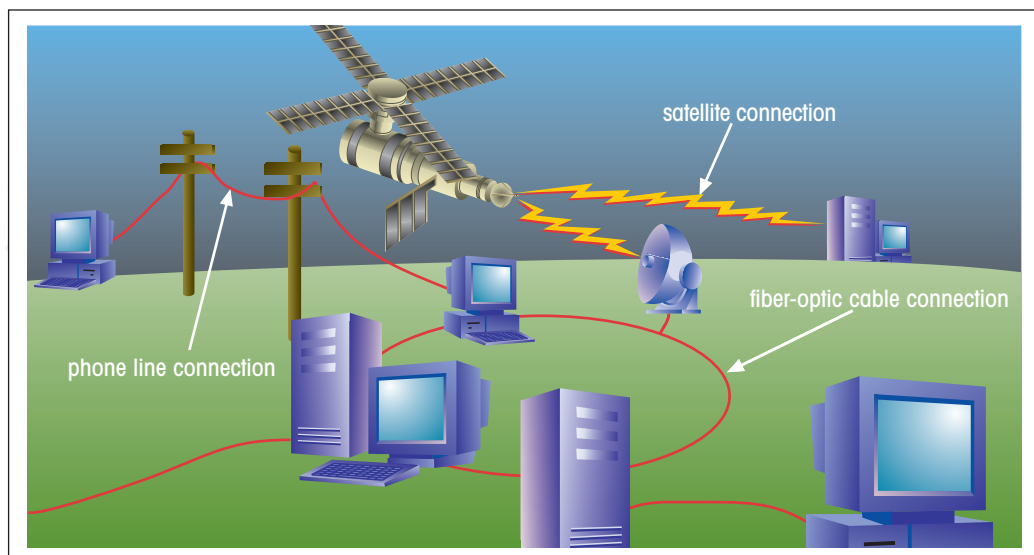
As the network operates, nodes are either providing data to other nodes on the network or requesting data. A node that provides information or a service is called a **server**. For example, a **print server** is a network node that provides printing services to the network; a **file server** is a node that provides storage space for saving and retrieving files. A computer or other device that requests services from a server is called a **client**. Networks can follow several different designs. One of the most commonly used designs is the **client-server network** in which several clients access information provided by one or more servers. You might be using such a network to access your data files for this tutorial.

Networks can also be classified based on the range they cover. A network confined to a small geographic area, such as within a building or department, is referred to as a **local area network** or **LAN**. A network that covers a wider area, such as several buildings or cities, is called a **wide area network** or **WAN**. Wide area networks typically consist of two or more local area networks connected together.

The largest WAN is the Internet. The origins of the Internet can be traced back to a WAN called the **ARPANET**, which started with two network nodes located at UCLA and Stanford connected by a single phone line. Today, the **Internet** has grown to an uncountable number of nodes involving computers, cell phones, PDAs, MP3 players, gaming systems, and television stations. The physical structure of the Internet uses fiber-optic cables, satellites, phone lines, wireless access points, and other telecommunications media, enabling a world-wide community to communicate and share information. See Figure 1-1. It is within this expansive network that Dave wants to advertise his devil sticks business.

Figure 1-1

Structure of the Internet



Locating Information on a Network

One of the biggest obstacles to effectively using a network is not mechanical—it's the human element. Users must be able to easily navigate the network and locate the information and services they need. Most of the early Internet tools required users to master a bewildering array of terms, acronyms, and commands. Because network users had to be well versed in computers and network technology, Internet use was limited to universities and departments of the government. To make the Internet accessible to the general public, it needed a simpler interface. This interface proved to be the World Wide Web.

The foundations for the **World Wide Web**, or the **Web** for short, were laid in 1989 by Timothy Berners-Lee and other researchers at the CERN nuclear research facility near Geneva, Switzerland. They needed an information system that would make it easy for their researchers to locate and share data with minimal training and support. To meet this need, they developed a system of hypertext documents that enabled users to easily navigate from one topic to another. **Hypertext** is a method of organization in which information is not presented linearly, but in whatever order is requested by the user. For example, if you read the operating manual for your car starting with page 1 and proceeding to the end, you are processing the information linearly and in the order determined by the manual's author. A hypertext approach would place the same information in a series of smaller documents, with each document dedicated to a single topic, allowing you—and not the author—to choose the order and selection of topics you'll view.

The key to hypertext is the use of **links**, which are the elements in a hypertext document that allow you to jump from one topic or document to another, usually by clicking a mouse button. Hypertext is ideally suited to use with networks because the end user does not need to know where a particular document, information source, or service is located—he or she only needs to know how to activate the link. In the case of an expansive network like the Internet, documents can be located anywhere in the world; but that is largely unseen by the user because of the hypertext structure. The fact that the Internet and the World Wide Web are synonymous in many users' minds is a testament to the success of the hypertext approach.

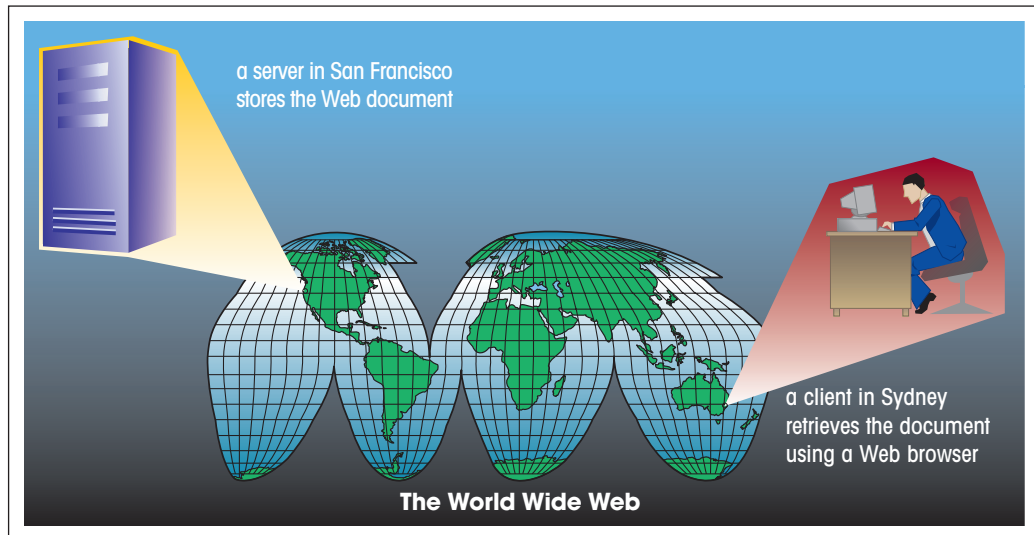
The original Web supported only textual documents, but the use of hypertext links has expanded through the years to encompass information in any form, including video, sound, interactive programs, conferencing, and online gaming. While the Web has greatly expanded to include these services, the basic foundation is still the same: a collection of interconnected documents linked through the use of hypertext.

Web Pages and Web Servers

Each document on the World Wide Web is referred to as a **Web page**. Web pages are stored on **Web servers**, which are computers that make Web pages available to any device connected to the Internet. To view a Web page, the end user's device needs a software program called a **Web browser**, which retrieves the page from the Web server and renders it on the user's computer or other device. See Figure 1-2.

Figure 1-2

Using a browser to view a Web document from a Web server



The earliest browsers, known as **text-based browsers**, were limited to displaying only text. Today's browsers are capable of displaying text, images, video, sound, and animations. In the early days of the Internet, Web browsing was limited to computers. Now browsers are installed on cell phones, PDAs (personal digital assistants), MP3 players, and gaming systems. How does a Web page work with so many combinations of browsers and clients and devices? To understand, you need to look at how Web pages are created.

Introducing HTML

A Web page is actually a text file written in **Hypertext Markup Language** or **HTML**. We've already discussed hypertext, but what is a markup language? A **markup language** is a language that describes the content and structure of a document. If this tutorial were written using a markup language, the language would identify the parts of the document, indicating which sections correspond to paragraphs, figure captions, tables, page headings, and so forth.

There are several things that HTML is not. While Web pages often contain interactive programs, HTML is not a programming language. In addition, while HTML can describe the content of a document, it is not a formatting language because it does not necessarily describe how content should be rendered. This is a necessary facet of HTML: the Web page author has no control over what device is used to view the Web page, so the browser—not the HTML—determines how the Web page will look. The end user might be using a large-screen television monitor, a cell phone, or even a device that renders Web pages in Braille or in aural speech.

If you want to format your document, the preferred method is to use styles. **Styles** are formatting rules written in a separate language from HTML telling the browser how to render each element for particular devices. A Web page author can write a style that displays page headings one way for computer monitors and another way for printed output. You'll explore some basic styles as you create your first Web pages.

The History of HTML

HTML evolved as the Web itself evolved. Thus in order to fully appreciate the nuances of HTML, it's a good idea to review the language's history. The first popular markup language was the **Standard Generalized Markup Language (SGML)**. Introduced in the 1980s, SGML is device- and system-independent, meaning that it can be applied to almost any type of document stored in almost any format. While powerful, SGML is also quite complex; and for this reason SGML is limited to those organizations that can afford the cost and overhead of maintaining complex SGML environments. However, SGML can also be used to create other markup languages that are tailored to specific tasks and are simpler to use and maintain. HTML is one of the languages created with SGML.

In the early years after HTML was created, no single organization was responsible for the language. Web developers were free to define and modify HTML in whatever ways they thought best. Eventually, competing browsers, seeking to dominate the market, added new features called **extensions** to the language. The two major browsers during the 1990s, Netscape Navigator and Microsoft Internet Explorer, added the most extensions to HTML. Netscape provided an extension to add background sounds to documents, while Internet Explorer added an extension to provide marquee-style text that would scroll automatically across the page. These extensions and others provided Web page authors with more options, but at the expense of complicating Web page development. A Web page that took advantage of extensions might work in one browser but not in another.

Thus Web page authors faced the challenge of determining which browser or browser version supported a particular extension, and they had to create a workaround for browsers that did not. By adding this layer of complexity to Web design, extensions, while often useful, diminished the promise of simplicity that made HTML so attractive in the first place.

Ultimately, a group of Web developers, programmers, and authors called the **World Wide Web Consortium**, or the **W3C**, created a set of standards or specifications that all browser manufacturers were to follow. The W3C has no enforcement power; but because a uniform language is in everyone's best interest, the W3C's recommendations are usually followed, though not always right away. The W3C also provides online tutorials, documentation, and quizzes that you can use to test your knowledge of HTML and other languages. For more information on the W3C and the services it offers, see its Web site at www.w3c.org.

Figure 1-3 summarizes the various versions of HTML that the W3C has released over the past decade. While you may not grasp all of the details of these versions yet, it's important to understand that HTML doesn't come in only one version.

CENGAGE **brain**.com

Figure 1-3 History of HTML and XHTML

Version	Date of Release	Description
HTML 1.0	1989	The first public version of HTML which included browser support for inline images and text controls.
HTML 2.0	1995	The first version supported by all graphical browsers. It introduced interactive form elements such as option buttons and text boxes. A document written to the HTML 2.0 specification is compatible with almost all browsers on the World Wide Web.
HTML 3.0	1996	A proposed replacement for HTML 2.0 that was never widely adopted.
HTML 3.2	1997	This version included additional support for creating and formatting tables and expanded the options for interactive form elements. It also supported limited programming using scripts.
HTML 4.01	1999	This version added support for style sheets to give Web designers greater control over page layout. It added new features to tables and forms and provided support for international features. This version also expanded HTML's scripting capability and added increased support for multimedia elements.
HTML 5.0	not yet released	This version supports elements that reflect current Web usage, including elements for Web site navigation and indexing for use with search engines. This version also removes support for purely presentational elements because those effects can be better handled with styles.
XHTML 1.0	2001	This version is a reformulation of HTML 4.01 in XML and combines the strength of HTML 4.0 with the power of XML. XHTML brings the rigor of XML to Web pages and provides standards for more robust Web content on a wide range of browser platforms.
XHTML 1.1	2002	A minor update to XHTML 1.0 that allows for modularity and simplifies writing extensions to the language.
XHTML 2.0	not yet released	The latest version, designed to remove most of the presentational features left in HTML. XHTML 2.0 is not backward compatible with XHTML 1.1.
XHTML 5.0	not yet released	A version of HTML 5.0 written under the specifications of XML, unlike XHTML 2.0, XHTML 5.0 will be backward-compatible with XHTML 1.1.

Tip

You can learn more about deprecated features by examining the documentation available at the W3C Web site and by viewing the source code of various pages on the Web.

When you work with HTML, you should keep in mind not only what the W3C has recommended, but also what HTML features the browser market actually supports. This might mean dealing with a collection of approaches: some browsers are new and meet the latest W3C specifications, while some are older but still widely supported. Older features of HTML are often **deprecated**, or phased out, by the W3C. While deprecated features might not be supported in current or future browsers, that doesn't mean that you won't encounter them—indeed, if you are supporting older browsers that recognize only early versions of HTML, you might need to use them. Because it's hard to predict how quickly deprecated features will disappear from common usage, it's crucial to be familiar with them.

Current Web developers are increasingly using **XML (Extensible Markup Language)**, a language for creating markup languages, like SGML, but without SGML's complexity and overhead. Using XML, developers can create documents that obey specific rules for their content and structure. This is in contrast with a language like HTML, which supported a wide variety of rules but did not include a mechanism for enforcing those rules.

Indeed, one of the markup languages created with XML is **XHTML (Extensible Hypertext Markup Language)**, a stricter version of HTML. XHTML is designed to confront some of the problems associated with the various competing versions of HTML and to better integrate HTML with other markup languages like XML. The current version of XHTML is XHTML 1.1, which is mostly (but still not completely) supported by all

browsers. Because XHTML is an XML version of HTML, most of what you learn about HTML can be applied to XHTML.

Another version of XHTML, **XHTML 2.0**, is still in the draft stage and has proved to be controversial because it is not backward-compatible with earlier versions of HTML and XHTML. In response to this controversy, another working draft of HTML called **HTML 5.0** is being developed. It provides greater support for emerging online technology while still providing support for older browsers. HTML 5 is also being developed under the XML specifications as **XHTML 5.0**. At the time of this writing, none of these versions has moved beyond the development stage nor has been adopted by the major browsers. This book discusses the syntax of HTML 4.01 and XHTML 1.1, but also brings in deprecated features and browser-supported extensions where appropriate.

Writing HTML Code

InSight

Part of writing good HTML code is being aware of the requirements of various browsers and devices as well as understanding the different versions of the language. Here are a few guidelines for writing good HTML code:

- Become well versed in the history of HTML and the various versions of HTML and XHTML. Unlike other languages, HTML's history does impact how you write your code.
- Know your market. Do you have to support older browsers, or have your clients standardized on one particular browser or browser version? Will your Web pages be viewed on a single device like a computer, or do you have to support a variety of devices?
- Test your code on several different browsers and browser versions. Don't assume that if your page works in one browser it will work on other browsers or even on earlier versions of the same browser. Also check on the speed of the connection. A large file that performs well under a high-speed connection might be unusable under a dial-up connection.
- Read the documentation on the different versions of HTML and XHTML at the W3C Web site and review the latest developments in new versions of the languages.

In general, any HTML code that you write should be compatible with the current versions of the following browsers: Internet Explorer (Windows), Firefox (Windows and Macintosh), Netscape Navigator (Windows), Opera (Windows), and Safari (Macintosh).

Tools for Creating HTML Documents

Because HTML documents are simple text files, you can create them with nothing more than a basic text editor such as Windows Notepad. Specialized HTML authoring programs, known as HTML converters and HTML editors, are also available to perform some of the rote work of document creation. An **HTML converter** is a program that translates text written in another language into HTML code. You can create the source document with a word processor such as Microsoft Word, and then use the converter to save the document as an HTML file. Converters free you from the laborious task of typing HTML code; and because the conversion is automated, you usually do not have to worry about introducing coding errors into your document. However, converters tend to create large and complicated HTML files resulting in "bloated" code, which is more difficult to edit if you need to make changes. So while a converter can speed up Web page development, you will probably still have to invest time in cleaning up the code.

An **HTML editor** is a program that helps you create an HTML file by inserting HTML codes for you as you work. HTML editors can save you a lot of time and can help you work more efficiently. Their advantages and limitations are similar to those of HTML converters. Like converters, HTML editors allow you to set up a Web page quickly, but you will still have to work directly with the underlying HTML code to create a finished product.

Creating an HTML Document

Now that you've had a chance to explore some of the history of the Web and HTML's role in its development, you are ready to work on the Web page for Dave's Devil Sticks. It's always a good idea to plan your Web page before you start coding it. You can do this by drawing a sketch or by creating a sample document using a word processor. The preparatory work can weed out errors or point to potential problems. In this case, Dave has already drawn up a flyer he's passed out at juggling and circus conventions. The handout provides information about Dave's company and his products. Figure 1-4 shows Dave's current flyer.

Figure 1-4 Elements of the Dave's Devil Sticks flyer

The flyer for "Dave's Devil Sticks" contains the following elements:

- image:** The logo "Dave's Devil Sticks" in a stylized font, accompanied by a photograph of a man (Dave) holding a stick.
- heading:** "Who Am I?"
- boldface text:** "Dave's Devil Sticks" in the logo.
- paragraph:** "Welcome to **Dave's Devil Sticks**. If you are looking for juggling balls, hacky sacks, pins, unicycles, or magic hats, you've come to the wrong place; but if you're looking for high-quality, hand-crafted devil and flower sticks, this is the site for you. I've been designing and building sticks for the past 20 years, and I know that my sticks are the best of their kind. Don't take my word for it; read the following testimonial:"
- horizontal line:** A line separating the introductory paragraph from the testimonial.
- block quote:** "I'm more than happy to recommend Dave Vinet's products. I came upon his work 10 years ago and was immediately impressed by his craftsmanship. I've been using his sticks in my shows ever since. They're durable, well-balanced, and attractive props and are the perfect complement to my performances. Thanks Dave!"
- horizontal line:** A line separating the testimonial from the main body text.
- Text:** "Every set is checked and tested before being shipped out to assure perfect quality. I take pride in every one of my sticks and I want my customers to feel that same pride."
- heading:** "My Products"
- bulleted list:**
 - **Basic Stick**
The easiest stick to learn with, but "grippy" enough for the most demanding tricks. Comes in red, green, and blue.
 - **Flower Stick**
A graceful stick with colored tassels. Flower Sticks float more slowly, giving you the ideal chance to practice new moves.
 - **Master Stick**
My most popular stick is shorter and heavier for fast play and more advanced tricks. Each Master set is available in custom colors.
 - **Glow Stick**
The Glow Stick shines brightly at night (without the danger of a fire stick). It combines the fun and versatility of the Master Stick, adding the fun tricks unique to a glow-in-the-dark stick.
- horizontal line:** A line separating the product list from the address.
- address:** "Dave's Devil Sticks • 541 West Highland Dr. • Auburn, ME 04210 • (207) 555 - 9001 •"

When you sketch a sample document, it is a good idea to identify the document's various elements. An **element** is a distinct object in the document, like a paragraph, a heading, or the page's title. Even the whole document is considered an element. Dave's flyer includes several elements: an image displays his company's logo, several headings break his flyer into sections, the text in his flyer is laid out in paragraphs, a bulleted list describes his products, and the address of his company is at the bottom of the flyer. Note that some elements are marked by their appearance in the text. For example, the name of his company is displayed in boldface text at the top of the flyer to set it off from other text in the opening paragraph. Italics are also used in several locations on the page. As you recreate this flyer as a Web page, you should periodically refer to Figure 1-4.

Marking Elements with Tags

The core building block of HTML is the **tag**, which marks the presence of an element. If the element contains content such as text or another element, it is marked using a **two-sided tag** in which an **opening tag** indicates the beginning of the content and a **closing tag** indicates the content's end. The syntax of a two-sided tag is:

```
<element>content</element>
```

where *element* is the name of the element and *content* is any content contained within the element. For example, the following code is used to mark a paragraph element within a document:

```
<p>Welcome to Dave's Devil Sticks.</p>
```

In this example, the `<p>` tag marks the beginning of the paragraph, the text "Welcome to Dave's Devil Sticks." constitutes the content of the paragraph element and the `</p>` tag marks the end of the paragraph.

Note that an "element" is an object in the Web document, and a "tag" is the part of the HTML code that marks the element. So you would mark a paragraph element in a document by enclosing the paragraph content within opening and closing paragraph tags.

Elements can also contain other elements. For example, the paragraph tags in the following code

```
<p>Welcome to <b>Dave's Devil Sticks</b>.</p>
```

enclose both the text of the paragraph as well as another set of tags ` ... ` that are used to mark content that should be treated by the browser as boldface text. Note that the `` tags have to be completely enclosed or nested within the `<p>` tags. It's improper syntax to have tags overlap as in the following code sample:

```
<p>Welcome to <b>Dave's Devil Sticks.</p></b>
```

In this example, the closing `` tag is placed *after* the closing `</p>` tag, which is improper because the boldface text marked with these tags must be completely enclosed *within* the paragraph.

The Structure of an HTML Document

All documents written in a markup language need to have a **root element** that contains all of the elements used in the document. For HTML documents, the root element is marked using the `<html>` tag as follows

```
<html>
  document content
</html>
```

where *document content* is the content of the entire document, including all other elements. The presence of the opening `<html>` tag in the first line of the file tells any device reading the document that this file is written in HTML. The closing `</html>` tag signals the end of the document and should not be followed by any other content or markup tags.

Web pages are divided into two main sections: a head and a body. The **head element** contains information about the document—for example, the document's title or a list of keywords that would aid a search engine on the Web identifying this document for other users. The **body element** contains all of the content that will appear on the Web page. Taken together, the syntax of the entire HTML file including the head and body elements is

```
<html>
  <head>
    head content
  </head>
  <body>
    body content
  </body>
</html>
```

where *head content* and *body content* are the content you want to place within the document's head and body. Note that the body element is always placed after the head element and that no other elements can be placed between the `html`, `head`, and `body` elements.

Tip

Enter your tags using all lowercase letters. For example, use `<html>` rather than `<HTML>`. While many browsers accept uppercase tag names, XHTML code requires tag names to be lowercase.

Reference Window | Creating the Basic Structure of an HTML Document

- Enter the following HTML tags

```
<html>
  <head>
    head content
  </head>
  <body>
    body content
  </body>
</html>
```

where *head content* and *body content* are the content you want to place within the document's head and body.

- To specify the page title, enter the following tag within the head section

```
<title>content</title>
```

where *content* is the text of the Web page title.

Now that you've learned about the basic structure of an HTML file, you can start writing the HTML code for Dave's Web page.

To create the basic structure of an HTML document:

1. Start your text editor, opening it to a blank document.
Trouble? If you don't know how to start or use your text editor, ask your instructor or technical support person for help.
2. Type the following lines of code in your document. Press the **Enter** key after each line. Press the **Enter** key twice for a blank line between lines of code. See Figure 1-5.

```
<html>

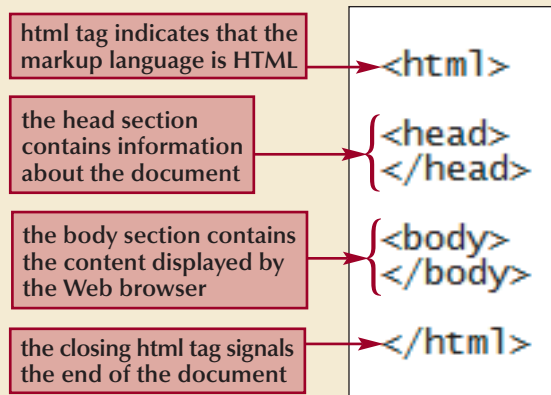
<head>
</head>

<body>
</body>

</html>
```

Basic structure of an HTML document

Figure 1-5



3. Save the file as **dave.htm** in the tutorial.01\tutorial folder included with your Data Files.

Trouble? If you are using the Windows Notepad text editor to create your HTML file, make sure you don't save the file with the extension `.txt`, which is the default file extension for Notepad. Instead, make sure you save the file with the file extension `.htm` or `.html`. Using the incorrect file extension might make the file unreadable to Web browsers, which require file extensions of `.htm` or `.html`.

Tip

To make it easier to link to your Web pages, follow the Internet convention in which HTML filenames and folder names use only lowercase letters with no spaces.

There is considerable overlap between HTML and XHTML. You can quickly change an HTML document into an XHTML document just by altering the first three lines of code. To convert an HTML file into an XHTML file, replace the opening `<html>` tag with the following three lines of code:

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
```

Each line has an important role in converting the HTML document into XHTML. XHTML documents are written in XML, so the first line notifies the browser that the document is an XML file. The version number—1.0—tells the browser that the file is written in XML 1.0.

XHTML files differ from HTML files in that XHTML files have to be tested against a set of rules that define exactly which markup tags are allowed and how they can be used. To reference the set of rules, you have to include a DOCTYPE declaration in the second line of the file, indicating the collection of rules to be used. XHTML documents can be tested against several different rules. The code sample above assumes a strict interpretation of the rules is being enforced.

The third line of the file contains the opening `<html>` tag. In XHTML, the `<html>` tag must include what is known as a namespace declaration indicating that any markup tags in the document should, by default, be considered part of the XHTML language. This is necessary because XML documents can contain a mixture of several different markup languages and there must be a way of defining the default language of the document.

With these three lines in place, browsers recognize the file as an XHTML rather than an HTML document. After these three lines, there is little difference between the code in an HTML file and in an XHTML file.

Defining the Page Title

One of the elements you can add to the document head is the document title. The syntax of the document title is

```
<title>document title</title>
```

where *document title* is the text of the document title. The document title is not displayed within the page, but is usually displayed in the browser's title bar. The document title is also used by search engines like Google or Yahoo! to report on the contents of the file.

To add a title to a Web page:

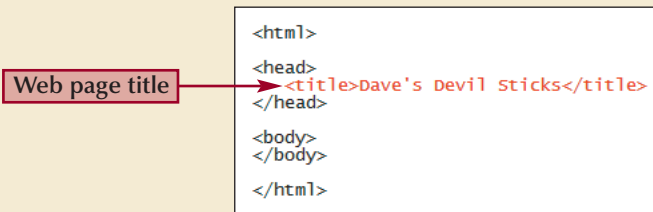
1. Click at the end of the `<head>` tag, and then press the **Enter** key to insert a new line in your text editor.
2. Press the **Spacebar** three times to indent the new line of code, and then type `<title>Dave's Devil Sticks</title>` as shown in Figure 1-6.

Tip

Indent your markup tags and insert extra blank spaces as shown in this book to make your code easier to read. It does not affect how the page is rendered by the browser.

Defining the page title

Figure 1-6



Adding Comments

As you create a Web page, you might want to add notes or comments about your code. These comments might include the name of the document's author and the date the document was created. Such notes are not intended to be displayed by the browser, but are instead used to help explain your code to yourself and others. To add notes or comments, insert a **comment tag** with the syntax

```
<!-- comment -->
```

where *comment* is the text of the comment or note. For example, the following code inserts a comment describing the page you'll create for Dave's business:

```
<!-- Page created for Dave Vinet's devil stick business -->
```

A comment can also be spread out over several lines as follows:

```
<!-- Dave's Devil Sticks
     A Web page created for Dave Vinet -->
```

Because they are ignored by the browser, comments can be added anywhere within the HTML document.

Adding an HTML Comment

| Reference Window

- To insert an HTML comment anywhere within your document, enter

```
<!-- comment -->
```

where *comment* is the text of the HTML comment.

You'll add a comment to the head of Dave's file indicating its purpose, author, and date created.

To add a comment to Dave's file:

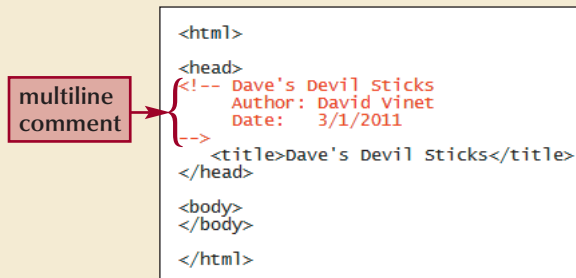
1. Click at the end of the `<head>` tag, and then press the **Enter** key to insert a new line directly above the title element you've just entered.
2. Type the following lines of code, as shown in Figure 1-7

```
<!-- Dave's Devil Sticks
     Author: your name
     Date:  the date
-->
```

where *your name* is your name and *the date* is the current date.

Figure 1-7

Adding a comment tag



```
<html>
<head>
<!-- Dave's Devil sticks
      Author: David Vinet
      Date: 3/1/2011
-->
<title>Dave's Devil sticks</title>
</head>
<body>
</body>
</html>
```

Displaying an HTML File

As you continue modifying the HTML code, you should occasionally view the page with your Web browser to verify that you have not introduced any errors. You might even want to view the results using different browsers to check for compatibility. In this book Web pages are displayed using the Windows Internet Explorer 7.0 browser. Be aware that if you are using a different browser or a different operating system, you might see slight differences in the layout and appearance of the page.

To view Dave's Web page:

1. Save your changes to the **dave.htm** file.
2. Start your Web browser. You do not need to be connected to the Internet to view local files stored on your computer.

Trouble? If you start your browser and are not connected to the Internet, you might get a warning message. Click the OK button to ignore the message and continue.

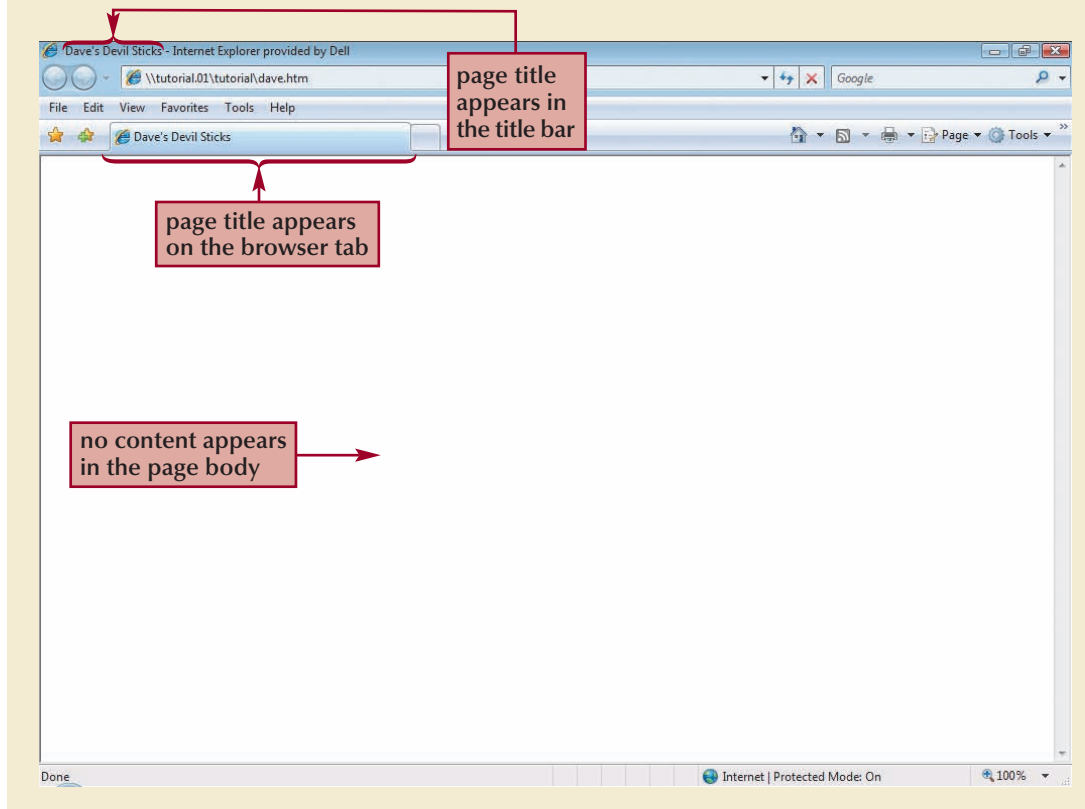
3. After your browser loads its home page, open the **dave.htm** file from the tutorial.01\tutorial folder.

Trouble? If you're not sure how to open a local file with your browser, check for an Open or Open File command under the browser's File menu. If you are still having problems accessing the dave.htm file, talk to your instructor or technical resource person.

Your browser displays the Web page shown in Figure 1-8. Note that the page title appears in the browser's title bar; and if your browser supports tabs, it also appears in the tab title. The page itself is empty because you have not yet added any content to the body element.

Viewing a page title in the browser window

Figure 1-8



You've completed the head section of the HTML document. In the next session, you'll define the elements that are displayed in the body section. For now, you can close your files and Web browser if you want to take a break before starting the next session.

Session 1.1 Quick Check

Review

1. What is a hypertext document?
2. What is a Web server? What is a Web browser? Describe how they work together.
3. How do HTML documents differ from documents created with a word processor such as Word or WordPerfect?
4. What is a deprecated feature?
5. What element do you use to mark the beginning and end of an HTML document?
6. What code would you enter in your document to set the page title to "Technical Support"? Where would you enter this code?
7. Specify the code needed to add the comment "Page Updated on 4/15/2011" to an HTML file.
8. What error was made in the following HTML code?:

```
<head>
  <title>Customer Comments Form
</head>
</title>
```

Session 1.2

Working with Block-Level Elements

You're now ready to begin entering content into the body of Dave's Web page. The first elements you'll add are **block-level elements**, which are elements that contain content that is viewed as a distinct block within the Web page. When rendered visually, block-level elements start on a new line in the document. Paragraphs are one example of a block-level element. To explore block-level and other HTML elements, a demo page has been prepared for you.

To open the HTML Tags demo page:

- ▶ 1. Use your browser to open the **demo_html.htm** file from the tutorial.01\demo folder.
- ▶ 2. If your browser prompts you to allow code on the Web page to be run, click the **OK** button.

Working with Headings

The first block-level elements you'll explore are heading elements. **Heading elements** are elements that contain the text of main headings on the Web page. They are often used for introducing new topics or dividing the page into topical sections. The syntax to mark a heading element is

```
<hn>content</hn>
```

where n is an integer from 1 to 6. Content marked with the `<h1>` tag is considered a major heading and is usually displayed in large bold text. Content marked with `<h2>` down to `<h6>` tags is used for subheadings and is usually displayed in progressively smaller bold text. To see how these headings appear on your computer, use the demo page.

Reference Window | Marking Block-Level Elements

- To mark a heading, enter
`<hn>content</hn>`
where n is an integer from 1 to 6 and *content* is the text of heading.
- To mark a paragraph, enter
`<p>content</p>`
- To mark a block quote, enter
`<blockquote>content</blockquote>`
- To mark a generic block-level element, enter
`<div>content</div>`

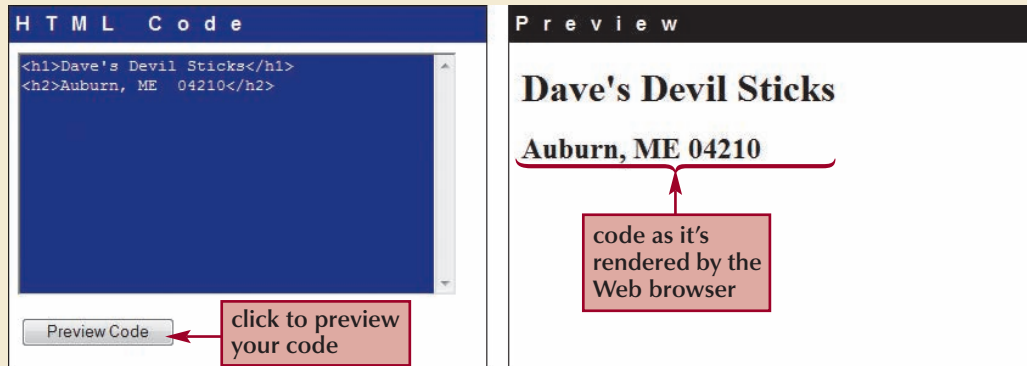
To view heading elements:

- ▶ 1. Click in the blue box on the bottom left of the demo page, type **<h1>Dave's Devil Sticks</h1>** and then press the **Enter** key to go to a new line.
- ▶ 2. Type **<h2>Auburn, ME 04210</h2>**.

- ▶ 3. Click the **Preview Code** button located below the blue code window. Your browser displays a preview of how this code would appear in your Web browser (see Figure 1-9).

Marking an h1 and h2 element

Figure 1-9



Trouble? If you are using a browser other than Internet Explorer 7.0 running on Windows Vista, your screen might look slightly different from that shown in Figure 1-9.

- ▶ 4. To see how an h3 heading would look, change the opening and closing tags for the store's city and state information from `<h2>` and `</h2>` to `<h3>` and `</h3>`. Click the **Preview Code** button again.

Your browser renders the code again, this time with the city and state information displayed in a smaller font. If you continued to change the heading element from h2 down to h6, you would see the text in the Preview box get progressively smaller.

It's important not to treat markup tags as simply a way of formatting the Web page. The h1 through h6 elements are used to identify headings, but the exact appearance of these headings depends on the browser and the device being used. Remember that the headings might not even be displayed visually. A browser that renders content aurally might convey an h1 heading using increased volume preceded by an extended pause.

Now that you've seen how to mark page headings, you can add some to Dave's Web page. Dave has three headings he wants to add to his document. The first is an h1 heading that will contain the company's name. The other two are h2 headings that preface two different sections of the document: one titled "Who Am I?" and the other titled "My Products."

To add headings to Dave's document:

- ▶ 1. Return to the **dave.htm** file in your text editor.
- ▶ 2. Between the opening and closing `<body>` tags, insert the following code:

```
<h1>Dave's Devil Sticks</h1>
<h2>Who Am I?</h2>
<h2>My Products</h2>
```

Indent your code to make it easy to read, as shown in Figure 1-10.

Figure 1-10 Adding <h1> and <h2> markup tags

```
<body>
  <h1>Dave's Devil Sticks</h1>
  <h2>Who Am I?</h2>
  <h2>My Products</h2>
</body>
```

3. Save your changes to the file, and then reload or refresh the **dave.htm** file in your Web browser. Figure 1-11 shows the revised Web page.

Figure 1-11 Headings on the Web page

Dave's Devil Sticks ← h1 heading

Who Am I?

My Products

← h2 headings

The next block-level elements you'll add are paragraphs about Dave's company and its wares.

Marking Paragraph Elements

As you saw earlier, you can mark a paragraph element using the <p> tag

```
<p>content</p>
```

where *content* is the content of the paragraph. When rendered in a browser, paragraphs are started on a new line. In older HTML code, you might occasionally see paragraphs marked with only the opening <p> tag but without a closing tag. In those situations, the <p> tag marks the start of each new paragraph. While this convention is still accepted by many browsers, it does violate HTML's syntax rules; if you want XHTML-compliant code, you must include the closing tags.

To add two paragraphs to Dave's Web page:

1. Return to the **dave.htm** file in your text editor.
2. Directly below the Who Am I? h2 heading, insert a new line, and then type the following code, as shown in Figure 1-12:

```
<p>Welcome to Dave's Devil Sticks. If you are looking for juggling balls, hacky sacks, pins, unicycles, or magic hats, you've come to the wrong place; but if you're looking for high-quality, hand-crafted devil and flower sticks, this is the site for you. I've been designing and building sticks for the past 20 years, and I know that my sticks are the best of their kind.</p>
```

```
<p>Every set is checked and tested before being shipped out to assure perfect quality. I take pride in every one of my sticks and I want my customers to feel that same pride.</p>
```

Marking paragraph elements

Figure 1-12

paragraphs

```

<body>
  <h1>Dave's Devil Sticks</h1>
  <h2>who Am I?</h2>
  <p>welcome to Dave's Devil sticks. If you are looking for juggling balls,
hacky sacks, pins, unicycles, or magic hats, you've come to the wrong
place; but if you're looking for high-quality, hand-crafted devil and
flower sticks, this is the site for you. I've been designing and building
sticks for the past 20 years, and I know that my sticks are the best of
their kind.</p>
  <p>Every set is checked and tested before being shipped out to assure perfect
quality. I take pride in every one of my sticks and I want my customers to
feel that same pride.</p>
  <h2>My Products</h2>
</body>

```

Trouble? Don't worry if your lines do not wrap at the same locations shown in Figure 1-12. As you'll see shortly, line wrap in the HTML code does not affect how the page is rendered by the browser.

3. Save your changes to the file and then refresh the **dave.htm** file in your Web browser. Figure 1-13 shows the new paragraphs added to the Web page.

Paragraphs added to Dave's Web page

Figure 1-13

Dave's Devil Sticks

Who Am I?

Welcome to Dave's Devil Sticks. If you are looking for juggling balls, hacky sacks, pins, unicycles, or magic hats, you've come to the wrong place; but if you're looking for high-quality, hand-crafted devil and flower sticks, this is the site for you. I've been designing and building sticks for the past 20 years, and I know that my sticks are the best of their kind.

Every set is checked and tested before being shipped out to assure perfect quality. I take pride in every one of my sticks and I want my customers to feel that same pride.

My Products

paragraphs

White Space and HTML

If you compare the paragraph text from the HTML code in Figure 1-12 to the way it's rendered on the Web page in Figure 1-13, you'll notice that the line returns in the code are not reflected in the Web page. When the browser renders HTML code, it ignores the presence of white space within the HTML text file. **White space** consists of blank spaces, tabs, and line breaks. As far as the browser is concerned, there is no difference between a blank space, a tab, or a line break. To explore this issue further, you'll experiment with the HTML demo page.

To explore how white space is treated by the Web browser:

- ▶ 1. Return to the **demo_html.htm** file in your Web browser.
- ▶ 2. Delete the HTML code in the left box and replace it with the following:

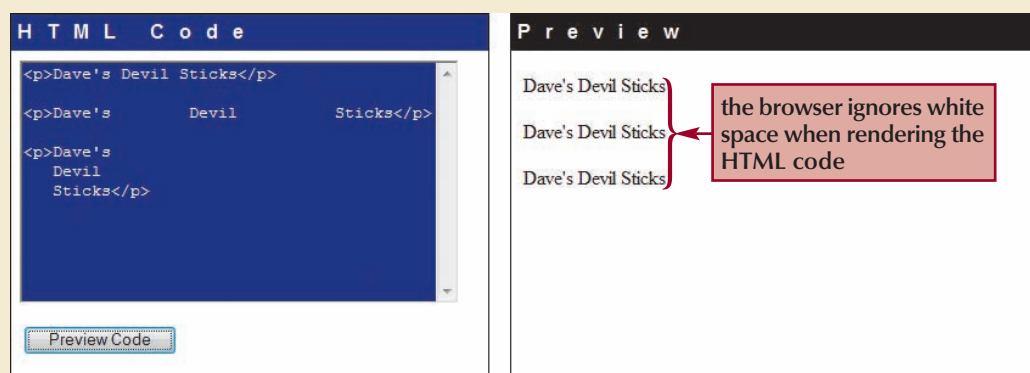

```
<p>Dave's Devil Sticks</p>

<p>Dave's           Devil           Sticks</p>

<p>Dave's
  Devil
  Sticks</p>
```
- ▶ 3. Click the **Preview Code** button. Figure 1-14 shows how the browser renders the three paragraphs of code.

Figure 1-14

Viewing the effects of white space on HTML code



Tip

To force the browser to retain the extra blank spaces, tabs, and line breaks from your HTML code, enclose the white space within a set of opening and closing `<pre>` tags.

Each paragraph is rendered exactly the same by the browser. Notice that even adding blank spaces within the paragraph does not change the appearance of the text. This is because the browser ignores extra blank spaces. Consequently, you are free to use white space however you wish as you lay out the HTML code to make it easier for you to read—without impacting the appearance of the Web page.

Marking a Block Quote

The next element that Dave wants you to enter into his document is a quote from a satisfied customer. Dave wants the quote, shown earlier in Figure 1-4, to be indented from the surrounding paragraphs to make it stand out more. However, as you just saw, any indenting you do in the HTML file will be ignored by the browser. So how do you achieve this effect? Remember that HTML is used to mark up document content, and so you need a markup tag that identifies quoted material. The syntax for marking an extended quote is

```
<blockquote>content</blockquote>
```

where *content* is the text of the quote. Most browsers will by default indent block quotes on the Web page, so you'll still achieve the visual effect that Dave wants. Note that some browsers might display block quotes differently, and the only way to ensure that block quotes are always indented is by using styles. You'll explore how to apply styles shortly.

To create a block quote:

1. Return to the **dave.htm** file in your text editor.
2. At the end of the first paragraph, directly *before* the closing `</p>` tag, insert a space and then type the following text:

Don't take my word for it; read the following testimonial:
3. Between the closing `</p>` tag from the first paragraph and the opening `<p>` tag of the second paragraph, insert the following code, as shown in Figure 1-15:

```
<blockquote>

  <p>I'm more than happy to recommend Dave Vinet's products. I came
  upon his work 10 years ago and was immediately impressed by his
  craftsmanship. I've been using his sticks in my shows ever since.
  They're durable, well-balanced, and attractive props and are
  the perfect complement to my performances. Thanks Dave!</p>

  <p>Thomas Gage, lead performer at Circus England</p>

</blockquote>
```

Marking a block quote

Figure 1-15

block quote

```
<p>welcome to Dave's Devil Sticks. If you are looking for juggling balls,
hacky sacks, pins, unicycles, or magic hats, you've come to the wrong
place; but if you're looking for high-quality, hand-crafted devil and
flower sticks, this is the site for you. I've been designing and building
sticks for the past 20 years, and I know that my sticks are the best of
their kind. Don't take my word for it; read the following testimonial:</p>

<blockquote>
  <p>I'm more than happy to recommend Dave Vinet's products. I came upon his
  work 10 years ago and was immediately impressed by his craftsmanship.
  I've been using his sticks in my shows ever since. They're durable,
  well-balanced, and attractive props and are the perfect complement to
  my performances. Thanks Dave!</p>

  <p>Thomas Gage, lead performer at Circus England</p>
</blockquote>

<p>Every set is checked and tested before being shipped out to assure perfect
quality. I take pride in every one of my sticks and I want my customers to
feel that same pride.</p>
```

4. Save your changes to the file, and then reload **dave.htm** in your Web browser. Figure 1-16 shows the revised page with the quoted material.

Dave's Web page with customer comment

Figure 1-16

Dave's Devil Sticks

Who Am I?

Welcome to Dave's Devil Sticks. If you are looking for juggling balls, hacky sacks, pins, unicycles, or magic hats, you've come to the wrong place; but if you're looking for high-quality, hand-crafted devil and flower sticks, this is the site for you. I've been designing and building sticks for the past 20 years, and I know that my sticks are the best of their kind. Don't take my word for it; read the following testimonial:

I'm more than happy to recommend Dave Vinet's products. I came upon his work 10 years ago and was immediately impressed by his craftsmanship. I've been using his sticks in my shows ever since. They're durable, well-balanced, and attractive props and are the perfect complement to my performances. Thanks Dave!

Thomas Gage, lead performer at Circus England

Every set is checked and tested before being shipped out to assure perfect quality. I take pride in every one of my sticks and I want my customers to feel that same pride.

block quote

Note that the customer quote also included two paragraph elements nested within the blockquote element. The indentation applied by the browser to the block quote was also applied to any content within that element, so those paragraphs were indented even though browsers do not indent paragraphs by default.

Marking a List

Dave has a list of products that he wants to display on his Web page. This information is presented on his flyer as a bulleted list. He wants something similar on the Web site. HTML supports three kinds of lists: ordered, unordered, and definition.

Ordered Lists

Use an **ordered list** for items that must appear in a numeric order. The beginning of an ordered list is marked by the `` (ordered list) tag. Each item within that ordered list is subsequently marked using the `` (list item) tag. The syntax of an ordered list is therefore

```
<ol>
  <li>item1</li>
  <li>item2</li>
  ...
</ol>
```

where *item1*, *item2*, and so forth are the items in the list. To explore creating an ordered list, return to the HTML demo page.

To create an ordered list:

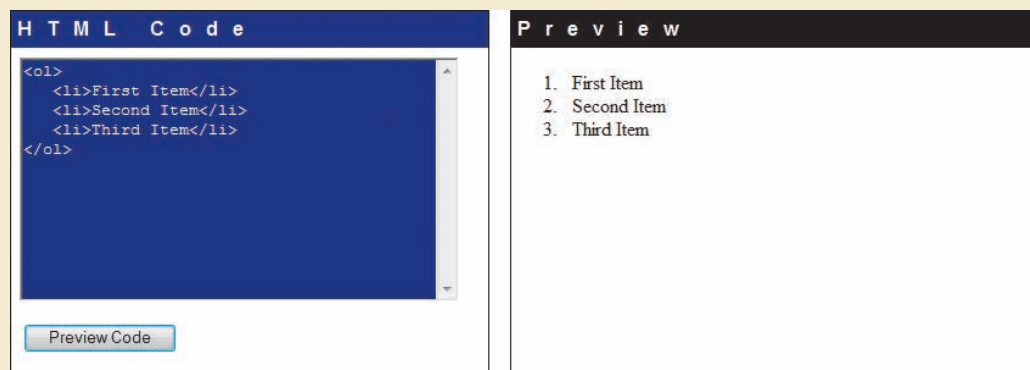
1. Return to the **demo_html.htm** file in your Web browser.
2. Delete the HTML code in the left box and replace it with the following:

```
<ol>
  <li>First Item</li>
  <li>Second Item</li>
  <li>Third Item</li>
</ol>
```

3. Click the **Preview Code** button. Figure 1-17 shows how the browser renders the ordered list contents.

Figure 1-17

Viewing an ordered list



By default, entries in an ordered list are numbered, with the numbers supplied automatically by the browser.

Unordered Lists

To mark a list in which the items do not need to occur in any special order, create an **unordered list**. The structure of ordered and unordered lists is the same, except that the list contents are contained with a set of `` (unordered list) tags:

```
<ul>
  <li>item1</li>
  <li>item2</li>
  ...
</ul>
```

Try creating an unordered list with the demo page.

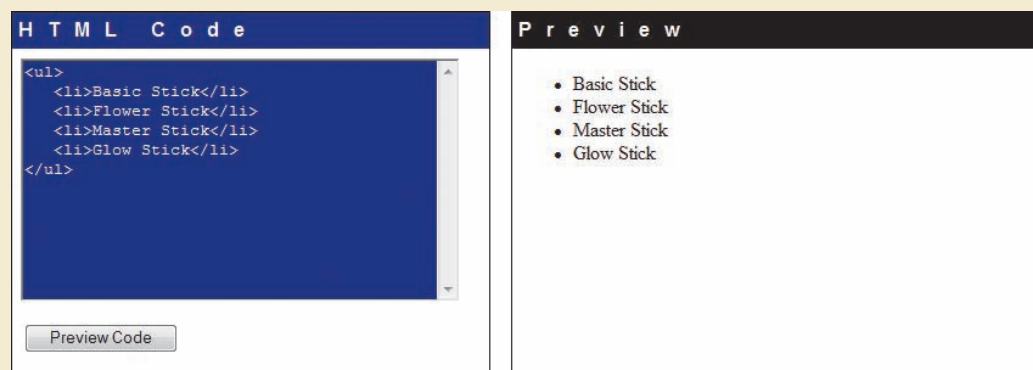
To create an unordered list:

1. Delete the HTML code in the left box and replace it with the following:

```
<ul>
  <li>Basic Stick</li>
  <li>Flower Stick</li>
  <li>Master Stick</li>
  <li>Glow Stick</li>
</ul>
```
2. Click the **Preview Code** button. Figure 1-18 shows how the browser renders the unordered list.

Viewing an unordered list

Figure 1-18



Trouble? On some browsers, the list appears with diamond shapes rather than circular bullets.

By default, unordered lists appear as bulleted lists. The exact bullet marker depends on the browser. Most browsers use a filled-in circle.

Reference Window | **Marking Lists**

- To mark an ordered list, enter

```
<ol>
  <li>item1</li>
  <li>item2</li>
  ...
</ol>
```

where *item1*, *item2*, and so forth are the items in the list.

- To mark an unordered list, use

```
<ul>
  <li>item1</li>
  <li>item2</li>
  ...
</ul>
```

- To mark a definition list, use

```
<dl>
  <dt>term1</dt>
  <dd>description1</dd>
  <dt>term2</dt>
  <dd>description2a</dd>
  <dd>description2b</dd>
  ...
</dl>
```

where *term1*, *term2*, etc. are the terms in the list and *description1*, *description2a*, *description2b*, etc. are the descriptions associated with each term.

Nesting Lists

You can place one list inside of another to create several levels of list items. The top level of the nested list contains the major items, with each sublevel containing items of lesser importance. Most browsers differentiate the various levels by using a different list symbol. Use the demo page to see how this works with unordered lists.

To create an unordered list:

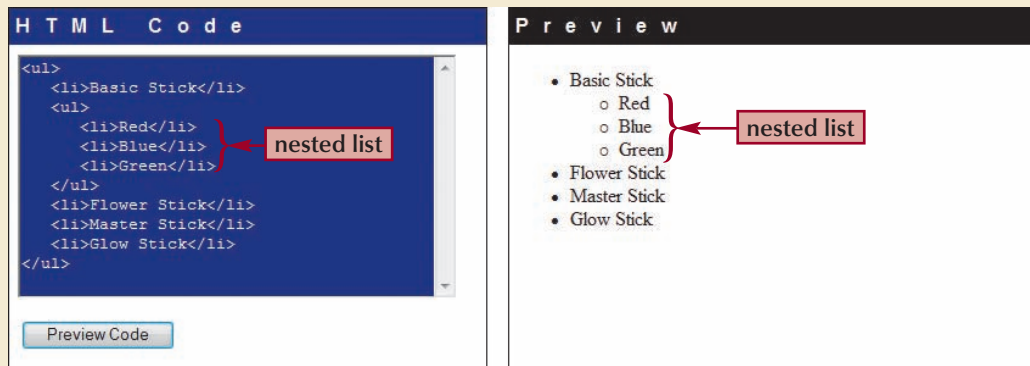
- ▶ 1. Click at the end of the `Basic Stick` line, and then press the **Enter** key to insert a new blank line.
- ▶ 2. Insert the following code between the `Basic Stick` and `Flower Stick` lines:

```
<ul>
  <li>Red</li>
  <li>Blue</li>
  <li>Green</li>
</ul>
```

- ▶ 3. Click the **Preview Code** button. Figure 1-19 shows the result of creating a nested list.

Viewing a nested list

Figure 1-19



Trouble? Depending on your browser, the sublist of basic stick colors might appear with solid bullets rather than open circles.

The lower level of items is displayed using an open circle as the list bullet and indented on the page. Once again, the exact choice of formatting a nested list is left to the browser at this point. As you continue your study of HTML you'll learn how to specify the appearance of nested lists using styles.

Definition Lists

A third type of list is the **definition list**, which contains a list of terms, each followed by the term's description. The syntax for creating a definition list is

```
<dl>
  <dt>term1</dt>
  <dd>description1</dd>
  <dt>term2</dt>
  <dd>description2a</dd>
  <dd>description2b</dd>
  ...
</dl>
```

where *term1*, *term2*, etc. are the terms in the list and *description1*, *description2a*, *description2b*, etc. are the descriptions associated with each term. Note that definition lists must follow a specified order, with each `dt` (definition term) element followed by one or more `dd` (definition description) elements.

To create a definition list:

1. Replace the code in the left box of the HTML demo page with:

```
<dl>
  <dt>Basic Stick</dt>
  <dd>Easiest stick to learn</dd>
  <dt>Flower Stick</dt>
  <dd>A graceful stick with tassels</dd>
  <dt>Master Stick</dt>
  <dd>Our most popular stick</dd>
</dl>
```

2. Click the **Preview Code** button. Figure 1-20 shows the appearance of the definition list in the browser.

Figure 1-20 Viewing a definition list

The screenshot shows two panels: 'HTML Code' and 'Preview'. The 'HTML Code' panel contains the following code:

```
<dl>
  <dt>Basic Stick</dt>
  <dd>Easiest stick to learn</dd>
  <dt>Flower Stick</dt>
  <dd>A graceful stick with tassels</dd>
  <dt>Master Stick</dt>
  <dd>Our most popular stick</dd>
</dl>
```

The 'Preview' panel shows the rendered output:

```
Basic Stick
  Easiest stick to learn
Flower Stick
  A graceful stick with tassels
Master Stick
  Our most popular stick
```

Callouts indicate that 'Basic Stick' is the 'definition term' and 'Easiest stick to learn' is the 'definition description'. A larger callout points to the descriptions, stating 'descriptions are indented by the browser'.

Tip

Definition lists can also mark dialogs, with each `dt` element naming a speaker, and each `dd` element containing the speaker's words. For long speeches, use multiple `dd` elements per speaker to break the speaker's words into separate blocks.

The demo page shows each term followed by a description that is placed in a new block below the term and indented on the page. If you had included multiple `dd` elements, each description would have been contained within its own block and indented.

Now that you've experimented with the three types of HTML lists, you'll add an unordered list of products to Dave's Web page. By default, the product names will appear as a bulleted list.

To add an unordered list to Dave's Web page:

1. Return to the **dave.htm** file in your text editor.
2. Directly below the `<h2>My Products</h2>` heading, insert the following code, as shown in Figure 1-21:

```
<ul>
  <li>Basic Stick</li>
  <li>Flower Stick</li>
  <li>Master Stick</li>
  <li>Glow Stick</li>
</ul>
```

Figure 1-21 Adding an unordered list

```
<h2>My Products</h2>
<ul>
  <li>Basic stick</li>
  <li>Flower stick</li>
  <li>Master stick</li>
  <li>Glow stick</li>
</ul>
</body>
```

3. Save your changes to the file, and then refresh the **dave.htm** file in your Web browser. As shown in Figure 1-22, the list of products appears as a bulleted list at the bottom of the page.

Product list on Dave's Web page

Figure 1-22

Dave's Devil Sticks

Who Am I?

Welcome to Dave's Devil Sticks. If you are looking for juggling balls, hacky sacks, pins, unicycles, or magic hats, you've come to the wrong place; but if you're looking for high-quality, hand-crafted devil and flower sticks, this is the site for you. I've been designing and building sticks for the past 20 years, and I know that my sticks are the best of their kind. Don't take my word for it; read the following testimonial:

I'm more than happy to recommend Dave Vinet's products. I came upon his work 10 years ago and was immediately impressed by his craftsmanship. I've been using his sticks in my shows ever since. They're durable, well-balanced, and attractive props and are the perfect complement to my performances. Thanks Dave!

Thomas Gage, lead performer at Circus England

Every set is checked and tested before being shipped out to assure perfect quality. I take pride in every one of my sticks and I want my customers to feel that same pride.

My Products

- Basic Stick
- Flower Stick
- Master Stick
- Glow Stick

bulleted list
of products

Exploring Other Block-Level Elements

HTML supports several other block-level elements you'll find useful. Dave wants to display the company's address at the bottom of the body of his page. Contact information like addresses can be marked using the `<address>` tag

```
<address>content</address>
```

where *content* is the contact information. Most browsers render addresses in italics, and some also indent or right-justify addresses. You'll use the address element to display the address of Dave's company.

To add an address to the bottom of Dave's Web page:

1. Return to the **dave.htm** file in your text editor.
2. Directly above the `</body>` tag, insert the following code, as shown in Figure 1-23:

```
<address>Dave's Devil Sticks
    541 West Highland Dr.
    Auburn, ME 04210
    (207) 555 - 9001
</address>
```

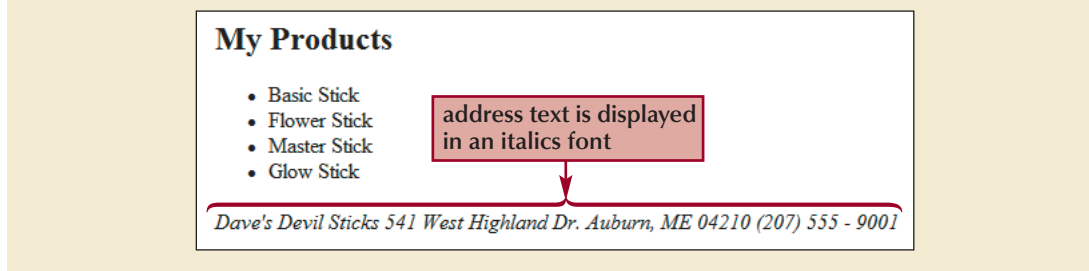
Adding an address element

Figure 1-23

```
<address>Dave's Devil Sticks
    541 west Highland Dr.
    Auburn, ME 04210
    (207) 555 - 9001
</address>
</body>
</html>
```

3. Save your changes to the file, and then refresh **dave.htm** in your Web browser. Figure 1-24 shows the revised page with the address text.

Figure 1-24 Address text on Dave's Web page



The address text appears in italics at the bottom of the page. Note that the company name, street address, city, state, and phone number all appear to run together. Remember that the browser ignores the occurrence of line breaks, tabs, and other white space in your text document. In the next session, you'll learn how to make this text more readable. For now, you'll leave the address text as it is.

At this point, you're done adding block-level elements to Dave's Web page. Figure 1-25 summarizes the properties and uses of HTML's block-level elements, including information on some block-level elements you did not add to Dave's document.

Figure 1-25 Block-level elements

Block-Level Element	Marks	Usual Visual Appearance
<code><address> ... </address></code>	Contact information	<i>Italicized text</i>
<code><blockquote> ... </blockquote></code>	An extended quotation	Plain text indented from the left and right
<code><center> ... </center></code>	Text horizontally centered with the block (deprecated)	Plain text, centered
<code><dd> ... </dd></code>	A definition description	Plain text
<code><dir> ... </dir></code>	A multicolumn directory list (deprecated)	Plain text
<code><div> ... </div></code>	A generic block-level element	Plain text
<code><dl> ... </dl></code>	A definition list	Plain text
<code><dt> ... </dt></code>	A definition term from a definition list	Plain text
<code><hn> ... </hn></code>	A heading where n is a value from 1 to 6 with h1 as the most prominent heading and h6 the least prominent	Boldfaced text of various font sizes
<code> ... </code>	A list item from an ordered or unordered list	Bulleted or numbered text
<code><menu> ... </menu></code>	A single column menu list (deprecated)	Plain text
<code> ... </code>	An ordered list	Plain text
<code><p> ... </p></code>	A paragraph	Plain text
<code><pre> ... </pre></code>	Preformatted text, retaining all white space and special characters	Fixed width text
<code> ... </code>	An unordered list	Plain text

Working with Inline Elements

Block-level elements place their content starting on a new line within the page. Another type of element is an **inline element**, which marks a section of text within a block-level element. If you think of a block-level element as a paragraph, an inline element is like a phrase or a collection of characters within that paragraph. Inline elements do not start out on a new line or block, but instead flow “in-line” with the rest of the characters in the block.

Character Formatting Elements

Inline elements are often used to format characters and words. For example, you can use an inline element to make a name or title appear in **boldface** letters or *italics*. Inline elements used in this fashion are referred to as **character formatting elements**. Figure 1-26 describes some of the inline elements supported by HTML.

Inline elements

Figure 1-26

Inline Element	Marks	Usual Visual Appearance
<abbr> ... </abbr>	An abbreviation	Plain text
<acronym> .. </acronym>	An acronym	Plain text
 ... 	Boldfaced text	Boldfaced text
<big> ... </big>	Big text	Larger text
<cite> ... </cite>	A citation	<i>Italicized text</i>
<code> ... </code>	Program code	Fixed width text
 ... 	Deleted text	Strikethrough text
<dfn> ... </dfn>	A definition term	<i>Italicized text</i>
 ... 	Emphasized content	<i>Italicized text</i>
<i> ... </i>	Italicized text	<i>Italicized text</i>
<ins> ... </ins>	Inserted text	<u>Underlined text</u>
<kbd> ... </kbd>	Keyboard-style text	Fixed width text
<q> ... </q>	Quoted text	“Quoted text”
<s> ... </s>	Strikethrough text (Deprecated)	Strikethrough text
<samp> ... </samp>	Sample computer code	Fixed width text
<small> ... </small>	Small text	Smaller text
 ... 	A generic inline element	Plain text
<strike> ... </strike>	Strikethrough text (Deprecated)	Strikethrough text
 ... 	Strongly emphasized content	Boldfaced text
_{...}	Subscripted text	Subscripted text
^{...}	Superscripted text	Superscripted text
<tt> ... </tt>	Teletype text	Fixed width text
<u> ... </u>	Underlined text (Deprecated)	<u>Underlined text</u>
<var> ... </var>	Programming variables	<i>Italicized text</i>

To see how to use inline elements in conjunction with block-level elements, you’ll return to the HTML demo page.

To explore the use of inline elements:

- ▶ 1. Return to the **demo_html.htm** file in your Web browser.
- ▶ 2. In the left box, enter the HTML code:

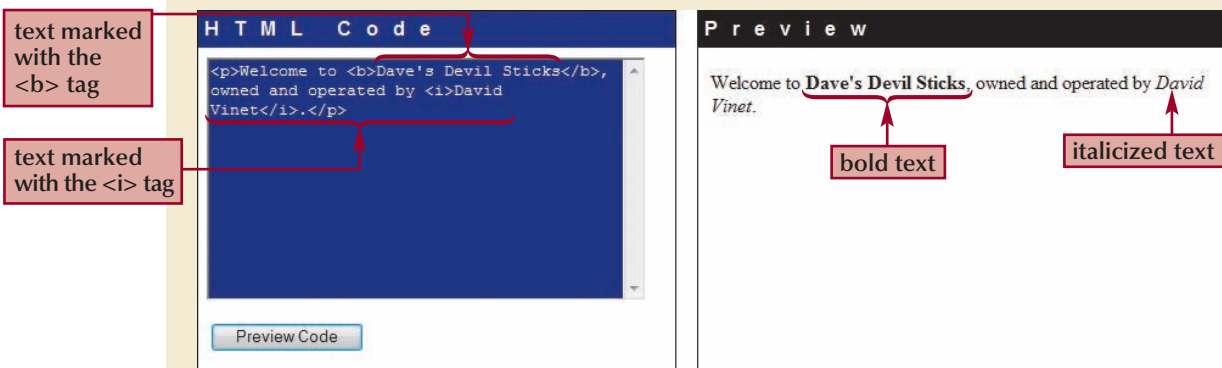

```
<p>Welcome to Dave's Devil Sticks, owned and operated by David Vinet.</p>
```
- ▶ 3. Click the **Preview Code** button to display this paragraph in the Preview box.

To mark "Dave's Devil Sticks" as boldface text, you can enclose that phrase within a set of `` tags.
- ▶ 4. Insert a `` tag directly before the word "Dave's" in the box on the left. Insert the closing `` tag directly after the word "Sticks." Click the **Preview Code** button to confirm that "Dave's Devil Sticks" is now displayed in bold.

You can use the `<i>` tag to mark italicized text. Try this now by enclosing "David Vinet" within a set of `<i>` tags.
- ▶ 5. Insert an `<i>` tag directly before the word "David" and insert the closing `</i>` tag directly after "Vinet". Click the **Preview Code** button to view the revised code.

Figure 1-27 shows the result of applying the `` and `<i>` tags to the paragraph text.

Figure 1-27 Using the `` and `<i>` tags



- ▶ 6. Close the HTML demo page.

You can nest inline elements to mark text with more than one character-formatting element. The code

```
<p>Welcome to <b><i>Dave's Devil Sticks</i></b>.</p>
```

displays "Dave's Devil Sticks" in a **bold italic** font.

Dave wants to use the `` and `<i>` tags in several locations in his document. He wants to display the name of his company and the names of all of his devil stick products in bold. He would also like to display the name of the juggling troupe, *Circus England*, in italics. Make these changes to his document now.

Marking Inline Elements

| Reference Window

- To mark boldface text, enter
`content`
 where *content* is the text to be displayed in boldface.
- To mark italicized text, use
`<i>content</i>`
- To mark text with a generic inline element, use
`content`

To mark boldface and italicized text:

- ▶ 1. Return to the **dave.htm** file in your text editor.
- ▶ 2. Go to the first paragraph of the body section and enclose the text "Dave's Devil Sticks" within a set of opening and closing `` tags.
- ▶ 3. Go to the second paragraph within the blockquote element and enclose the text "Circus England" within a set of opening and closing `<i>` tags.
- ▶ 4. Go to the unordered list and enclose each product item within a set of opening and closing `` tags. Nest the `` tags within the `` tags. Figure 1-28 highlights the revised code of Dave's document.

Marking bold and italicized text

Figure 1-28

```

<body>
  <h1>Dave's Devil sticks</h1>
  <h2>who Am I?</h2>
  <p>welcome to <b>Dave's Devil Sticks</b>. If you are looking for juggling balls,
    hacky sacks, pins, unicycles, or magic hats, you've come to the wrong
    place; but if you're looking for high-quality, hand-crafted devil and
    flower sticks, this is the site for you. I've been designing and building
    sticks for the past 20 years, and I know that my sticks are the best of
    their kind. Don't take my word for it; read the following testimonial:</p>

  <blockquote>
    <p>I'm more than happy to recommend Dave vinet's products. I came upon his
      work 10 years ago and was immediately impressed by his craftsmanship.
      I've been using his sticks in my shows ever since. They're durable,
      well-balanced, and attractive props and are the perfect complement to
      my performances. Thanks Dave!</p>

    <p>Thomas Gage, lead performer at <i>Circus England</i></p>
  </blockquote>

  <p>Every set is checked and tested before being shipped out to assure perfect
    quality. I take pride in every one of my sticks and I want my customers to
    feel that same pride.</p>

  <h2>My Products</h2>
  <ul>
    <li><b>Basic stick</b></li>
    <li><b>Flower stick</b></li>
    <li><b>Master stick</b></li>
    <li><b>Glow stick</b></li>
  </ul>

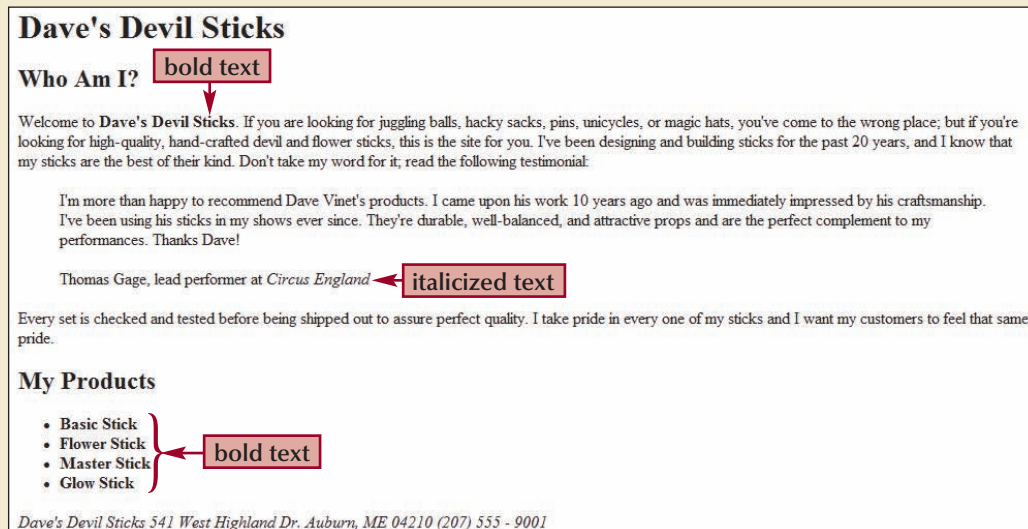
  <address>Dave's Devil Sticks
    541 West Highland Dr.
    Auburn, ME 04210
    (207) 555 - 9001
  </address>
</body>

```

- ▶ 5. Save your changes to the file.
- ▶ 6. Refresh the **dave.htm** file in your Web browser. Figure 1-29 shows the revised appearance of the Web page.

Figure 1-29

Dave's revised Web page



7. If you want to take a break before starting the next session, you can close your browser and any open files now.

Using the Generic Elements: div and span

Most of the block-level and inline elements you've examined have a specific meaning or purpose in your document. Sometimes you will want an element that represents a text block or a string of inline text without it having any other meaning. HTML supports two types of generic elements: div and span. The div element is used to mark general block-level content and has the syntax

```
<div>content</div>
```

The span element, used to mark general inline content, has the syntax

```
<span>content</span>
```

Browsers recognize both elements but do not assign any default format to content marked with those elements. Web authors like using the div and span elements because they know they can completely control the appearance of the content through the use of styles. This is not the case with elements such as addresses or headings, which have default formats assigned to them by the Web browser.

CENGAGE **brain**

Logical Elements vs. Physical Elements

InSight

As you learn more HTML, you'll notice some overlap in how certain elements are displayed by the browser. To display italicized text, you could use the `<dfn>`, ``, `<i>`, or `<var>` tags, or if you want to italicize an entire block of text, you could use the `<address>` tag. It's important to distinguish between how a browser displays an element and the element's purpose in the document. Page elements can be organized into two types: logical elements and physical elements. A logical element, marked with tags like `<cite>` or `<code>`, describes the nature of the enclosed content but not necessarily how that content should appear. A physical element, on the other hand, marked with tags like `` or `<i>`, describes how content should appear but doesn't indicate the content's nature.

While it can be tempting to use logical and physical elements interchangeably, your HTML code benefits in several ways when you respect the distinction. For one, different browsers can and do display logical elements differently. For example, both Netscape's browser and Internet Explorer display text marked with the `<cite>` tag in italics, but the text-based browser Lynx displays the citation text using a fixed width font. An aural browser that doesn't render pages visually might increase the volume when it encounters cited text. In addition, Web programmers can also use logical elements to extract information from a page. For example, a program could automatically generate a bibliography from all of the citations listed within a Web site.

In general, you should use a logical element that accurately describes the enclosed content whenever possible, letting the browser determine the appearance based on its function, and use physical elements only for general content.

You're finished working with block-level elements and inline elements. In the next session you'll learn how to add images to the document as well as how to use styles to control the appearance of your Web pages.

Session 1.2 Quick Check

Review

1. What is the difference between a block-level element and an inline element?
2. If you want to add an extra blank line between paragraphs on your Web page, why can't you simply add an extra blank line to the HTML file?
3. Specify the code to mark the main heading on your Web page.
4. Specify the tag to mark an extended quotation. How would that quotation be rendered in most visual browsers?
5. Specify the code you would use to display the seasons of the year (Winter, Spring, Summer, and Fall) as an unordered list.
6. The following is a dialog from Shakespeare's *Hamlet*. Indicate how you would use a definition list to mark up this text, distinguishing between the speaker and the lines spoken.
HAMLET
There's ne'er a villain dwelling in all Denmark; but he's an arrant knave.
HORATIO
There needs no ghost, my lord, come from the grave to tell us this.
7. What code would you enter to display the following text as a paragraph in your Web page? Include both the block-level and inline element tags.
Hamlet, a play by William Shakespeare.
8. What are the two generic page elements?

Session 1.3

Using Element Attributes

So far you've used markup tags only to create Dave's Web page. However, many markup tags contain **attributes** that control the use, behavior, and in some cases the appearance of elements in the document. You apply an attribute to an element by adding it to the element's markup tag using the syntax

```
<element attribute1="value1" attribute2="value2" ...>content</element>
```

where *attribute1*, *attribute2*, etc. are the names of attributes associated with the element and *value1*, *value2*, etc. are the values of those attributes. You can list attributes in any order, but you must separate them from one another with white space.

One attribute that is associated with most elements is the `id` attribute, which uniquely identifies the element in the Web page. The following code assigns the `id` value of "mainhead" to the `h1` heading "Dave's Devil Sticks," distinguishing it from other `h1` headings that might exist in the document:

```
<h1 id="mainhead">Dave's Devil Sticks</h1>
```

You'll learn more about the `id` attribute in the next tutorial. For a list of attributes associated with each element, you can also refer to the appendices.

Tip

Attribute names should be entered in lowercase letters to be completely compliant with the syntax rules of XHTML. Attribute values must be enclosed within single or double quotation marks.

Reference Window | Adding an Attribute to an Element

- To add an element attribute, enter


```
<element attribute1="value1" attribute2="value2" ...>content</element>
```

 where *attribute1*, *attribute2*, etc. are the names of attributes associated with the element and *value1*, *value2*, etc. are the values of those attributes.

The Style Attribute

Another important attribute is the `style` attribute. As you've seen, an element's appearance on the Web page is determined by the browser. If you want to change how the browser displays an element, you can use the `style` attribute. The syntax of the `style` attribute is

```
<element style="rules" ...>content</element>
```

where *rules* is a set of style rules. Style rules are entered by specifying a style name followed by a colon and then a style value. You can have multiple style rules with each style name/value pair, separated from each other by a semicolon. The general form of the `style` attribute is therefore

```
style="name1:value1; name2:value2; ..."
```

where *name1*, *name2*, etc. are style names and *value1*, *value2* and so forth are the values of those styles.

As you proceed in your study of HTML you'll learn more about styles and how to apply them. For now you'll focus only on a few basic ones. The first is a style to align text. As you may have noticed, Web page text is usually aligned with the page's left margin. To choose a different alignment, you can apply the following text-align style to the element

```
style="text-align: alignment"
```

where *alignment* is left, right, center, or justify. For example, to center an h1 heading, you would enter the following markup tag:

```
<h1 style="text-align: center"> ... </h1>
```

A second style you'll explore defines the text color used in an element. Most browsers display text in a black font. To apply a different text color, use

```
style="color: color"
```

where *color* is a color name such as red, blue, green, and so forth. Applying the following attribute to an h1 heading causes the browser to render the heading text in a red font:

```
<h1 style="color: red"> ... </h1>
```

You can both center the text and change its font color to red by combining the two styles in one style attribute:

```
<h1 style="text-align: center; color: red"> ... </h1>
```

Applying the Style Attribute

| Reference Window

- To add the style attribute, in the opening tag enter
`style="name1:value1; name2:value2; ..."`
where *name1*, *name2*, etc. are style names and *value1*, *value2* and so forth are the values of those styles.
- To center text horizontally, use
`style="text-align: alignment"`
where *alignment* is left, right, center, or justify.
- To set the font color, use
`style="color: color"`
where *color* is a color name.

To explore how to apply these two styles to a page element, you'll return to the HTML demo page.

To explore the style attribute:

- ▶ 1. Return to the **demo_html.htm** file in your Web browser.
- ▶ 2. Enter the following code in the left box, and then click the **Preview Code** button:

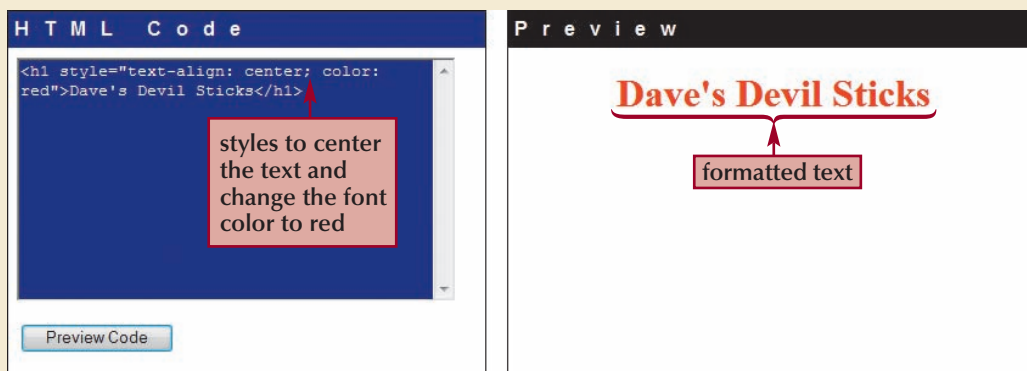
```
<h1>Dave's Devil Sticks</h1>
```

The demo page displays Dave's Devil Sticks as an h1 heading in the Preview box. Now you'll change the font color to red and center this heading in the box.

- ▶ 3. Within the opening `<h1>` tag, insert a space after "h1," and then type the attribute `style="text-align: center; color: red"`
- ▶ 4. Click the **Preview Code** button. As shown in Figure 1-30, the h1 heading is now centered and displayed in a red font.

Figure 1-30

Applying styles to an element



- ▶ 5. Continue exploring the text-align and color styles by creating headings right-aligned or centered in a blue, green, and gray-colored text.
- ▶ 6. Close the HTML demo page when you are finished exploring these two styles.

Dave has had a chance to examine your progress on his Web page and suggests that you center the address text at the bottom of the page. Now that you've explored the uses of the style attribute, you can make this change to his document.

To apply a style to the address element:

- ▶ 1. Return to the **dave.htm** file in your text editor.
- ▶ 2. Locate the address element at the bottom of the file and insert the following style attribute, as shown in Figure 1-31:

```
style="text-align: center"
```

Figure 1-31

Apply the text-align style to an address

```
<address style="text-align: center">Dave's Devil Sticks
  541 West Highland Dr.
  Auburn, ME 04210
  (207) 555 - 9001
</address>
</body>
</html>
```

- ▶ 3. Save your changes to the file and refresh the **dave.htm** file in your Web browser. Verify that the address text is centered horizontally at the bottom of the page.

Presentational Attributes

You learned in the first session that early versions of HTML were used mostly by scientists and researchers. HTML was intended to be a language that described the structure but not necessarily the appearance of documents. Scientists and researchers didn't need flashy graphics, various fonts, or even much color on a page. The earliest Web pages weren't fancy and did not require much from the browsers that displayed them. This changed as the Web became more popular and attracted the attention of graphic designers and artists.

One way that HTML changed to accommodate this new class of users was to introduce **presentational attributes**, which are attributes that specifically describe how any element should be rendered. Rather than using styles, early versions of HTML would align text using the align attribute

```
<element align="alignment">content</element>
```

where *alignment* is either left, right, center, or justify. Thus to center an h1 heading you could use either of the following:

```
<h1 style="text-align: center">Dave's Devil Sticks</h1>
```

or

```
<h1 align="center">Dave's Devil Sticks</h1>
```

Almost all presentational attributes are now deprecated in favor of styles, but you will still see them used. Many HTML editors and converters use presentational attributes in place of styles. Even though using a deprecated attribute like align will probably not cause your Web page to fail, you should still use styles because that will ensure compatibility with future browser versions and with XHTML.

Working with Empty Elements

As he examines your work on the Web page, Dave notices that the product list you created in the last session lacks descriptions of the items. Dave wants you to take the information from his original flyer and add it to the Web page.

To add a description of each item in the product list:

- ▶ 1. Return to the **dave.htm** file in your text editor.
- ▶ 2. Locate the first closing `` tag in the unordered list directly after the text "Basic Stick." Press the **Enter** key and type the following text, indenting it to make the code easier to read:

```
The easiest stick to learn with, but "grippy" enough for the most demanding tricks. Comes in red, green, and blue.
```
- ▶ 3. Insert the following description for the Flower Stick, directly after the closing `` tag for that product name:

```
A graceful stick with colored tassels. Flower Sticks float more slowly, giving you the ideal chance to practice new moves.
```
- ▶ 4. Add the following description for the Master Stick after the closing `` tag:

```
My most popular stick is shorter and heavier for fast play and more advanced tricks. Each Master set is available in custom colors.
```

▶ 5. Finally, add the following description for the Glow Stick:

The Glow Stick shines brightly at night (without the danger of a fire stick). It combines the fun and versatility of the Master Stick, adding the fun tricks unique to a glow-in-the-dark stick.

Figure 1-32 highlights the newly added product descriptions.

Figure 1-32

Adding product descriptions

```
<h2>My Products</h2>
<ul>
  <li><b>Basic stick</b>
    The easiest stick to learn with, but "grippy" enough for the most demanding
    tricks. Comes in red, green, and blue.</li>
  <li><b>Flower stick</b>
    A graceful stick with colored tassels. Flower Sticks float more slowly, giving
    you the ideal chance to practice new moves.</li>
  <li><b>Master stick</b>
    My most popular stick is shorter and heavier for fast play and more advanced
    tricks. Each Master set is available in custom colors.</li>
  <li><b>Glow Stick</b>
    The Glow Stick shines brightly at night (without the danger of a fire stick).
    It combines the fun and versatility of the Master Stick, adding the fun tricks
    unique to a glow-in-the-dark stick.</li>
</ul>
```

▶ 6. Save your changes to the file and then refresh the **dave.htm** file in your Web browser. Figure 1-33 shows the new product descriptions as they appear in the browser.

Figure 1-33

Product descriptions on the Web page

My Products

- **Basic Stick** The easiest stick to learn with, but "grippy" enough for the most demanding tricks. Comes in red, green, and blue.
- **Flower Stick** A graceful stick with colored tassels. Flower Sticks float more slowly, giving you the ideal chance to practice new moves.
- **Master Stick** My most popular stick is shorter and heavier for fast play and more advanced tricks. Each Master set is available in custom colors.
- **Glow Stick** The Glow Stick shines brightly at night (without the danger of a fire stick). It combines the fun and versatility of the Master Stick, adding the fun tricks unique to a glow-in-the-dark stick.

Dave thinks the revised product list is difficult to read and suggests that you place the descriptions on a new line directly below the product name. To do that you'll have to insert a line break into the Web page. The line break element is an example of an **empty element** because it contains no content. Empty elements appear in code as **one-sided tags** using

```
<element />
```

where *element* is the name of the empty element. As with other markup tags, one-sided tags can also contain attributes that define how the element is used in the document. The one-sided tag to mark a line break is

```
<br />
```

Line breaks need to be placed within block-level elements such as paragraphs or headings. Some browsers accept line breaks placed anywhere within the body of the Web page, but this is not good coding technique. XHTML in particular will reject code in which an inline element like the `br` element is not placed within a block-level element.

Use the `br` element now to mark a line break between the names and descriptions in Dave's product list.

To create line breaks between the product names and descriptions:

- ▶ 1. Return to the **dave.htm** file in your text editor.
- ▶ 2. Insert a **
** tag directly after the closing **** tag for the Basic Stick product description.
- ▶ 3. Add a **
** tag after the closing **** tag for each of the remaining three product names. Figure 1-34 shows the revised HTML code.

Adding line breaks to the Web page

Figure 1-34

```
<h2>My Products</h2>
<ul>
  <li><b>Basic Stick</b><br />
    The easiest stick to learn with, but "grippy" enough for the most demanding
    tricks. Comes in red, green, and blue.</li>
  <li><b>Flower Stick</b><br />
    A graceful stick with colored tassels. Flower sticks float more slowly, giving
    you the ideal chance to practice new moves.</li>
  <li><b>Master Stick</b><br />
    My most popular stick is shorter and heavier for fast play and more advanced
    tricks. Each Master set is available in custom colors.</li>
  <li><b>Glow Stick</b><br />
    The glow stick shines brightly at night (without the danger of a fire stick).
    It combines the fun and versatility of the Master stick, adding the fun tricks
    unique to a glow-in-the-dark stick.</li>
</ul>
```

forces the browser to insert a line break before rendering the next line of text

- ▶ 4. Save your changes to the file and then refresh the Web page in your browser. Verify that each of the four product descriptions is displayed on a new line directly below the product name.

Marking a Horizontal Rule

Another useful empty element is the `hr` or horizontal rule element, which places a horizontal line across the Web page. The syntax of the `hr` element is

```
<hr />
```

The exact appearance of the horizontal rule is left to the browser. Most browsers display a gray-shaded line a few pixels in height. Horizontal rules are considered block-level elements because they are displayed starting on a new line in the Web page. The `hr` element can be nested either within the `<body>` tag, in which case the horizontal rule will extend across the width of the Web page, or within a blockquote element, in which case the horizontal rule will be indented like other contents of the block quote.

Horizontal rules are useful in breaking up a long Web page into topical sections. Dave suggests that you place a horizontal rule above and one below the customer quotation and a third at the bottom of the page directly above the company address.

To create three horizontal rules:

- ▶ 1. Return to the **dave.htm** file in your text editor.
- ▶ 2. Directly below the opening `<blockquote>` tag, press the **Enter** key to insert a blank line, and then insert an **<hr />** tag.
- ▶ 3. Insert one **<hr />** tag directly above the closing `</blockquote>` tag and another above the opening `<address>` tag. Figure 1-35 highlights the revised code.

Figure 1-35

Marking horizontal rules with the `<hr />` tag

```

<blockquote>
  <hr />
  <p>I'm more than happy to recommend Dave Vinet's products. I came upon his
  work 10 years ago and was immediately impressed by his craftsmanship.
  I've been using his sticks in my shows ever since. They're durable,
  well-balanced, and attractive props and are the perfect complement to
  my performances. Thanks Dave!</p>

  <p>Thomas Gage, lead performer at <i>Circus England</i></p>
  <hr />
</blockquote>

<p>Every set is checked and tested before being shipped out to assure perfect
quality. I take pride in every one of my sticks and I want my customers to
feel that same pride.</p>

<h2>My Products</h2>
<ul>
  <li><b>Basic stick</b><br />
    The easiest stick to learn with, but "grippy" enough for the most demanding
    tricks. Comes in red, green, and blue.</li>
  <li><b>Flower stick</b><br />
    A graceful stick with colored tassels. Flower Sticks float more slowly, giving
    you the ideal chance to practice new moves.</li>
  <li><b>Master stick</b><br />
    My most popular stick is shorter and heavier for fast play and more advanced
    tricks. Each Master set is available in custom colors.</li>
  <li><b>Glow stick</b><br />
    The Glow stick shines brightly at night (without the danger of a fire stick).
    It combines the fun and versatility of the Master stick, adding the fun tricks
    unique to a glow-in-the-dark stick.</li>
</ul>

<hr />

<address style="text-align: center;">Dave's Devil sticks
  541 West Highland Dr.
  Auburn, ME 04210
  (207) 555 - 9001
</address>
</body>

```

4. Save your changes to the file and then refresh the Web page in your browser. As shown in Figure 1-36, three horizontal rules have been added to the document, visually breaking the Web page into sections.

Figure 1-36

Adding horizontal rules to the Web page

Dave's Devil Sticks**Who Am I?**

Welcome to **Dave's Devil Sticks**. If you are looking for juggling balls, hacky sacks, pins, unicycles, or magic hats, you've come to the wrong place; but if you're looking for high-quality, hand-crafted devil and flower sticks, this is the site for you. I've been designing and building sticks for the past 20 years, and I know that my sticks are the best of their kind. Don't take my word for it; read the following testimonial:

I'm more than happy to recommend Dave Vinet's products. I came upon his work 10 years ago and was immediately impressed by his craftsmanship. I've been using his sticks in my shows ever since. They're durable, well-balanced, and attractive props and are the perfect complement to my performances. Thanks Dave!

Thomas Gage, lead performer at *Circus England*

Every set is checked and tested before being shipped out to assure perfect quality. I take pride in every one of my sticks and I want my customers to feel that same pride.

My Products

- **Basic Stick**
The easiest stick to learn with, but "grippy" enough for the most demanding tricks. Comes in red, green, and blue.
- **Flower Stick**
A graceful stick with colored tassels. Flower Sticks float more slowly, giving you the ideal chance to practice new moves.
- **Master Stick**
My most popular stick is shorter and heavier for fast play and more advanced tricks. Each Master set is available in custom colors.
- **Glow Stick**
The Glow Stick shines brightly at night (without the danger of a fire stick). It combines the fun and versatility of the Master Stick, adding the fun tricks unique to a glow-in-the-dark stick.

Dave's Devil Sticks 541 West Highland Dr. Auburn, ME 04210 (207) 555 - 9001

You show the revised page to Dave and he's pleased with the addition of the horizontal rules. They have made the text easier to read and have nicely highlighted Thomas Gage's tribute.

Inserting an Inline Image

Dave wants you to replace the name of the company with the company logo centered at the top of the page. Because HTML files are simple text files, nontextual content like graphics must be stored in separate files, which are then loaded by the browser as it renders the page. The location of the graphic is marked as an **inline image** using the one-sided tag

```

```

where *file* is the name of the graphic image file and *text* is text displayed by the browser in place of the graphic image. In this tutorial, you'll assume that the graphic image file is located in the same folder as the Web page, so you don't have to specify the location of the file. In the next tutorial, you'll learn how to reference files placed in other folders or locations on the Web.

As the name implies, inline images are another example of an inline element and thus must be placed within a block-level element such as a heading or a paragraph. Inline images are most widely stored in one of two formats: GIF (Graphics Interchange Format) or JPEG (Joint Photographic Experts Group). You can use an image editing application such as Adobe Photoshop to convert images to either of these two formats. Dave has already created such a graphic and stored it with the filename **logo.jpg**, located in the tutorial.01\tutorial folder included with your Data Files.

Tip

Always include alternate text for inline images. The alt attribute is required in XHTML code and is highly recommended as a way of accommodating users running nonvisual Web browsers.

Marking Empty Elements

| Reference Window

- To mark a line break, use
`
`
- To mark a horizontal rule, use
`<hr />`
- To mark an inline image, use
``
where *file* is the name of the graphic image file and *text* is text displayed by the browser in place of the graphic image.

To insert Dave's logo centered at the top of the page:

- ▶ 1. Return to the **dave.htm** file in your text editor.
- ▶ 2. Go to the h1 heading element at the top of the body section and insert the following attribute into the opening `<h1>` tag:

```
style="text-align: center"
```

- ▶ 3. Delete the text **Dave's Devil Sticks** from between the opening and closing `<h1>` tags and replace it with

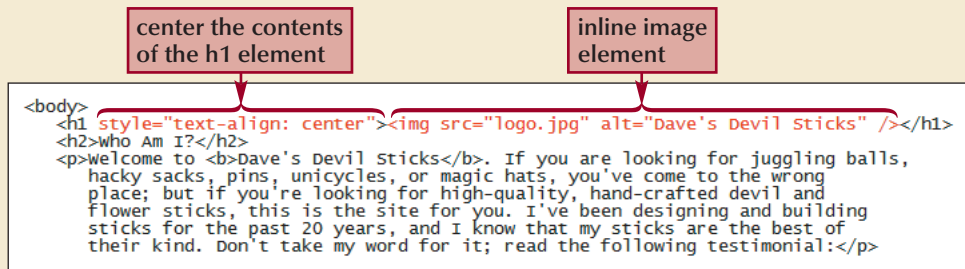
```

```

Figure 1-37 shows the revised code in the **dave.htm** file.

Figure 1-37

Adding an inline image to a Web page



```
<body>
<h1 style="text-align: center"></h1>
<h2>who Am I?</h2>
<p>welcome to <b>Dave's Devil Sticks</b>. If you are looking for juggling balls,
hacky sacks, pins, unicycles, or magic hats, you've come to the wrong
place; but if you're looking for high-quality, hand-crafted devil and
flower sticks, this is the site for you. I've been designing and building
sticks for the past 20 years, and I know that my sticks are the best of
their kind. Don't take my word for it; read the following testimonial:</p>
```

4. Save your changes to the file, and then refresh the Web page in your browser. Figure 1-38 shows the new heading with the logo centered across the page.

Figure 1-38

Viewing Dave's logo



Working with Character Sets and Special Characters

Dave likes the work you've done so far on the Web page. He has only one remaining concern: he feels that the address information at the bottom of the page is difficult to read and would like you to add a solid circular marker separating the different sections of the address. However, this marker is not represented by any keys on your keyboard. How then do you insert this symbol into the Web page?

Character Sets

To add dots to Dave's address, you must reference a symbol that your browser will be able to display but is not found on your keyboard. This is done by using a collection of characters and symbols called a **character set**. Character sets come in a wide variety of sizes, based on the number of symbols required for communication in the chosen language. For English, no more than about 127 characters are needed to represent all of the upper- and lowercase letters, numbers, punctuation marks, spaces, and special typing symbols in the English language. Other languages, such as Japanese or Chinese, require character sets containing thousands of symbols.

Each character set has a name. The character set representing the alphabet of English characters is called **ASCII (American Standard Code for Information Interchange)**. A more extended character set called **Latin-1** or the **ISO 8859-1** character set supports 255 characters and can be used by most languages that employ the Latin alphabet, including English, French, Spanish, and Italian. The most extended character set is **Unicode**, which can be used for any of the world's languages, supporting up to 65,536 symbols. The most commonly used character set on the Web is **UTF-8**, which is a compressed version of Unicode and is probably the default character set assumed by your browser. You can learn more about character sets by visiting the W3C Web site and the Web site for the Internet Assigned Numbers Authority at www.iana.org.

Numeric Character References

To store a character set, browsers need to associate each symbol with a number in a process called **character encoding**. The number is called the **numeric character reference**. For example, the copyright symbol © from the UTF-8 character set has the number 169. If you know the numeric reference, you can insert the number directly into your code to display the symbol. The syntax to insert a numeric character reference is

```
&#code;
```

where *code* is the reference number. Thus to display the © symbol in your Web page, you would enter

```
&#169;
```

into your HTML file. To render a numeric character reference correctly, the browser needs to know the character set and encoding being used in the Web page. This information is typically sent by the Web server as it transfers the HTML page to the browser; and unless you are working with specialized international documents, you usually do not have to worry about specifying the character set for the browser.

Character Entity References

Another way to insert a special symbol is to use a **character entity reference**, in which a short memorable name is used in place of the numeric character reference. The syntax to insert a character entity reference is

```
&char;
```

where *char* is the character's name. The character entity reference for the copyright symbol is "copy," so to display the © symbol in your Web page you could also insert

```
&copy;
```

into your HTML code. One of the advantages of character entity references is that browsers can use them without knowing the character set or encoding. A disadvantage is that older browsers might not recognize the character entity reference and will thus display the reference name but not the symbol it represents.

Reference Window | **Inserting Character Codes**

- To insert a character based on a numeric character reference, use `&#code;` where *code* is the character code number.
- To insert a character based on the character entity reference, use `&char;` where *char* is the name assigned to the character.
- To insert a nonbreaking space, use ` `
- To insert the < symbol, use `<`
- To insert the > symbol, use `>`

To explore various numeric character references and character entity references, you can view a demo page supplied with your Data Files.

To view the demo page:


- ▶ 1. Use your Web browser to open the **demo_characters.htm** file from the tutorial.01\demo data folder.
- ▶ 2. Type **£** in the input box at the top of the page, and then click the **Show** button. The Web browser displays the £ symbol in the ivory-colored box below. As you can see, to display the British pound symbol (£), you can use the **£** numeric character reference.
- ▶ 3. Now try to display a special symbol using a character entity reference. Replace the value in the input box with **®** and then click the **Show** button. The browser now displays the ® symbol, which is the symbol for registered trademarks.

You can view a collection of numeric character references and character entity references by selecting a table from the list box on the page.
- ▶ 4. Verify that General Symbols is displayed in the selection list box, and then click the **Show Table** button. As shown in Figure 1-39, the browser displays a list of 35 symbols with the character entity reference and numeric references displayed beneath each symbol.

HTML characters demo page

Figure 1-39

Enter a character code or character name:

 **character symbol** **character entity reference**

Select a table of characters from the list box:

General Symbols

" quot (34)	& amp (38)	< lt (60)	> gt (62)	nbsp nbsp (160)	¡ iexcl (161)	¢ cent (162)	£ pound (163)	¤ curren (164)	¥ yen (165)
 brvbar (166)	§ sect (167)	¨ uml (168)	© copy (169)	ª ordf (170)	« laquo (171)	¬ not (172)	¸ shy (173)	® reg (174)	¯ macr (175)
° deg (176)	± plusmn (177)	² ² sup2 (178)	³ ³ sup3 (179)	´ acute (180)	µ micro (181)	¶ para (182)	· middot (183)	¸ cedil (184)	¹ ¹ sup1 (185)
º ^o ordm (186)	» raquo (187)	¼ ^{1/4} frac14 (188)	½ ^{1/2} frac12 (189)	¾ ^{3/4} frac34 (190)	numeric character reference	character entity reference			

- Take some time to explore the variety of numeric references and character entity references supported by your browser. Close the demo when you're finished, but leave your browser open.

Special Characters

One use of character codes is to insert text about HTML itself. For example, if you want your Web page to describe the use of the `<h1>` tag, you cannot simply type

```
The <h1> tag is used to mark h1 headings.
```

because the browser will interpret the `<h1>` text as marking the beginning of an h1 heading! Instead, you have to use the `<` and `>` entity references to insert the `<` and `>` symbols. The text would then be

```
The &lt;h1&gt; tag is used to mark h1 headings.
```

Another use of character codes is to add extra spaces to your Web page. Remember that browsers ignore extra blank spaces in the HTML file. To insert additional spaces, use the ` ` entity reference ("`nbsp`" stands for nonbreaking space), which forces the browser to insert extra spaces.

On Dave's Web page, you decide to use the bullet symbol to break up the address text into sections. The symbol has the numeric character code value of 8226 and the entity character reference name "bull." Dave suggests you also add a long horizontal line called an em-dash to mark Thomas Gage's name as the author of the recommendation for Dave's business. The character code and entity reference for an em-dash are 8212 and "mdash," respectively.

Tip

Use the ` ` character entity reference to fine-tune the layout of your documents by adding extra blank spaces to your text.

To add bullets and an em-dash to Dave's Web page:

- ▶ 1. Return to the **dave.htm** file in your text editor.
- ▶ 2. Locate the paragraph within the blockquote element containing the name of Thomas Gage. Directly after the opening `<p>` tag, insert the following numeric character code, followed by a space:


```
&#8212;
```
- ▶ 3. Scroll down to the address element at the bottom of the file. At the end of each line within the address element (except the last line), insert a space and then type the **•** character entity reference. Figure 1-40 highlights the newly added code in the file.

Figure 1-40

Adding character references to the HTML file

numeric
character
reference

```
<blockquote>
<hr />
<p>I'm more than happy to recommend Dave Vinet's products. I came upon his
work 10 years ago and was immediately impressed by his craftsmanship.
I've been using his sticks in my shows ever since. They're durable,
well-balanced, and attractive props and are the perfect complement to
my performances. Thanks Dave!</p>
<p>&#8212; Thomas Gage, lead performer at <i>Circus England</i></p>
<hr />
</blockquote>

<p>Every set is checked and tested before being shipped out to assure perfect
quality. I take pride in every one of my sticks and I want my customers to
feel that same pride.</p>

<h2>My Products</h2>
<ul>
<li><b>Basic Stick</b><br />
The easiest stick to learn with, but "grippy" enough for the most demanding
tricks. Comes in red, green, and blue.</li>
<li><b>Flower Stick</b><br />
A graceful stick with colored tassels. Flower Sticks float more slowly, giving
you the ideal chance to practice new moves.</li>
<li><b>Master Stick</b><br />
My most popular stick is shorter and heavier for fast play and more advanced
tricks. Each Master set is available in custom colors.</li>
<li><b>Glow Stick</b><br />
The Glow Stick shines brightly at night (without the danger of a fire stick).
It combines the fun and versatility of the Master Stick, adding the fun tricks
unique to a glow-in-the-dark stick.</li>
</ul>

<hr />

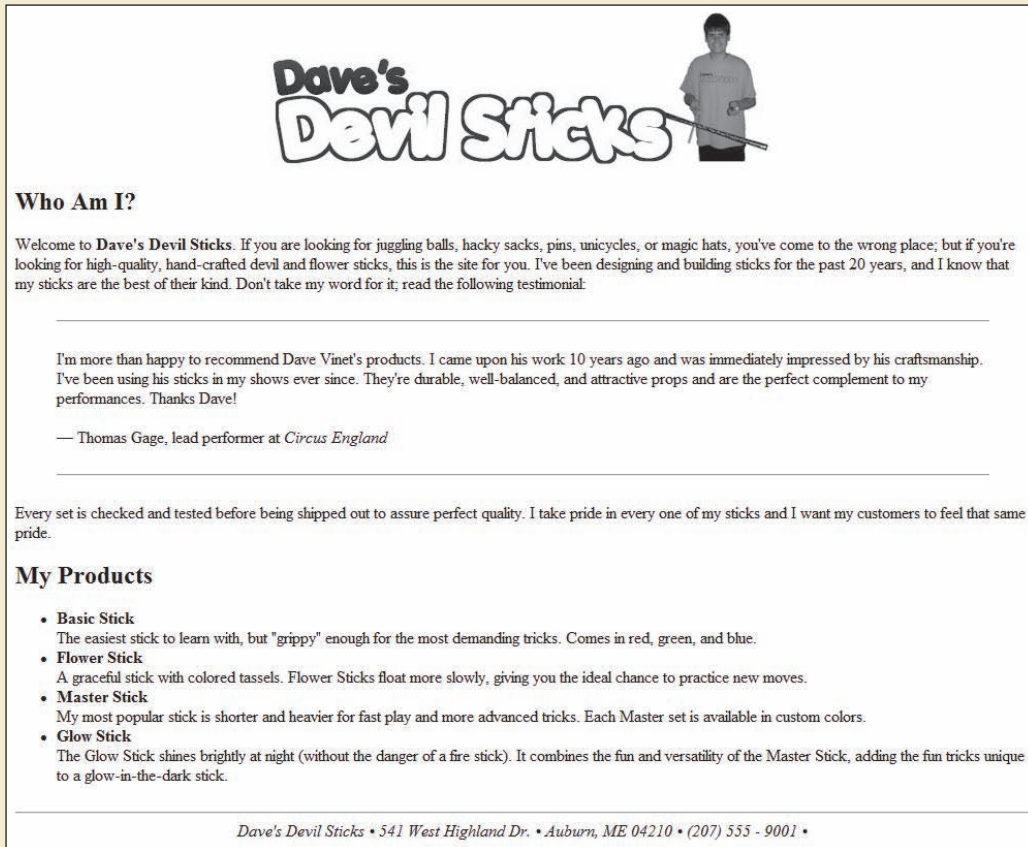
<address style="text-align: center;">Dave's Devil sticks &bull;
541 west Highland Dr. &bull;
Auburn, ME 04210 &bull;
(207) 555 - 9001 &bull;
</address>
</body>
```

character entity
reference

- ▶ 4. Close the file, saving your changes.
- ▶ 5. Refresh the **dave.htm** file in your Web browser. Figure 1-41 shows the final version of Dave's Web page.

Dave's completed Web page

Figure 1-41



Dave's Devil Sticks

Who Am I?

Welcome to **Dave's Devil Sticks**. If you are looking for juggling balls, hacky sacks, pins, unicycles, or magic hats, you've come to the wrong place; but if you're looking for high-quality, hand-crafted devil and flower sticks, this is the site for you. I've been designing and building sticks for the past 20 years, and I know that my sticks are the best of their kind. Don't take my word for it; read the following testimonial:

I'm more than happy to recommend Dave Vinet's products. I came upon his work 10 years ago and was immediately impressed by his craftsmanship. I've been using his sticks in my shows ever since. They're durable, well-balanced, and attractive props and are the perfect complement to my performances. Thanks Dave!

— Thomas Gage, lead performer at *Circus England*

Every set is checked and tested before being shipped out to assure perfect quality. I take pride in every one of my sticks and I want my customers to feel that same pride.

My Products

- **Basic Stick**
The easiest stick to learn with, but "grippy" enough for the most demanding tricks. Comes in red, green, and blue.
- **Flower Stick**
A graceful stick with colored tassels. Flower Sticks float more slowly, giving you the ideal chance to practice new moves.
- **Master Stick**
My most popular stick is shorter and heavier for fast play and more advanced tricks. Each Master set is available in custom colors.
- **Glow Stick**
The Glow Stick shines brightly at night (without the danger of a fire stick). It combines the fun and versatility of the Master Stick, adding the fun tricks unique to a glow-in-the-dark stick.

Dave's Devil Sticks • 541 West Highland Dr. • Auburn, ME 04210 • (207) 555 - 9001 •

- ▶ 6. If you plan on taking a break before working on the end of tutorial problems, you can close your Web browser and any other open files or programs.

You show the completed Web page to Dave. He's pleased that you were able to duplicate much of what was on his original flyer. As you and Dave grow in your understanding of HTML and the Web, you'll add more pages to his site; but for now, this is a good start on giving him a presence on the Internet.

Once you've completed your Web page, your next step is to research ways of getting it on the Web. You first need to find a Web server to host the page. Some of the issues you'll need to consider are how much you want to pay, how much space you need, and how important it is for you to have a highly trafficked Web site. You might first look toward the company that provides your Internet access. Most **ISPs (Internet Service Providers)** offer space on their Web server as part of their regular service or for a small fee. However, they usually limit the amount of space available to you, unless you pay an extra fee to host a larger site. There are also free Web hosts, which provide space on servers for personal or noncommercial use. Once again, the amount of space you get is limited. Free Web hosting services make their money from selling advertising space on your site, so you should be prepared to act as a billboard in return for space on their server.

Web sites are identified by their domain names. If you're planning to create a commercial site to advertise a product or service, you want the domain name to reflect your business. Free Web hosts usually include their names in your Web address. Thus instead of having a Web address like

davesdevilsticks.com

you might have something like

freewebhosting.net/members/~davesdevilsticks.html

If you're running a site for personal use, this might not be a problem—but it would look unprofessional on a commercial site. If you are planning a commercial site and simply want to advertise your product by publishing an online brochure, you can usually find an inexpensive host and pay a nominal yearly fee to reserve a Web address that reflects your company's name. On the other hand, if you intend to run an e-commerce site where users can purchase products online, you will need to invest in software and storage space to manage customer orders and inventory. You will also need to invest in getting your Web site noticed in the increasingly crowded Internet market. Commercial pages require careful planning and good design so that the investment in publishing the site is not wasted.

1. Specify the code you would enter to display the text "Product List" as an h2 heading, centered horizontally on the page.
2. What is a presentational attribute? What is a reason for using presentational attributes? What is a reason for avoiding them?
3. Specify the code you would enter to mark the text "Hamlet by William Shakespeare" as a centered h1 heading with a line break after the word "Hamlet."
4. You want to add the graphic file portrait.gif to your Web page as an inline image. For nonvisual browsers, your page should display the text "David Vinet" in place of the image. Specify the code to do this.
5. What is ISO-8859-1?
6. The trademark symbol [™] has the Unicode number 8482. How would you enter this symbol into your Web page?
7. The Greek letter β has the character entity name of "beta." How would you enter this symbol into your Web page?
8. Specify the code you would enter to add three consecutive blank spaces to your Web page.
9. Specify the code you would enter to display the text "<h2>Hamlet</h2>" on your Web page.

Tutorial Summary

Review

In this tutorial you learned how to create a basic Web page using HTML. The tutorial began by examining concepts and history surrounding networks and the development of the World Wide Web. It then explored the history of HTML, explaining how the development of HTML was a key component in the development of the Web. The first session concluded with the creation of a simple Web page consisting only of the page head. Work on designing the page body began in the second session by first exploring how to mark block-level elements. The discussion of block-level elements included work with headings, paragraphs, block quotes, and lists. The second session concluded with coverage of inline elements and discussed the issue of physical elements versus logical elements. The third session began by exploring element attributes, showing how to use the style attribute to center the contents of a block-level element. Because Web pages often need to display nontextual content, the third session then examined how to use empty elements such as line breaks, horizontal rules, and inline images. The session and the tutorial concluded by discussing character sets and explored how to insert special character symbols into a Web page.

Key Terms

ARPANET	HTML 5.0	root element
ASCII	HTML converter	server
attribute	HTML editor	SGML
block-level element	hypertext	Standard Generalized
body element	Hypertext Markup	Markup Language
character encoding	Language	style
character entity reference	inline element	tag
character formatting	inline image	text-based browser
element	Internet Service Provider	two-sided tag
character set	ISO-8859-1	Unicode
client	ISP	unordered list
client-server network	LAN	UTF-8
closing tag	Latin-1	W3C
comment tag	link	WAN
definition list	local area network	Web
deprecated	logical element	Web browser
element	markup language	Web page
empty element	network	Web server
Extensible Hypertext	node	white space
Markup Language	numeric character	wide area network
Extensible Markup	reference	World Wide Web
Language	one-sided tag	World Wide Web
extensions	opening tag	Consortium
file server	ordered list	XHTML
head element	physical element	XHTML 2.0
heading element	presentational attribute	XHTML 5.0
host	print server	XML
HTML		

Practice | Review Assignments

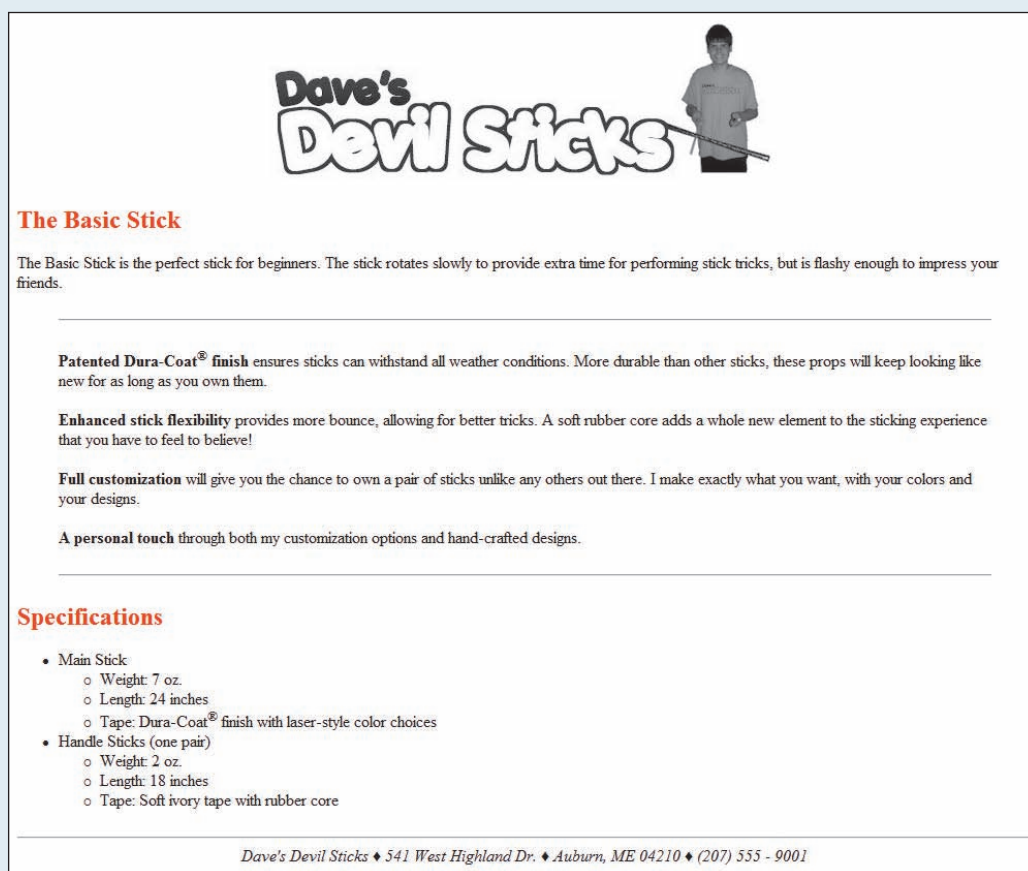
Practice the skills you learned in the tutorial using the same case scenario.

Data File needed for the Review Assignments: logo.jpg

Dave has found a host for his Web page and has published the document you helped him create on the Internet. He wants to start adding more pages to his Web site. He's come to you for help in creating a page describing his basic stick. He's already written the text for the Web page; he needs you to translate that text into HTML code.

Figure 1-42 shows a preview of the page you'll create for Dave.

Figure 1-42



Complete the following:

1. Use your text editor to create a new file named **basic.htm**, and save it in the tutorial.01\review folder included with your Data Files.
2. Within the basic.htm file, insert the structure of the HTML file, including the head and body sections.
3. Within the head section, insert a comment containing

Dave's Devil Sticks
Basic Stick
Author: your name
Date: the date

 where *your name* is your name and *the date* is the current date.

4. Add the page title **Basic Sticks** to the head section.
5. Within the body section, insert an h1 heading centered horizontally on the page and containing the inline image file **logo.jpg**, located in the tutorial.01\review folder included with your Data Files. Specify the following alternate text for the image:
Dave's Devil Sticks: The Basic Stick.
6. Add two h2 headings containing the text **The Basic Stick** and **Specifications**. Set the font color of the heading 2 text to red.
7. Directly below the first h2 heading, insert a paragraph containing the text: **The Basic Stick is the perfect stick for beginners. The stick rotates slowly to provide extra time for performing stick tricks, but is flashy enough to impress your friends.**
8. Directly below the paragraph but above the second h2 heading, insert a block quote that contains the following:
 - a. Place a horizontal rule at the top and the bottom of the block quote.
 - b. Between the two horizontal rules, insert the following four paragraphs:

Patented Dura-Coat finish ensures sticks can withstand all weather conditions. More durable than other sticks, these props will keep looking like new for as long as you own them.

Enhanced stick flexibility provides more bounce, allowing for better tricks. A soft rubber core adds a whole new element to the sticking experience that you have to feel to believe!

Full customization will give you the chance to own a pair of sticks unlike any others out there. I make exactly what you want, with your colors and your designs.

A personal touch through both my customization options and hand-crafted designs.
 - c. Change the first few words of each of the four paragraphs to a bold font, as indicated in Figure 1-42.
9. Directly below the second h2 heading, insert an unordered list. The list should contain two items: **Main Stick** and **Handle Sticks (one pair)**.
10. Directly below the Main Stick list item, insert an unordered list containing the following items:

Weight: 7 oz.

Length: 24 inches

Tape: Dura-Coat finish with laser-style color choices
11. Directly below the Handle Sticks (one pair) list item, insert an unordered list containing:

Weight: 2 oz.

Length: 18 inches

Tape: Soft ivory tape with rubber core
12. Locate the two occurrences of "Dura-Coat" in the document. Directly after the word "Dura-Coat," insert the registered trademark symbol ®. The character entity name of the ® symbol is "reg." Display the ® symbol as a superscript by placing the character within the sup inline element.
13. At the bottom of the body section, insert the company's address:

Dave's Devil Sticks
541 West Highland Dr.
Auburn, ME 04210
(207) 555 - 9001

14. Center the address on the page.
15. Separate the different sections of the address using a solid diamond ♦ (character code 9830).
16. Add a horizontal rule directly above the address element.
17. Save your changes to the file, and then open it in your Web browser to verify that the content and layout are correct.
18. Submit your completed files to your instructor.

Apply | Case Problem 1

Apply your knowledge of HTML to create a Web page for a mathematics department at a university.

Data Files needed for this Case Problem: euler.jpg and eulertext.htm

Mathematics Department, Coastal University Professor Lauren Coe of the Mathematics Department of Coastal University in Anderson, South Carolina, is preparing material for a course on the history of mathematics. As part of the course, she has written biographies of famous mathematicians. Lauren would like you to use content she's already written to create Web pages that students can access on Coastal University's Web server. You'll create the first one in this exercise. Figure 1-43 shows a preview of this page, which profiles the mathematician Leonhard Euler.

Figure 1-43



Leonhard Euler (1707-1783)

The greatest mathematician of the eighteenth century, **Leonhard Euler** was born in Basel, Switzerland. There, he studied under another giant of mathematics, **Jean Bernoulli**. In 1731 Euler became a professor of physics and mathematics at St. Petersburg Academy of Sciences. Euler was the most prolific mathematician of all time, publishing over *800 different books and papers*. His influence was felt in physics and astronomy as well. Euler's work on mathematical analysis, *Introductio in analysin infinitorum* (1748) remained a standard textbook for well over a century. For the princess of Anhalt-Dessau he wrote *Lettres à une princesse d'Allemagne* (1768-1772), giving a clear non-technical outline of the main physical theories of the time.

One can hardly do math without copying Euler. Notations still in use today, such as e and π , were introduced in Euler's writings. He is perhaps best known for his research into mathematical analysis. Euler's formula

$$\cos(x) + i\sin(x) = e^{ix}$$

demonstrates the relationship between algebra, complex analysis, and trigonometry. From this equation, it's easy to derive the equation

$$e^{i\pi} + 1 = 0$$

which relates the fundamental constants: 0, 1, π , e , and i in a single beautiful and elegant statement.

Leonhard Euler died in 1783, leaving behind a legacy perhaps unmatched, and certainly unsurpassed, in the annals of mathematics.

Math 895: The History of Mathematics

Complete the following:

1. Open the **eulertext.htm** file from the tutorial.01\case1 folder included with your Data Files. Save the file as **euler.htm** in the same folder.

2. Add opening and closing `<html>` tags to the file. Insert a head section and enclose Lauren's text on Euler within a body element.
3. Within the head section, insert a comment containing
History of Math 895: Leonhard Euler
Author: your name
Date: the date
 where *your name* is your name and *the date* is the current date.
4. Add the page title **History of Math 895: Leonhard Euler** to the head section.
5. Directly below the opening `<body>` tag, insert a paragraph containing the inline image file **euler.jpg**, located in the tutorial.01\case1 folder included with your Data Files. Specify "Portrait of Leonhard Euler" as the alternative text.
6. Mark the next line containing "Leonhard Euler (1707 - 1783)" as an h1 heading.
7. Mark the five blocks of text describing Euler's life as paragraphs.
8. Mark the two equations as block quotes. Change the font color of the two block quotes to red.
9. Mark the name of the course title at the bottom of the file as an address.
10. Insert horizontal rules directly above the h1 heading and the address element.
11. Within the first paragraph, display the names "Leonhard Euler" and "Jean Bernoulli" in boldface. Mark the phrase "800 different books and papers" as emphasized text using the `em` element. Mark the phrase "Introduction in analysin infinitorum" as a citation.
12. In the phrase, "Lettres a une princesse d'Allemagne" replace the one-letter word a with à (the character entity name is "agrave"). Mark the entire publication as a citation.
13. In the second paragraph, italicize the notation for e and replace "pi" with the character π (the character name is "pi").
14. In the first equation, italicize the letters e, x, and i (but do not italicize the "i" in "sin"). Display the term (ix) as a superscript.
15. In the second equation, replace "pi" with the character π . Italicize the letter e and i. Display (πi) as a superscript.
16. In the last paragraph, italicize the notations for e and i and replace "pi" with π .
17. Save your changes to the file, and then verify that the page appears correctly in your Web browser.
18. Submit your completed files to your instructor.

EXPLORE

Apply

Case Problem 2

Apply your knowledge of HTML to create a page showing text from a scene of a Shakespeare play.

Data Files needed for this Case Problem: macbeth.jpg and macbethtxt.htm

Mansfield Classical Theatre Steve Karls is the director of Mansfield Classical Theatre, a theatre company for young people located in Mansfield, Ohio. This summer the company is planning to perform the Shakespeare play *Macbeth*. Steve wants to put the text of the play on the company's Web site and has asked for your help in designing and completing the Web page. Steve wants to have a separate page for each scene from the play. A preview of the page you'll create for Act I, Scene 1 is shown in Figure 1-44. Steve has already typed the text of the scene. He needs you to supply the HTML code.

Figure 1-44



Macbeth

ACT I

SCENE 1.

Summary A thunderstorm approaches and three witches convene. They agree to confront the great Scot general Macbeth upon his victorious return from a war between Scotland and Norway. Soon, heroic Macbeth will receive the title of Thane of Cawdor from King Duncan. However, Macbeth learns from the witches that he is fated for greater things and he will be led down to the path of destruction by his unquenchable ambition.

A desert place.

Thunder and lightning. Enter three Witches.

First Witch
When shall we three meet again
In thunder, lightning, or in rain?

Second Witch
When the hurlyburly's done,
When the battle's lost and won.

Third Witch
That will be ere the set of sun.

First Witch
Where the place?

Second Witch
Upon the heath.

Third Witch
There to meet with Macbeth.

First Witch
I come, Graymalkin!

Second Witch
Paddock calls.

Third Witch
Anon.

ALL
Fair is foul, and foul is fair:
Hover through the fog and filthy air.

Exeunt

[Go to Scene 2 ⇒](#)

Text provided by Online Shakespeare

Complete the following:

1. Open the **macbeth.txt.htm** file from the tutorial.01\case2 folder included with your Data Files. Save the file as **macbeth.htm** in the same folder.
2. Enclose the entire Macbeth text within the structure of an HTML document.
3. Within the head section, insert a comment containing the following text:
Macbeth: Act I, Scene 1
Author: your name
Date: the date
4. Add the page title **Macbeth: Act I, Scene 1** to the head section.
5. Directly below the opening <body> tag, insert an h1 heading containing the inline image file **macbeth.jpg** (located in the tutorial.01\case2 folder included with your Data Files) with **Macbeth** as the alternate text for nonvisual browsers. Add a horizontal rule directly below the h1 heading.
6. Mark the text "ACT I" as an h2 heading. Mark "SCENE 1." as an h3 heading.

7. Mark the summary of the scene as a paragraph. Display the word “Summary” in bold.
8. In the text of the play, mark the descriptions of setting, scene, and exits as separate paragraphs, and italicize the text, as shown in Figure 1-44.
9. Mark the dialog as a definition list, with each character’s name marked as a definition term and each speech marked as a definition description. Where the speech goes over one line, use a line break to keep the speech on separate lines, as shown in the figure.
10. Directly below the Exeunt paragraph, insert the line **Go to Scene 2**. Mark this line as a div element and align it with the right page margin. (Steve will mark this as a link later.) At the end of the line, insert a right arrow character using the 8658 character number. Add horizontal rules directly above and below this statement.
11. Mark the line “Text provided by Online Shakespeare” as a paragraph, with the text itself marked with the cite element. Align the text with the right page margin.
12. Save your changes to the file, and then confirm the layout and content of the page in your Web browser.
13. Submit the completed files to your instructor.

 **EXPLORE**

 **EXPLORE**

Challenge


| Case Problem 3

Explore how to use HTML to create a recipe page.


Data Files needed for this Case Problem: [dessertweb.jpg](#), [torte.jpg](#), and [tortetxt.htm](#)

dessertWEB Amy Wu wants to take her love of cooking and sharing recipes to the World Wide Web. She’s interested in creating a new Web site called *dessertWEB* where other cooks can submit and review dessert recipes. Each page within her site will have a photo and description of the dessert, the ingredients, the cooking directions, and a list of reviews. Each recipe will be rated on a 5-star scale. She already has information on one recipe: Apple Bavarian Torte. She’s asked for your help in creating a Web page from the data she’s collected. A preview of the completed page is shown in Figure 1-45.

Figure 1-45



Apple Bavarian Torte (★★★★)



A classic European torte baked in a springform pan. Cream cheese, sliced almonds, and apples make this the perfect holiday treat (12 servings).

INGREDIENTS

- 1/2 cup butter
- 1/3 cup white sugar
- 1/4 teaspoon vanilla extract
- 1 cup all-purpose flour
- 1 (8 ounce) package cream cheese
- 1/4 cup white sugar
- 1 egg
- 1/2 teaspoon vanilla extract
- 6 apples - peeled, cored, and sliced
- 1/3 cup white sugar
- 1/2 teaspoon ground cinnamon
- 1/4 cup sliced almonds

DIRECTIONS

1. Preheat oven to 450° F (230° C).
2. Cream together butter, sugar, vanilla, and flour.
3. Press crust mixture into the flat bottom of a 9-inch springform pan. Set aside.
4. In a medium bowl, blend cream cheese and sugar. Beat in egg and vanilla. Pour cheese mixture over crust.
5. Toss apples with sugar and cinnamon. Spread apple mixture over all.
6. Bake for 10 minutes. Reduce heat to 400° F (200° C) and continue baking for 25 minutes.
7. Sprinkle almonds over top of torte. Continue baking until lightly browned. Cool before removing from pan.

REVIEWS

★★★★★
I loved the buttery taste of the crust which complements the apples very nicely.
— Reviewed on Sep. 22, 2010 by MMASON.

★★★☆☆
Nothing special. I like the crust, but there was a little too much of it for my taste, and I liked the filling but there was too little of it. I thought the crunchy apples combined with the sliced almonds detracted from the overall flavor.
— Reviewed on Sep. 1, 2010 by GLENDACHEF.

★★★★★
Delicious!! I recommend microwaving the apples for 3 minutes before baking, to soften them. Great dessert - I'll be making it again for the holidays.
— Reviewed on August 28, 2010 by BBABS.

Complete the following:

1. Open the **tortetxt.htm** file from the tutorial.01\case3 folder included with your Data Files. Save the file as **torte.htm** in the same folder.
2. Add the structure of an HTML document around the recipe text. Within the head section, insert a comment containing the following text:
Apple Bavarian Torte
Author: your name
Date: the date
3. Add the page title **Apple Bavarian Torte Recipe** to the head section.
4. Directly below the opening <body> tag, insert a div element containing the inline image **dessertweb.jpg** located in the tutorial.01\case3 folder included with your Data Files. Specify the alternative text **dessertWEB**. Insert a horizontal rule directly below the div element.

 EXPLORE

5. Mark the text “Apple Bavarian Torte” as an h2 heading.
6. Change the text “(4 stars)” to a set of 4 star symbols (character number 9733). Enclose the star symbols in a span element, setting the font color to teal.
7. Directly below the h2 heading, insert another div element containing the inline image **torte.jpg**, located in the tutorial.01\case3 folder included with your Data Files. Specify the alternative text **Torte image**.
8. Mark the description of the dessert as a paragraph.
9. Mark “INGREDIENTS,” “DIRECTIONS,” and “REVIEWS” as h3 headings.
10. Enclose the list of ingredients in a block quote. Add line breaks after each item in the list.

 EXPLORE

11. Mark the list of directions as an ordered list, with each direction a separate item in the list. Replace the word “degrees” with the degree symbol (character name deg).
12. Enclose the list of reviews in a block quote. Turn the list into a definition list. Mark up the definition list as follows:

 EXPLORE

- a. The definition term is the number of stars assigned by each reviewer. Change the number of stars in the text file to star symbols (character number 9733). Amy wants you to display 5 stars for each review with the number of stars displayed in a teal font matching the stars given by the reviewer and the remaining stars displayed in a gray font. Use the span element to enclose the two different groups of stars.
 - b. There are two definition descriptions for each review. The first encloses the text of the review. The second encloses the date of the review and the name of the reviewer.
 - c. Insert an em-dash (character name “mdash”) before the word “Reviewed” in each of the reviews.
13. Save your changes to the file, and then verify the layout and content of the page in your Web browser.
 14. Submit the completed files to your instructor.

Create

| **Case Problem 4**

Test your knowledge of HTML and use your creativity to design a Web page for an exercise equipment company.

Data Files needed for this Case Problem: logo.jpg, smith.jpg, and smith.txt

Body Systems Body Systems is a leading manufacturer of home gyms. The company recently hired you to assist in developing its Web site. Your first task is to create a Web page for the LSM400, a popular weight machine sold by the company. You’ve been given a text file describing the features of the LSM400. You’ve also received two image files: one of the company’s logo and one of the LSM400. You are free to supplement these files with any other resources available to you. You are responsible for the page’s content and appearance.

Complete the following:

1. Create a new HTML file named **smith.htm** and save it in the tutorial.01\case4 folder included with your Data Files.
2. Add a comment to the head section of the document describing the document’s content and containing your name and the date.
3. Add an appropriate page title to the document.

4. Use the contents of the **smith.txt** document (located in the tutorial.01\case4 folder) as the basis of the text in the Web page. Include at least one example of each of the following:
 - a heading
 - a paragraph
 - an ordered or unordered list
 - an inline element
 - an inline image
 - a horizontal rule
 - a special character
 - an element attribute
5. Structure your HTML code so that it will be easy for others to read and understand.
6. Save your changes to the file, and then open it in your Web browser to verify that it is readable and attractive.
7. Submit your completed files to your instructor.

Review

| **Quick Check Answers****Session 1.1**

1. A hypertext document is an electronic file containing elements that users can select, usually by clicking a mouse, to open other documents.
2. A Web server is a computer on a network that stores a Web site and makes it available to clients. Users access the Web site by running a program called a Web browser on their computers.
3. HTML documents do not exactly specify the appearance of a document; rather they describe the purpose of various elements in the document and leave it to the Web browser to determine the final appearance. A word processor like Microsoft Word exactly specifies the appearance of each document element.
4. Deprecated features are those features that are being phased out by the W3C and might not be supported by future browsers.
5. The `html` element.
6. In the head section of the document you would enter the code
`<title>Technical Support</title>`
7. `<!-- Page Updated on 4/15/2011 -->`
8. The title element was not properly nested within the head element.

Session 1.2

1. Block-level elements contain content that is displayed in a separate section within the page, such as a paragraph or a heading. An inline element is part of the same block as its surrounding content—for example, individual words or phrases within a paragraph.
2. Web browsers will strip out extra occurrences of white space and thus will ignore the extra blank line.
3. `<h1>content</h1>`
4. Use the blockquote element. Most browsers indent blockquote text.

5.

```
<ul>
  <li>Winter</li>
  <li>Spring</li>
  <li>Summer</li>
  <li>Fall</li>
</ul>
```
6.

```
<dl>
  <dt>HAMLET</dt>
  <dd> There's ne'er a villain dwelling in all Denmark; but he's
  an arrant knave.</dd>
  <dt>HORATIO</dt>
  <dd> There needs no ghost, my lord, come from the grave to tell
  us this.</dd>
</dl>
```
7.

```
<p><i>Hamlet</i>, a play by William Shakespeare</p>
```
8. `div` and `span`

Session 1.3

1.

```
<h2 style="text-align: center">Product List</h2>
```
2. Presentational attributes are HTML attributes that exactly specify how the browser should render an HTML element. Most presentational attributes have been deprecated, replaced by styles. You should use presentational attributes when you need to support older browsers.
3.

```
<h1 style="text-align: center">Hamlet<br /> by William
Shakespeare</h1>
```
4.

```

```
5. ISO-8859-1 is a character set that supports 255 characters and can be used by most languages that use the Latin alphabet.
6. `™`
7. `β`
8. ` `
9. `<h2>Hamlet</h2>`

Ending Data Files

