

SHELLY CASHMAN SERIES®

HTML, XHTML, AND CSS

Sixth Edition

COMPLETE



SHELLY | WOODS

This is an electronic version of the print textbook. Due to electronic rights restrictions, some third party content may be suppressed.

Editorial review has deemed that any suppressed content does not materially affect the overall learning experience.

The publisher reserves the right to remove content from this title at any time if subsequent rights restrictions require it.

For valuable information on pricing, previous editions, changes to current editions, and alternate formats,
please visit www.cengage.com/highered to search by ISBN#, author, title, or keyword for materials in your areas of interest.

HTML, XHTML, and CSS

**Complete
Sixth Edition**



**Gary B. Shelly
Denise M. Woods**

 COURSE TECHNOLOGY
CENGAGE Learning™

Australia • Brazil • Japan • Korea • Mexico • Singapore • Spain • United Kingdom • United States

HTML, XHTML, and CSS**Complete, Sixth Edition**

Gary B. Shelly, Denise M. Woods

Vice President, Publisher: Nicole Pinard

Executive Editor: Kathleen McMahon

Product Manager: Crystal Parenteau

Associate Product Manager: Aimee Poirier

Editorial Assistant: Lauren Brody

Director of Marketing: Cheryl Costantini

Marketing Manager: Tristen Kendall

Marketing Coordinator: Stacey Leasca

Print Buyer: Julio Esperas

Director of Production: Patty Stephan

Content Project Manager: Jennifer Feltri

Development Editor: Deb Kaufmann

Copyeditor: Camille Kiolbasa

Proofreader: Christine Clark

Indexer: Ken Hassman

QA Manuscript Reviewers: John Freitas,
Serge Palladino, and Susan Whalen

Art Director: Marissa Falco

Cover Designer: Lisa Kuhn, Curio Press, LLC

Cover Photo: Tom Kates Photography

Composer: PreMediaGlobal

© 2011 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

Library of Congress Control Number: 2010928175

ISBN-13: 978-0-538-74745-5

ISBN-10: 0-538-74745-5

Course Technology

20 Channel Center Street

Boston, MA 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.

For your course and learning solutions, visit www.cengage.com

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

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

HTML, XHTML, and CSS

Complete
Sixth Edition

Contents

Preface

viii

HTML, XHTML, and CSS

CHAPTER ONE

Introduction to HTML, XHTML, and CSS

Objectives	HTML 1
Introduction	HTML 2
What Is the Internet?	HTML 2
What Is the World Wide Web?	HTML 3
Web Servers	HTML 4
Web Site Types and Purposes	HTML 4
Web Browsers	HTML 7
What Is Hypertext Markup Language?	HTML 8
HTML Elements	HTML 9
HTML Coding Practices	HTML 10
HTML Versions	HTML 11
Cascading Style Sheets	HTML 11
Document Object Model (DOM)	HTML 11
Extensible Hypertext Markup Language (XHTML)	HTML 12
Tools for Creating HTML Documents	HTML 13
Web Development Life Cycle	HTML 14
Web Site Planning	HTML 15
Web Site Analysis	HTML 16
Web Site Design and Development	HTML 16
Web Site Testing	HTML 22
Web Site Implementation and Maintenance	HTML 24
Be an Observant Web User	HTML 24
Chapter Summary	HTML 25
Learn It Online	HTML 25
Apply Your Knowledge	HTML 26
Extend Your Knowledge	HTML 27
Make It Right	HTML 27
In the Lab	HTML 28
Cases and Places	HTML 31

CHAPTER TWO

Creating and Editing a Web Page Using Inline Styles

Objectives	HTML 33
Introduction	HTML 34

Project — Fight Against World Hunger

Web Page	HTML 34
Overview	HTML 34
Elements of a Web Page	HTML 37
Window Elements	HTML 37
Text Elements	HTML 38
Image Elements	HTML 38
Hyperlink Elements	HTML 38
Defining Web Page Structure	HTML 39
Defining the HTML Document	HTML 39
To Start Notepad++	HTML 40
To Enable Word Wrap in Notepad++	HTML 42
To Define the Web Page Structure Using HTML Tags	HTML 42
Entering Web Page Content	HTML 45
To Enter a Heading	HTML 47
To Enter a Paragraph of Text	HTML 47
Using Lists to Present Content	HTML 48
To Create an Unordered List	HTML 50
More About List Formats	HTML 51
Saving and Organizing HTML Files	HTML 53
To Save an HTML File	HTML 54
Using a Browser to View a Web Page	HTML 57
To Start a Browser	HTML 57
To View a Web Page in a Browser	HTML 59
To Activate Notepad++	HTML 61
Improving the Appearance of Your Web Page	HTML 61
Using Style Sheets	HTML 62
Web Page Images	HTML 65
Using Web Page Divisions	HTML 65
Image Types	HTML 65
Image Attributes	HTML 66
To Add an Image	HTML 66
Other Visual Enhancements	HTML 68
To Add Color to Web Page Headings	HTML 70
To Change the Bulleted List Style	HTML 71
To Add a Horizontal Rule	HTML 72
To Refresh the View in a Browser	HTML 73
Validating and Viewing HTML Code	HTML 73
To Validate HTML Code	HTML 75
To View HTML Source Code for a Web Page	HTML 77

To Print a Web Page and an HTML File	HTML 78	To Validate HTML Code	HTML 122																																																																																																																																		
To Quit Notepad++ and a Browser	HTML 80	To Print an HTML File	HTML 123																																																																																																																																		
Chapter Summary	HTML 80	To View a Web Page	HTML 124																																																																																																																																		
Learn It Online	HTML 80	To Test Links on a Web Page	HTML 125																																																																																																																																		
Apply Your Knowledge	HTML 81	To Print a Web Page	HTML 127																																																																																																																																		
Extend Your Knowledge	HTML 82	Editing the Second Web Page	HTML 127																																																																																																																																		
Make It Right	HTML 83	To Open an HTML File	HTML 129																																																																																																																																		
In the Lab	HTML 84	Working with Classes in Style Statements	HTML 130																																																																																																																																		
Cases and Places	HTML 87	Adding an Image with Wrapped Text	HTML 132																																																																																																																																		
CHAPTER THREE																																																																																																																																					
Creating Web Pages with Links, Images, and Embedded Style Sheets																																																																																																																																					
Objectives	HTML 89	Using Thumbnail Images	HTML 135																																																																																																																																		
Introduction	HTML 90	Wrap Text Around Images Using																																																																																																																																			
Project — Getting Greener Web Site	HTML 90	CSS Classes	HTML 136																																																																																																																																		
Overview	HTML 91	To Clear Text Wrapping	HTML 138																																																																																																																																		
Using Links on a Web Page	HTML 93	Adding Links Within a Web Page	HTML 139																																																																																																																																		
Linking to Another Web Page Within the Same Web Site	HTML 95	To Set Link Targets	HTML 140																																																																																																																																		
Linking to a Web Page in Another Web Site	HTML 96	To Add Links to Link Targets Within a Web Page	HTML 141																																																																																																																																		
Linking Within a Web Page	HTML 98	To Add Links to a Target at the Top of the Page	HTML 142																																																																																																																																		
Linking to an E-Mail Address	HTML 98	To Copy and Paste HTML Code	HTML 143																																																																																																																																		
Creating a Home Page	HTML 99	To Add an Image Link to a Web Page	HTML 143																																																																																																																																		
To Start Notepad++	HTML 100	To Save, Validate, and Print the HTML File	HTML 144																																																																																																																																		
To Enter Initial HTML Tags to Define the Web Page Structure	HTML 100	To View and Test a Web Page	HTML 145																																																																																																																																		
To Save an HTML File	HTML 101	To Print a Web Page	HTML 145																																																																																																																																		
To Add a Center-Aligned Banner Image Using an Inline Style	HTML 102	To Quit Notepad++ and a Browser	HTML 146																																																																																																																																		
To Add Paragraphs of Text	HTML 104	Chapter Summary	HTML 146																																																																																																																																		
Using an Inline Style to Add Interest to a Bulleted List	HTML 105	Learn It Online	HTML 146																																																																																																																																		
To Create an Unordered (Bulleted) List Using Images as Bullets	HTML 106	Apply Your Knowledge	HTML 147																																																																																																																																		
Adding a Text Link to Another Web Page Within the Same Web Site	HTML 107	Extend Your Knowledge	HTML 148																																																																																																																																		
To Add a Text Link to Another Web Page Within the Same Web Site	HTML 108	Make It Right	HTML 149																																																																																																																																		
Adding an E-Mail Link	HTML 109	In the Lab	HTML 150																																																																																																																																		
To Add an E-Mail Link	HTML 109	Cases and Places	HTML 155																																																																																																																																		
Adding Other Information to an E-Mail Link	HTML 109	CHAPTER FOUR																																																																																																																																			
To Add a Subject to an E-Mail Link	HTML 110	Creating Tables in a Web Site Using an External Style Sheet																																																																																																																																			
To Add a Subject Together with Body Message Text	HTML 110					Objectives	HTML 157	To Add a Text Link to a Web Page in Another Web Site	HTML 111	Introduction				HTML 158	Using Absolute and Relative Paths	HTML 111	Project — Sabatina's Pizza Web Site				HTML 158	Adding Interest and Focus with Styles	HTML 114	Overview				HTML 160	Adding Interest and Focus with HTML Tags	HTML 115	Planning and Designing a Multipage Web Site				HTML 161	Style Sheet Precedence Review	HTML 116	Creating Web Pages with Tables				HTML 161	Using Embedded Style Sheets	HTML 117	Table Elements				HTML 163	To Add Embedded Style Sheet Statements	HTML 119	Adding Style to Table Elements				HTML 164	To Add an Inline Style for Color	HTML 121	Table Borders, Headers, Captions, and Rules				HTML 165	To Save an HTML File	HTML 121	Determining the Need for, Planning, and					Validating the HTML, Viewing the Web Page, and Testing Links	HTML 121	Coding a Table				HTML 166			Determining If a Table Is Needed				HTML 166			Planning the Table				HTML 167			Coding the Table				HTML 168			Table Tag Attributes				HTML 170			Creating a Home Page with Banner Logo and Borderless Table				HTML 171			To Start Notepad++				HTML 171			To Enter Initial HTML Tags to Define the Web Page Structure				HTML 171			To Save an HTML File				HTML 172
				Objectives	HTML 157																																																																																																																																
To Add a Text Link to a Web Page in Another Web Site	HTML 111	Introduction				HTML 158																																																																																																																															
Using Absolute and Relative Paths	HTML 111	Project — Sabatina's Pizza Web Site				HTML 158																																																																																																																															
Adding Interest and Focus with Styles	HTML 114	Overview				HTML 160																																																																																																																															
Adding Interest and Focus with HTML Tags	HTML 115	Planning and Designing a Multipage Web Site				HTML 161																																																																																																																															
Style Sheet Precedence Review	HTML 116	Creating Web Pages with Tables				HTML 161																																																																																																																															
Using Embedded Style Sheets	HTML 117	Table Elements				HTML 163																																																																																																																															
To Add Embedded Style Sheet Statements	HTML 119	Adding Style to Table Elements				HTML 164																																																																																																																															
To Add an Inline Style for Color	HTML 121	Table Borders, Headers, Captions, and Rules				HTML 165																																																																																																																															
To Save an HTML File	HTML 121	Determining the Need for, Planning, and																																																																																																																																			
Validating the HTML, Viewing the Web Page, and Testing Links	HTML 121	Coding a Table				HTML 166																																																																																																																															
		Determining If a Table Is Needed				HTML 166																																																																																																																															
		Planning the Table				HTML 167																																																																																																																															
		Coding the Table				HTML 168																																																																																																																															
		Table Tag Attributes				HTML 170																																																																																																																															
		Creating a Home Page with Banner Logo and Borderless Table				HTML 171																																																																																																																															
		To Start Notepad++				HTML 171																																																																																																																															
		To Enter Initial HTML Tags to Define the Web Page Structure				HTML 171																																																																																																																															
		To Save an HTML File				HTML 172																																																																																																																															

To Insert and Center an Image	HTML 173	Adding Keywords	HTML 224
Using a Table to Create a Horizontal Menu Bar	HTML 174	To Open the File	HTML 224
To Create a Horizontal Menu Bar with Text Links	HTML 175	Meta Names	HTML 225
To Add Paragraphs of Text	HTML 176	To Add Keywords	HTML 226
To Save the HTML File	HTML 176	To Add a Description	HTML 226
Viewing the Web Page and Testing Links	HTML 177	To Save, Validate, and Print a Document	HTML 227
To View a Web Page	HTML 177	Determining a Domain Name	HTML 228
To Test Links on a Web Page	HTML 178	Check Domain Name Availability	HTML 228
External Style Sheets	HTML 178	Finding a Web Hosting Site	HTML 229
Adding an External Style Sheet	HTML 180	Publishing Your Web Site	HTML 229
To Create an External Style Sheet	HTML 181	Advertising Your Web Site	HTML 230
Examining the External Style Sheet	HTML 183	Registering with Search Engines	HTML 230
Linking to the External Style Sheet	HTML 186	To Register Your Web Site with Search Engines	HTML 231
To Link to an External Style Sheet	HTML 186	Feature Summary	HTML 231
Validating and Printing the HTML, Viewing the Web Page, and Testing Links	HTML 187	In the Lab	HTML 232
To Validate a Web Page	HTML 187		
To Print an HTML File	HTML 188		
To View Test, and Print a Web Page	HTML 189		
Creating a Second Web Page	HTML 189		
To Copy and Paste HTML Code to a New File	HTML 190	CHAPTER FIVE	
To Save an HTML File	HTML 191	Creating an Image Map	
Adding a Table with Borders	HTML 192	Objectives	HTML 233
To Create a Table with Borders and Insert Text	HTML 193	Introduction	HTML 234
To Save, Validate, View, and Print the Web Page	HTML 194	Project — Bright Idea®	HTML 234
To Test Links on a Web Page	HTML 196	Overview	HTML 236
Adding a Link, Cellspacing, and Cellpadding	HTML 197	Introduction to Image Maps	HTML 237
To Open an HTML File	HTML 198	Using Image Maps	HTML 237
To Link to an External Style Sheet	HTML 199	Server-Side vs. Client-Side Image Maps	HTML 242
To Add Cellspacing and Cellpadding to a Table	HTML 199	Creating an Image Map	HTML 243
To Save, Validate, Print, and View the HTML File and Print the Web Page	HTML 200	Selecting Images	HTML 243
Adding a Caption and Spanning Rows	HTML 200	Sketching the Borders of Hotspots	HTML 244
To Open an HTML File	HTML 202	Mapping Image Coordinates	HTML 246
To Link the External Style Sheet	HTML 203	Coding the Map	HTML 248
To Add a Table Caption	HTML 203	Using Paint to Locate X- and Y-Coordinates	HTML 249
To Create the Headings That Span Rows	HTML 204	To Start Paint	HTML 249
To Save, Validate, and Print the HTML File	HTML 205	The Paint Window	HTML 250
To View and Print the Web Page	HTML 206	To Open an Image File in Paint	HTML 251
To Quit Notepad++ and a Browser	HTML 206	Locating X- and Y-Coordinates of an Image	HTML 251
Chapter Summary	HTML 207	To Locate X- and Y-Coordinates of an Image	HTML 252
Learn It Online	HTML 207	Other Software Tools	HTML 254
Apply Your Knowledge	HTML 208	Creating the Home Page	HTML 255
Extend Your Knowledge	HTML 210	Creating a Table	HTML 257
Make It Right	HTML 212	Inserting Special Characters	HTML 259
In the Lab	HTML 213	To Insert a Special Character	HTML 260
Cases and Places	HTML 218	Inserting an Image to Use as an Image Map	HTML 260
SPECIAL FEATURE 1		To Insert an Image to Use as an Image Map	HTML 261
Attracting Visitors to Your Web Site		Coding the Image Map Using HTML Tags and Attributes	HTML 265
Objectives	HTML 221	To Create an Image Map	HTML 266
Introduction	HTML 222	Viewing the Web Page and Testing Links	HTML 267
Project — Attracting Visitors	HTML 222	Creating an External Style Sheet	HTML 270
Overview	HTML 222	To Create an External Style Sheet	HTML 271
Chapter Summary		Learn It Online	HTML 274
Learn It Online		Apply Your Knowledge	HTML 275
Extend Your Knowledge		Extend Your Knowledge	HTML 276

Make It Right	HTML 278	
In the Lab	HTML 279	
Cases and Places	HTML 283	
CHAPTER SIX		
Creating a Form on a Web Page		
Objectives	HTML 285	
Introduction	HTML 286	
Project — Creating Forms on a Web Page	HTML 286	
Overview	HTML 288	
Web Page Forms	HTML 289	
Input Controls	HTML 289	
HTML Tags Used to Create Forms	HTML 293	
Attributes of HTML Tags Used to Create Forms	HTML 293	
Creating a Form on a Web Page	HTML 294	
Creating a Form and Identifying the Form Process	HTML 296	
To Create a Form and Identify the Form Process	HTML 296	
To Change the Text Message	HTML 297	
Adding Text Boxes	HTML 298	
To Add Text Boxes	HTML 298	
Adding Radio Buttons	HTML 299	
To Add Radio Buttons	HTML 300	
Adding Check Boxes	HTML 300	
To Add Check Boxes	HTML 301	
Adding a Selection Menu	HTML 301	
To Add a Selection Menu	HTML 302	
Adding More Advanced Selection Menus	HTML 303	
Adding a Textarea Box	HTML 305	
To Add Textarea Box	HTML 305	
Adding Submit and Reset Buttons	HTML 306	
To Add Submit and Reset Buttons	HTML 307	
Organizing a Form Using Form Groupings	HTML 308	
To Add Fieldset Controls to Create Form Groupings	HTML 309	
Adding an Embedded Style Sheet	HTML 310	
Chapter Summary	HTML 316	
Learn It Online	HTML 316	
Apply Your Knowledge	HTML 317	
Extend Your Knowledge	HTML 318	
Make It Right	HTML 319	
In the Lab	HTML 320	
Cases and Places	HTML 323	
CHAPTER SEVEN		
Using Advanced Cascading Style Sheets		
Objectives	HTML 325	
Introduction	HTML 326	
Project — Using Advanced Cascading Style Sheets	HTML 326	
Overview	HTML 328	
Using Style Sheets	HTML 329	
Style Sheet Precedence Review	HTML 332	
Adding Style Sheets to the Sapperzein Galleries Site	HTML 332	
Adding a Navigation Menu with an Embedded Style Sheet		HTML 334
Setting the Body Style and Link Style, and Adding a Drop-down Menu		HTML 335
To Add an Embedded Style Sheet		HTML 340
Adding Pop-ups with an External Style Sheet	HTML 343	
Structuring the Web Page		HTML 343
Creating a Pop-up Using Cascading Style Sheets		HTML 344
To Create and Print an External Style Sheet		HTML 345
Linking to an External Style Sheet		HTML 346
To Link to an External Style Sheet		HTML 346
Creating an External Style Sheet for Printing		HTML 348
Understanding the Page Box Model		HTML 348
To Create and Print an External Style Sheet for Printing		HTML 348
Chapter Summary		HTML 351
Learn It Online		HTML 351
Apply Your Knowledge		HTML 352
Extend Your Knowledge		HTML 353
Make It Right		HTML 354
In the Lab		HTML 355
Cases and Places		HTML 358
CHAPTER EIGHT		
Adding Multimedia Content to Web Pages		
Objectives		HTML 361
Introduction		HTML 362
Project — Adding Multimedia to a History Class Web Site		HTML 362
Overview		HTML 362
Using Multimedia		HTML 364
Creating or Finding Multimedia Files		HTML 366
Embedded vs. External Multimedia		HTML 366
Media Players and Plug-Ins		HTML 367
Audio and Video File Formats		HTML 368
Adding an Audio File to a Web Page		HTML 369
The Object Tag		HTML 369
Object Tag Parameters		HTML 372
To Add an Audio Clip to a Web Page		HTML 375
To Validate and View a Web Page Using ActiveX Controls		HTML 376
Adding a Video Clip to a Web Page		HTML 379
To Add a Video Clip to a Web Page		HTML 382
Chapter Summary		HTML 384
Learn It Online		HTML 385
Apply Your Knowledge		HTML 385
Extend Your Knowledge		HTML 386
Make It Right		HTML 387
In the Lab		HTML 388
Cases and Places		HTML 391
SPECIAL FEATURE 2		
Converting Frames on Your Web Site		
Objectives		HTML 393
Introduction		HTML 394

Project — Converting Frames	HTML 394	Section 508 Guidelines Examples	APP 13
Overview	HTML 394	WAI Guidelines	APP 16
The Frame Definition File	HTML 396	APPENDIX D	
To View a Web Site with Frames	HTML 398	CSS Properties and Values	
Frame Layouts	HTML 399	CSS Concepts and Terminology	APP 21
Determining a Conversion Strategy	HTML 401	CSS Properties	APP 23
To Copy HTML Code for a Menu Bar	HTML 402	Acceptable Units of Measure	APP 23
To Paste Code into the Home Page		Background and Color Styles	APP 24
and Edit the Code		Border Styles	APP 25
To Copy and Paste Code		Classification Styles	APP 26
into Other Web Pages	HTML 403	Font Styles	APP 26
Feature Summary	HTML 405	List Styles	APP 27
In the Lab	HTML 406	Margin and Padding Styles	APP 27
		Miscellaneous Formatting Styles	APP 28
		Text Styles	APP 28

Appendices

APPENDIX A	APPENDIX E		
HTML Quick Reference	Publishing Web Pages		
HTML and XHTML Coding Standards	APP 1	to a Web Server	
HTML Tags and Attributes	APP 2	Choosing a Web Host	APP 31
		Uploading Files to the Host	APP 31
APPENDIX B	APPENDIX F		
Browser-Safe Color Palette	Symbols and Characters		
Browser-Safe Colors	APP 11	Quick Reference	
		Using Symbols and Special Characters	APP 33
APPENDIX C	Index		
Accessibility Standards and the Web	IND 1		
Making the Web Accessible	APP 13		



Preface

The Shelly Cashman Series® offers the finest textbooks in computer education. We are proud of the fact that our previous HTML books have been so well received. With each new edition of our HTML books, we have made significant improvements based on the comments made by instructors and students. The *HTML, XHTML, and CSS, Sixth Edition* books continue with the innovation, quality, and reliability you have come to expect from the Shelly Cashman Series.

In 2006 and 2007, the Shelly Cashman Series development team carefully reviewed our pedagogy and analyzed its effectiveness in teaching today's student. An extensive customer survey produced results confirming what the series is best known for: its step-by-step, screen-by-screen instructions, its project-oriented approach, and the quality of its content.

We learned, though, that students entering computer courses today are different than students taking these classes just a few years ago. Students today read less, but need to retain more. They need not only to be able to perform skills, but to retain those skills and know how to apply them to different settings. Today's students need to be continually engaged and challenged to retain what they're learning.

As a result, we've renewed our commitment to focusing on the user and how they learn best. This commitment is reflected in every change we've made to our HTML book.

Objectives of This Textbook

HTML, XHTML, and CSS: Complete, Sixth Edition is intended for use in combination with other books in an introductory course on creating Web pages, or as a stand-alone in a two-credit hour course or a continuing education course. This book also is suitable for use as a stand alone in a one-credit hour course or a continuing education course. No experience with Web page development or computer programming is required. Specific objectives of this book are as follows:

- To teach the fundamentals of developing Web pages using a comprehensive Web development life cycle
- To acquaint students with the HTML and CSS languages and creating Web pages suitable for course work, professional purposes, and personal use
- To expose students to common Web page formats and functions
- To promote curiosity and independent exploration of World Wide Web resources
- To develop an exercise-oriented approach that allows students to learn by example
- To encourage independent study and help those who are learning how to create Web pages in a distance education environment

Distinguishing Features

A Proven Pedagogy with an Emphasis on Project Planning Each chapter presents a practical problem to be solved, within a project planning framework. The project orientation is strengthened by the use of Plan Ahead boxes that encourage critical thinking about how to proceed at various points in the project. Step-by-step instructions with supporting screens guide students through the steps. Instructional steps are supported by the Q&A, Experimental Step, and BTW features.

A Visually Engaging Book that Maintains Student Interest The step-by-step tasks, with supporting figures, provide a rich visual experience for the student. Call-outs on the screens that present both explanatory and navigational information provide students with information they need when they need to know it.

Supporting Reference Materials (Appendices) The appendices provide additional information about HTML, XHTML, and CSS topics, with appendices such as the HTML Quick Reference, CSS Properties and Values, and Publishing Web Pages to a Web Server.

Integration of the World Wide Web The World Wide Web is integrated into the HTML, XHTML, and CSS learning experience by (1) BTW annotations; and (2) the Learn It Online section for each chapter.

End-of-Chapter Student Activities Extensive end-of-chapter activities provide a variety of reinforcement opportunities for students where they can apply and expand their skills through individual and group work.

Organization of This Textbook

HTML, XHTML, and CSS: Complete, Sixth Edition consists of eight chapters and two Special Features on HTML, XHTML, and CSS, and six appendices. The Chapters and Appendices are organized as follows:

Chapter 1 – Introduction to HTML This introductory chapter provides students with an overview of the Internet, World Wide Web, Web pages, HTML, and Web development. Topics include the types and purposes of Web sites, Web browsers, HTML standards, and Document Object Model (DOM) and Extensible Hypertext Markup Language (XHTML) and their relationship to HTML. Additionally, Web editors, the five phases of the Web development life cycle, and the importance of validating and testing Web pages for usability are defined.

Chapter 2 – Creating and Editing a Web Page Using Inline Styles In Chapter 2, students are introduced to basic HTML tags and the various parts of a Web page. Topics include starting and quitting Notepad++ and a browser, entering headings and text into an HTML file, creating a bulleted list with HTML using Cascading Style Sheets and inline styles, adding an image and a horizontal rule, saving the HTML file and viewing it in the browser, validating the HTML code, viewing the HTML source code for a Web page, printing the HTML file and the Web page, and Web page design.

Chapter 3 – Creating Web pages with Links, Images, and Embedded Style Sheets In Chapter 3, students are introduced to linking terms and definitions. Topics include adding an e-mail link, linking to another page on the same Web site, linking to another Web site, setting link targets within a page, linking to targets, using absolute and relative paths, using different types of image files, specifying alternative text for images, defining image size, wrapping text around an image, inserting images onto Web pages, and using embedded style sheets.

Chapter 4 – Creating Tables in a Web Site Using an External Style Sheet In Chapter 4, students learn how to create tables using HTML tags. First, students assess table needs and then plan the table. Topics include table definitions and terms, table uses, creating

borderless tables; inserting images into tables; vertical and horizontal alignment within a table, adding color to a cell, adding links to another page, adding an e-mail link, using the rowspan and colspan attributes, adding captions, spacing within and between cells, and creating and using an external style sheet.

Special Feature 1 – Attracting Visitors to Your Web Site In this feature, students learn how to advertise their Web pages. Topics include using meta tags and keywords, submitting Web pages to a search engine, publicizing Web pages, and finding Web site hosting sites.

Chapter 5 – Creating an Image Map In Chapter 5, students learn how to use an image map to create more advanced Web page navigation. Topics include image mapping purpose and considerations, selecting appropriate images for mapping, dividing an image into hotspots, creating links from those hotspots, and using text links in conjunction with image links.

Chapter 6 – Creating a Form on a Web Page In Chapter 6, students create a form for collecting user input. Topics include form basics, adding controls such as check boxes, radio buttons, text boxes, and other controls on a form, creating Submit and Reset buttons, and creating an e-mail link to submit the form information back to the Web page data collector. Students also are introduced to using advanced selection menus and fieldset tags to segregate groups of information.

Chapter 7 – Using Advanced Cascading Style Sheets In Chapter 7, students are introduced to some more advanced uses of Cascading Style Sheets (CSS). Topics include adding a drop-down menu for navigation, adding pop-up image effects, and using CSS for tables instead of using HTML table tags.

Chapter 8 – Adding Multimedia Content to Web Pages In Chapter 8, students learn how to add audio and video clips to a Web page. Topics include the benefits and limitations of multimedia content for Web pages, audio and video formats, and commonly used parameters for embedded multimedia.

Special Feature 2 – Converting Frames on Your Web Site In this feature, students learn ways to convert a Web site with a frame layout to one without frames. Topics include the frame definition file, different frame layouts, and converting frames to a traditional Web page.

Appendix A – HTML Quick Reference Appendix A includes an HTML quick reference that contains the most frequently used tags and their associated attributes.

Appendix B – Browser-Safe Color Palette Appendix B summarizes the 216 browser-safe colors that appear equally well on different monitors, operating systems, and browsers.

Appendix C – Accessibility Standards for the Web Appendix C provides an overview of Web accessibility issues and the Section 508 Web accessibility guidelines used by developers to create accessible Web sites.

Appendix D – CSS Properties and Values Appendix D provides a listing of Cascading Style Sheet (CSS) properties and values together with a description of use.

Appendix E – Publishing Your Web Site Appendix E provides a quick overview of how to publish Web pages to a Web host and also discusses how to choose a Web host and how to upload your Web files via FTP.

Appendix F – Symbols and Characters Quick Reference Appendix F shows the entity characters used to represent special symbols and characters in HTML and XHTML.

End-of-Chapter Student Activities

A notable strength of the Shelly Cashman Series HTML, XHTML, and CSS books is the extensive student activities at the end of each chapter. Well-structured student activities can make the difference between students merely participating in a class and students retaining the information they learn. The activities in the Shelly Cashman Series books include the following.

CHAPTER SUMMARY A concluding paragraph, followed by a listing of the tasks completed within a chapter together with the pages on which the step-by-step, screen-by-screen explanations appear.

LEARN IT ONLINE Every chapter features a Learn It Online section that is comprised of six exercises. These exercises include True/False, Multiple Choice, Short Answer, Flash Cards, Practice Test, and Learning Games.

APPLY YOUR KNOWLEDGE This exercise usually requires students to open and manipulate a file from the Data Files that parallels the activities learned in the chapter. To obtain a copy of the Data Files for Students, follow the instructions on the inside back cover of this text.

EXTEND YOUR KNOWLEDGE This exercise allows students to extend and expand on the skills learned within the chapter.

MAKE IT RIGHT This exercise requires students to analyze a document, identify errors and issues, and correct those errors and issues using skills learned in the chapter.

IN THE LAB Three in-depth assignments per chapter require students to utilize the chapter concepts and techniques to solve problems on a computer.

CASES AND PLACES Five unique real-world case-study situations, including Make It Personal, an open-ended project that relates to student's personal lives, and one small-group activity.

Instructor Resources CD-ROM

The Instructor Resources include both teaching and testing aids.

INSTRUCTOR'S MANUAL Includes lecture notes summarizing the chapter sections, figures and boxed elements found in every chapter, teacher tips, classroom activities, lab activities, and quick quizzes in Microsoft Word files.

SYLLABUS Easily customizable sample syllabi that cover policies, assignments, exams, and other course information.

FIGURE FILES Illustrations for every figure in the textbook in electronic form.

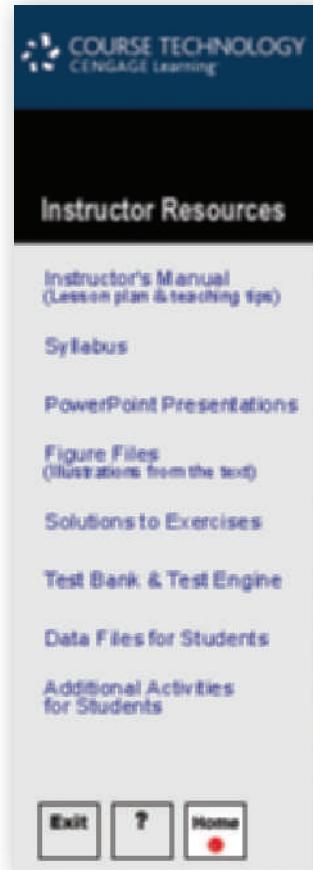
POWERPOINT PRESENTATIONS A multimedia lecture presentation system that provides slides for each chapter. Presentations are based on chapter objectives.

SOLUTIONS TO EXERCISES Includes solutions for all end-of-chapter and chapter reinforcement exercises.

TEST BANK & TEST ENGINE Test Banks include 112 questions for every chapter, featuring objective-based and critical thinking question types, and including page number references. Also included is the test engine, ExamView, the ultimate tool for your objective-based testing needs.

DATA FILES FOR STUDENTS Includes all the files that are required by students to complete the exercises.

ADDITIONAL ACTIVITIES FOR STUDENTS Consists of Chapter Reinforcement Exercises, which are true/false, multiple-choice, and short answer questions that help students gain confidence in the material learned.





Content for Online Learning

Course Technology has partnered with the leading distance learning solution providers and class-management platforms today. To access this material, Instructors will visit our password-protected instructor resources available at www.cengage.com/coursetechnology. Instructor resources include the following: additional case projects, sample syllabi, PowerPoint presentations per chapter, and more. For additional information or for an instructor username and password, please contact your sales representative. For students to access this material, they must have purchased a WebTutor PIN-code specific to this title and your campus platform. The resources for students may include (based on instructor preferences), but not limited to: topic review, review questions and practice tests.

CourseNotes

Course Technology's CourseNotes are six-panel quick reference cards that reinforce the most important concepts and features of a software application in a visual and user-friendly format. CourseNotes serve as a great reference tool during and after the student completes the course. CourseNotes are available for software applications such as Microsoft Office 2007, Word 2007, Excel 2007, Access 2007, PowerPoint 2007, and Windows 7. There are also topic-based CourseNotes available for Best Practices in Social Networking, Hot Topics in Technology, and Web 2.0. Visit www.cengage.com/ct/coursenotes to learn more!

Guided Tours

Add excitement and interactivity to your classroom with "*A Guided Tour*" product line. Play one of the brief mini-movies to spice up your lecture and spark classroom discussion. Or, assign a movie for homework and ask students to complete the correlated assignment that accompanies each topic. "*A Guided Tour*" product line takes the prep-work out of providing your students with information on new technologies and software applications and helps keep students engaged with content relevant to their lives; all in under an hour!



About Our Covers

The Shelly Cashman Series is continually updating our approach and content to reflect the way today's students learn and experience new technology. This focus on student success is reflected on our covers, which feature real students from Westfield State College using the Shelly Cashman Series in their courses, and reflect the varied ages and backgrounds of the students learning with our books. When you use the Shelly Cashman Series, you can be assured that you are learning computer skills using the most effective courseware available.

Textbook Walk-Through

The Shelly Cashman Series Pedagogy: Project-Based — Step-by-Step — Variety of Assessments

Plan Ahead boxes prepare students to create successful projects by encouraging them to think strategically about what they are trying to accomplish before they begin working.

Step-by-step instructions now provide a context beyond the point-and-click. Each step provides information on why students are performing each task, or what will occur as a result.

HTML 36 HTML Chapter 2 Creating and Editing a Web Page Using Inline Styles

Plan Ahead

General Project Guidelines

When creating a Web page, the actions you perform and decisions you make will affect the appearance and characteristics of the finished page. As you create a Web page, such as the project shown in Figure 2-1 on the previous page, you should follow these general guidelines:

1. **Complete Web page planning.** Before developing a Web page, you must know the purpose of the Web site, identify the users of the site and their computing environments, and decide who owns the information on the Web page.
2. **Analyze the need for the Web page.** In the analysis phase of the Web development life cycle, you should analyze what content to include on the Web page. In this phase, you determine the tasks and the information that the users need. Refer to Table 1-4 on page HTML 15 for information on the phases of the Web development life cycle.
3. **Choose the content for the Web page.** Once you have completed the analysis, you need to determine what content to include on the Web page. Follow the *less is more* principle. The less text, the more likely the Web page will be read. Use as few words as possible to make a point.
4. **Determine the file naming convention that you will use for this Web page.** Before you start creating and saving files, you should decide on a standard way of naming your files. Should you use the .htm or .html extension? As explained later in the chapter, you use the .htm extension when the host Web server only allows short file names. You use .html when the host Web server allows long file names. What name should you give your file to indicate the file's content or purpose? For instance, naming a Web page page1.html does not describe what that Web page is; a more descriptive name is helpful in development of the Web site.
5. **Determine where to save the Web page.** You can store a Web page permanently, or save it, on a variety of storage media, including a hard disk, USB flash drive, CD, or DVD. Your instructor or the company for whom you are developing the Web page may have specific storage media requirements.
6. **Determine what folder structure to use on your storage device.** Once you have determined the storage media to use, you should also determine folder location, structure, and names on which to save the Web page. This should be done before you start to save any of your files.

HTML 54 HTML Chapter 2 Creating and Editing a Web Page Using Inline Styles

To Save an HTML File

You have performed many steps in creating this project and do not want to risk losing the work you have done so far. Also, to view HTML in a browser, you must save the file. The following steps show how to save an HTML file.

- 1 With a USB flash drive connected to one of the computer's USB ports, click File on the Notepad++ menu bar (Figure 2-15).

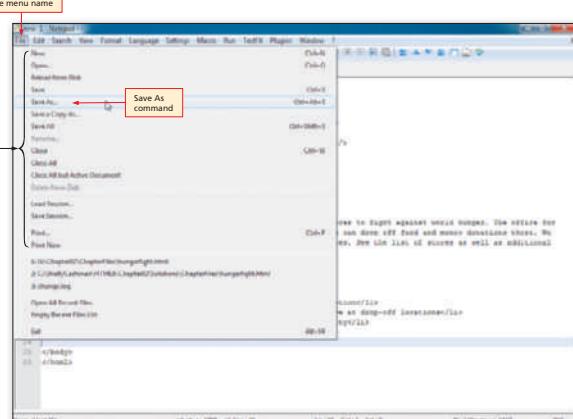


Figure 2-15

- 2 Click Save As on the File menu to display the Save As dialog box (Figure 2-16).

Q&A Do I have to save to a USB flash drive?
No. You can save to any device or folder. A folder is a specific location on a storage medium. Use the same process, but select your device or folder.

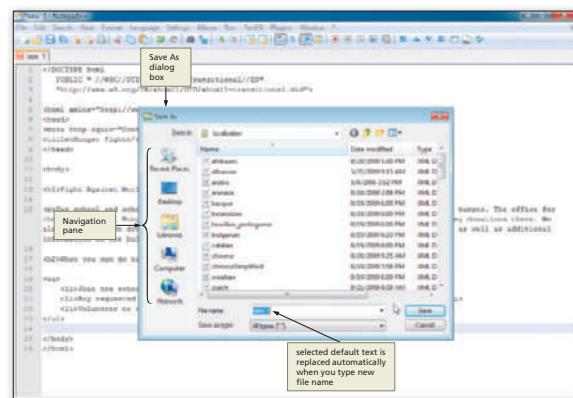


Figure 2-16

format various elements of the Web page. The overall appearance of a page greatly affects its ability to communicate clearly. Examples of how you can affect the appearance, or format, of the Web page include adding an image, color, horizontal rules, and graphical images.

horizontal rules. Eye-catching graphical images help convey the Web message and add visual interest. Graphics can be used to show a product, benefit, or visually convey a message that is not expressed easily

to position and how to format the graphical images. The position and how graphical images should grab the attention of passersby and draw them to the Web page.

for XHTML compliance. An important part of Web development is making sure that your Web page follows XHTML standards. The World Wide Web Consortium (W3C) has an online validator that allows you to test your Web page and find errors.

more specific details concerning the above guidelines are presented in the chapter. The chapter also will identify the actions performed regarding these guidelines during the creation of the Web page shown

Textbook Walk-Through

HTML 76 HTML Chapter 2 Creating and Editing a Web Page Using Inline Styles

2

- Click the Browse button.
- Locate the hungerfight.html file on your storage device and then click the file name.
- Click the Open button on the Choose File to Upload dialog box and the file path and name will be inserted into the File box, as shown in Figure 2-35.

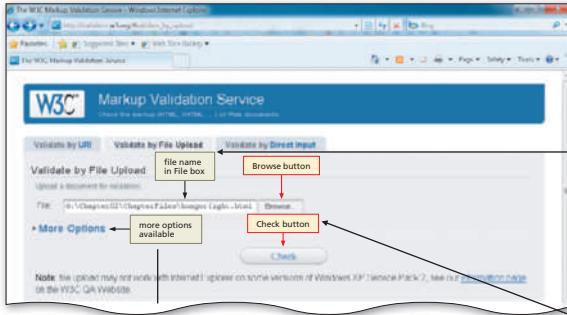


Figure 2-35

3

- Click the Check button (Figure 2-35). The resulting validation should be displayed, as shown in Figure 2-36.
- Return to the Hunger Fight Web page, either by clicking the Back button on your browser or by clicking the Internet Explorer button in the taskbar.

Q&A How do I know if my HTML code is valid? The notification header will be green, and in the Result area, you should see the word "Passed."

Q&A What can I do if my HTML code is not validated?

If your code has errors, edit your HTML file to correct the errors. The Markup Validation Service will tell you what is wrong with your code. Once you make the necessary changes and save the corrected HTML file, then scroll down and click the Revalidate button.

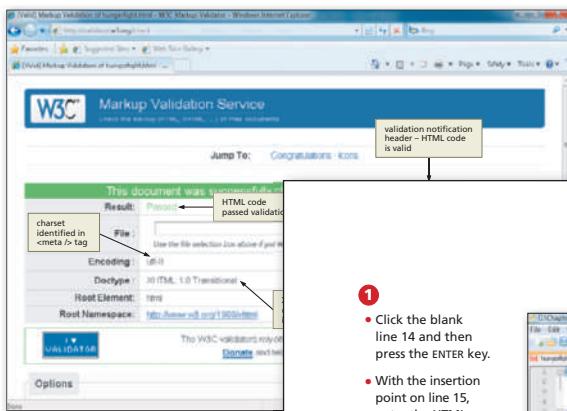


Figure 2-36

Q&A boxes offer questions students may have when working through the steps and provide additional information about what they are doing right where they need it.

BTW

Screen Shots

Callouts in screenshots give students information they need, when they need to know it. The Series has always used plenty of callouts to ensure that students don't get lost. Now, use color to distinguish the content in the callouts to make them more meaningful.

Explanatory callouts summarize what is happening on screen.

Navigational callouts in red show students where to click.

1

- Click the blank line 14 and then press the ENTER key.
- With the insertion point on line 15, enter the HTML code, as shown in Table 2-9. Press ENTER at the end of each line (Figure 2-26).

Q&A

What is the purpose of the alt attribute?

The alt attribute has three important purposes. First, screen readers used by users with visual impairments read the alternate text out loud. Second, the alternate text is displayed while the image is being loaded. Finally, the alt tag is required for XHTML compliance.

Q&A

What is the purpose of the <div> and </div> tags? The <div> tag establishes a division within a Web page. When you use the <div>-</div> tags, you are able to design a layout that uses CSS, including inserting images.

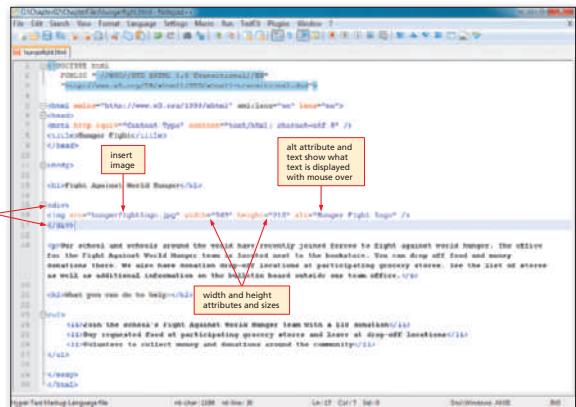


Figure 2-26

Make other visual enhancements.

In addition to images, there are several ways to add visual interest.

Add color to headings. Web developers often use colors to call attention to elements on a Web page. The color selected should coordinate with the images selected for the page. It also should allow the Web page text to be read easily. Some colors, such as dark colors, may not be appropriate because the default black text cannot be displayed effectively on a dark background. When changing the color of an element such as a heading, it is usually best to apply the same style to all headings on the Web page for consistency.

Change the font style of a bulleted list. It is sometimes aesthetically pleasing to change the style of the text in a bulleted list. When you want to call attention to the information, you might want to italicize or bold the bullet text.

Insert a horizontal rule. It is useful to use a horizontal rule to break up text on a Web page. A horizontal rule is used as a divider for a page to separate text sections.

Plan Ahead

Textbook Walk-Through

HTML 60 **HTML Chapter 2** Creating and Editing a Web Page Using Inline Styles

Q&A

2

- Press the ENTER key to display the hungerfight.html page as it were available on the Web (Figure 2–23).

What if I get a warning from Internet Explorer (IE) that says, “Internet Explorer needs to open a new window to display this webpage.”? If this happens, you should click the OK button to continue. You then will see your Web page displayed in another IE window.

What if my page is not displayed correctly? Check your entries for hungerfight.html carefully against Figure 2–19 to make sure you have not made any typing errors or left anything out. Correct the errors, resave the file, and try again.

Other Ways

- In Windows Explorer, double-click HTML file name to open in default browser
- In Windows Explorer, right-click HTML file name, click Open with, click browser name
- Click Tools, Toolbars, Menu Bar if menu is not displayed; on Menu bar click File, Open, and browse to desired file

BTW

User Interface Design

The user interface design is a very important aspect of a Web site. If a site is designed poorly, users may be unable to find the desired information or complete a task, which makes the Web site ineffective. There are many good Web sites available that discuss Web design principles.

title

main h1 heading

h2 heading

bulleted list

body of Web page

Internet Explorer button selected (active)

Notepad++ button not selected (inactive)

Figure 2-23

Other Ways boxes that follow many of the step sequences explain the other ways to complete the task presented.

HTML 82 **HTML Chapter 2** Creating and Editing a Web Page Using Inline Styles

Apply Your Knowledge *continued*

- Save the revised HTML file in the Chapter02\Apply folder using the file name apply2-1solution.html.
- Validate your HTML code at <http://validator.w3.org/>.
- Enter g:\Chapter02\Apply\apply2-1solution.html as the URL to view the revised Web page in your browser.
- Print the Web page.
- Submit the revised HTML file and Web page in the format specified by your instructor.

STUDENT ASSIGNMENTS

Extend Your Knowledge

Extend the skills you learned in this chapter and experiment with new skills.

Creating a Definition List

Instructions: Start Notepad++. Open the file, extend2-1.html from the Chapter02\Extend folder of the Data Files for Students. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files. This sample Web page contains all of the text for the Web page in bulleted list format. You will add the necessary tags to make this a definition list with terms that are bold, as shown in Figure 2–42.

definition terms

definitions

Figure 2-42

Perform the following tasks:

- Using the text given in the file extend2-1.html, make changes to the HTML code to change the Web page from a bulleted list to a definition list by following the definition list code shown in Figure 2–14 on page HTML 52.

Textbook Walk-Through

Make It Right projects call on students to analyze a file, discover errors in it, and fix them using the skills they learned in the chapter.

2. Add the additional HTML code necessary to make the terms bold (see font-weight for style in Appendix D) and a background color of #e0e0e0 (see background-color in Appendix D).
3. Save the revised document in the Chapter02\Extend folder with the file name extend2-1solution.html, validate the Web page, and then submit it in the format specified by your instructor.

Make It Right

Analyze a document and correct all errors and/or improve the design.

Correcting the Friendly Reminder Web Page

Instructions: Start Notepad++. Open the file makeitright2-1.html from the Chapter02\MakeItRight folder of the Data Files for Students. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files. The Web page is a modified version of what you see in Figure 2-43. Make the necessary corrections to the Web page to make it look like Figure 2-43. Format the heading to use the Heading 1 style center-aligned on the Web page (see the text-align property in Appendix D to center a heading). Add two size-10px horizontal rules, as shown in Figure 2-43. Save the file in the Chapter02\MakeItRight folder as makeitright2-1solution.html, validate the Web page, and then submit it in the format specified by your instructor.

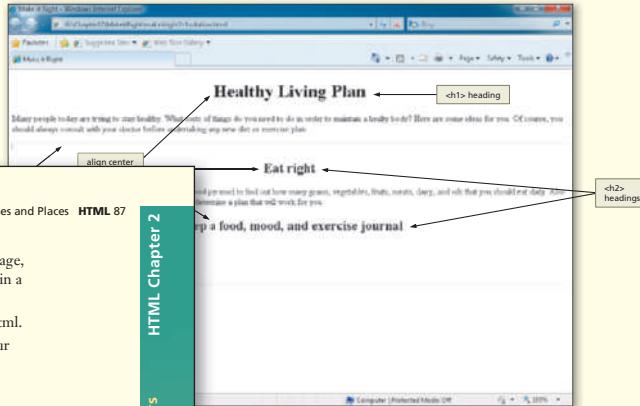


Figure 2-43

3. Insert the image file cloudy.jpg, stored in the Chapter02\IntheLab folder. Right-click the image, click Properties, and then click the Details tab to find out the image's dimensions, or open it in a graphics program. Note that the bullets used for the list are square in shape.
4. Save the HTML file in the Chapter02\IntheLab folder using the file name lab2-3solution.html.
5. Enter g:\Chapter02\IntheLab\lab2-3solution.html as the URL to view the Web page in your browser.
6. Print the Web page from your browser.
7. Submit the revised HTML file and Web page in the format specified by your instructor.

Cases and Places

Apply your creative thinking and problem-solving skills to design and implement a solution.

• EASIER • MORE DIFFICULT

• 1: Add to the Food Drive Web Page

Mr. Wattigney, the director of the Community Food Bank, likes the Web page you created for Lab 2-1. Now that the Food Drive is over, he would like you to update the Web page with new information on upcoming community events. Before updating the page, search the Web to review the Web pages at other food banks or departments of community services for ideas on content to include or formatting to change. What do their Web sites look like? Are there changes you can make to the Lab 2-1 Web page that reflect what other places have done? Using the concepts presented in this chapter, include additional information or change the formatting to make the page more interesting and timely.

• 2: Create an Artist Web Site

You are creating a new Web site for a local photographer. The photographer has asked that you use descriptive alt attributes for images on the Web page, because many of the viewers of his Web page have very slow Internet connections and images often do not load quickly. Search the Web for information on adding useful, descriptive alt attributes for images. Also find information on using thumbnail images. Give suggestions for loading images faster. Create a document with a brief paragraph explaining the information that you found in your research. Make suggestions about how the photographer's Web site can be made more effective.

•• 3: Create a Web Page of CSS Properties

You have learned a lot about using Cascading Style Sheets (CSS) so far in the class. You still have some questions, though, about how to insert certain styles. Research information on the Web (don't forget to look at the W3.org site) to find Web sites that list CSS properties and values that complement Appendix D. In a Word document, add the URLs that you found to be very helpful. Under each URL write a brief paragraph that describes why you thought the site could be useful for new Web developers. Share your list of URLs with fellow students.

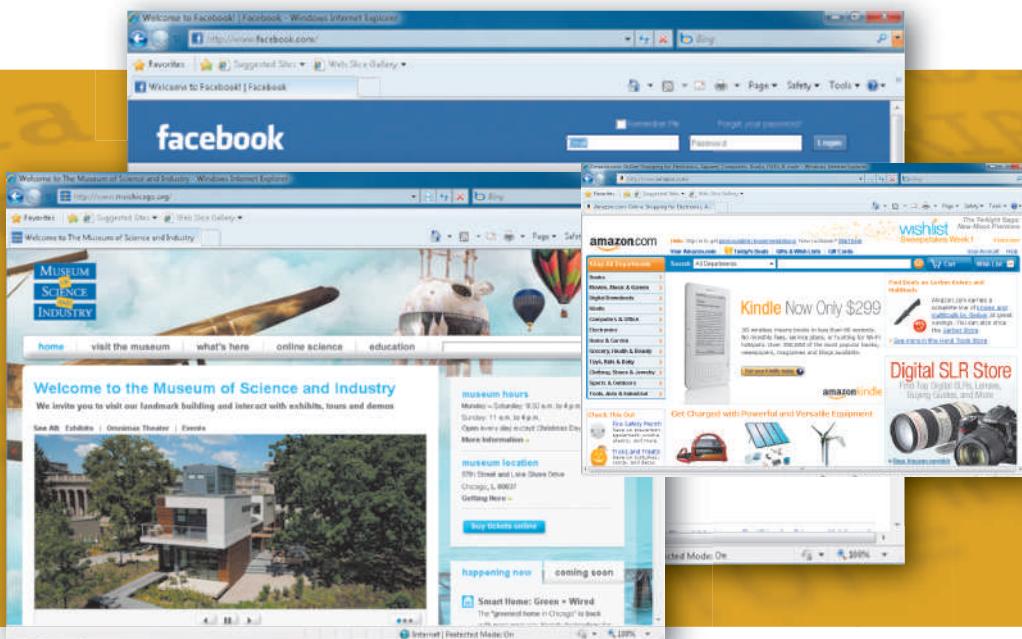
•• 4: Create a Personal Web Page

Make It Personal

Your class instructor wants to post all of the students' Web pages on the school server to show what his or her students are interested in. Create a Web page of personal information, listing items such as your school major, jobs that you have had in the past, and your hobbies and interests. To make your personal

Found within the Cases & Places exercises, the **Make It Personal** exercises call on students to create an open-ended project that relates to their personal lives.

1 Introduction to HTML, XHTML, and CSS



Objectives

You will have mastered the material in this chapter when you can:

- Describe the Internet and its associated key terms
- Describe the World Wide Web and its associated key terms
- Describe the types and purposes of Web sites
- Discuss Web browsers and identify their purpose
- Define Hypertext Markup Language (HTML) and its associated standards used for Web development
- Discuss the use of Cascading Style Sheets (CSS) in Web development
- Define the Document Object Model (DOM) and describe its relationship to HTML
- Define Extensible Hypertext Markup Language (XHTML) and describe its relationship to HTML
- Identify tools used to create HTML documents
- Describe the five phases of the Web development life cycle
- Describe the different methods of Web site design and the purpose of each Web site structure
- Discuss the importance of testing throughout the Web development life cycle
- Explain the importance of being an observant Web user

1 | Introduction to HTML, XHTML, and CSS

Introduction

Before diving into the details of creating Web pages with HTML, XHTML, and CSS, it is useful to look at how these technologies relate to the development of the Internet and the World Wide Web. The Internet began with the connection of computers and computer networks. This connectivity has had a huge impact on our daily lives. Today, millions of people worldwide have access to the Internet, the world's largest network. Billions of Web pages, providing information on any subject you can imagine, are currently available on the World Wide Web. People use the Internet to search for information, to communicate with others around the world, and to seek entertainment. Students register for classes, pay tuition, and find out final grades via this computer network. Stores and individuals sell their products using computer connectivity, and most industries rely on the Internet and the World Wide Web for business transactions.

Hypertext Markup Language (HTML) and more recently Extensible Hypertext Markup Language (XHTML) and Cascading Style Sheets (CSS) allow the World Wide Web to exist. In order to utilize these technologies effectively, you need to understand the main concepts behind the Internet and HTML. In this chapter, you learn some basics about the Internet, the World Wide Web, intranets, and extranets. You are introduced to Web browsers, definitions of HTML and associated key terms, the five phases of the Web development life cycle, and the tasks that are involved in each phase.

What Is the Internet?

Most people today have had exposure to the Internet at school, in their homes, at their jobs, or at their local library. The **Internet** is a worldwide collection of computers and computer networks that links billions of computers used by businesses, government, educational institutions, organizations, and individuals using modems, phone lines, television cables, satellite links, fiber-optic connections, and other communications devices and media (Figure 1–1).

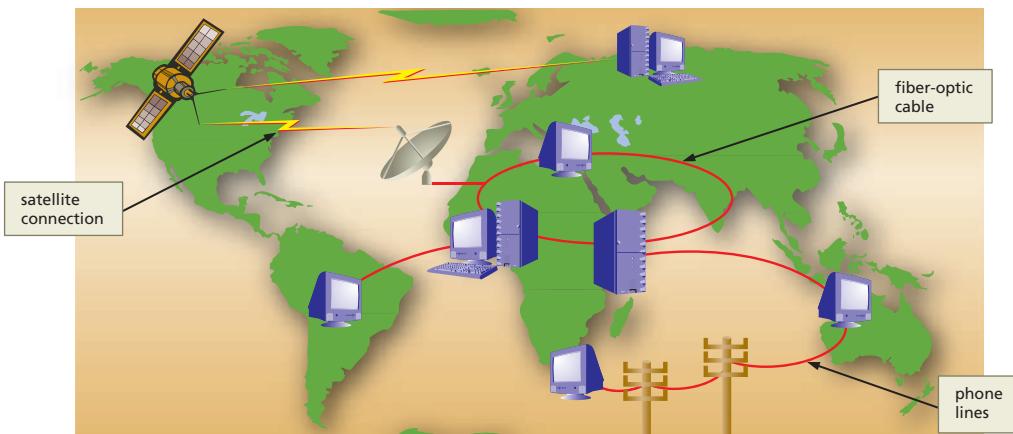


Figure 1–1 The Internet is a worldwide collection of computer networks.

A **network** is a collection of two or more computers that are connected to share resources and information. Today, high-, medium-, and low-speed data lines connect networks. These data lines allow data (including text, graphical images, and audio and video data) to move from one computer to another. The **Internet backbone** is a collection of high-speed data lines that connect major computer systems located around the world. An **Internet service provider (ISP)** is a company that has a permanent connection to the Internet backbone. ISPs utilize high- or medium-speed data lines to allow individuals and companies to connect to the backbone for access to the Internet. An Internet connection at home generally is a DSL or cable data line that connects to an ISP.

Millions of people in most countries around the world connect to the Internet using computers in their homes, offices, schools, and public locations such as libraries. Users with computers connected to the Internet can access a variety of services, including e-mail, social networking, online shopping, and the World Wide Web (Figure 1–2).



Figure 1–2 The Internet makes available a variety of services such as e-mail and the World Wide Web.

What Is the World Wide Web?

The **World Wide Web**, also called the **Web**, is the part of the Internet that supports multimedia and consists of a collection of linked documents. To support multimedia, the Web relies on the **Hypertext Transfer Protocol (HTTP)**, which is a set of rules for exchanging text, graphic, sound, video, and other multimedia files. The linked documents, or pages of information, on the Web are known as **Web pages**. Because the Web supports text, graphics, sound, and video, a Web page can include any of these multimedia elements. The Web is ever-changing and consists of billions of Web pages. Because of the ease of creating Web pages, more are being added all the time.

BTW

Internet and WWW History

The World Wide Web Consortium (W3C or w3.org), the de facto organization that governs HTML, provides a particularly rich history of the Internet and the World Wide Web. Search on “Internet history” or “WWW history” in your browser for many additional sources.

A **Web site** is a related collection of Web pages that is created and maintained by an individual, company, educational institution, or other organization. For example, as shown in Figure 1–3, many organizations, such as the Museum of Science and Industry in Chicago, publish and maintain Web sites. Each Web site contains a **home page**, which is the first document users see when they access the Web site. The home page often serves as an index or table of contents to other documents and files displayed on the site.



Figure 1–3 A Web site is a related collection of Web pages that is created and maintained by an individual, company, educational institution, or other organization.

Web Servers

Web pages are stored on a **Web server**, or **host**, which is a computer that stores and sends (serves) requested Web pages and other files. Any computer that has Web server software installed and is connected to the Internet can act as a Web server. Every Web site is stored on, and runs from, one or more Web servers. A large Web site may be spread over several servers in different geographic locations.

In order to make the Web pages that you have developed available to your audience, you have to publish those pages. **Publishing** is copying the Web pages and associated files such as graphics and audio to a Web server. Once a Web page is published, anyone who has access to the Internet can view it, regardless of where the Web server is located. For example, although the Chicago Museum of Science and Industry Web site is stored on a Web server somewhere in the United States, it is available for viewing by anyone in the world. Once a Web page is published, it can be read by almost any computer: whether you use the Mac, Windows, or Linux operating system, with a variety of computer hardware, you have access to billions of published Web pages.

Web Site Types and Purposes

The three general types of Web sites are Internet, intranet, and extranet. Table 1–1 lists characteristics of each of these three types of Web sites.

Table 1–1 Types of Web Sites

Type	Users	Access	Applications
Internet	Anyone	Public	Used to share information such as personal information, product catalogs, course information with the public
intranet	Employees or members	Private	Used to share information such as forms, manuals, organization schedules with employees or members
extranet	Select business partners	Private	Used to share information such as inventory updates, product specifications, financial information with business partners and customers

An **Internet site**, also known as a **Web site**, is a site generally available to the public. Individuals, groups, companies, and educational institutions use Web sites for a variety of purposes. Intranets and extranets also use Internet technology, but access is limited to specified groups. An **intranet** is a private network that uses Internet technologies to share company information among employees. An intranet is contained within a company or organization's network, which makes it private and only available to those who need access. Policy and procedure manuals usually are found on an intranet. Other documents such as employee directories, company newsletters, product catalogs, and training manuals often are distributed through an intranet.

An **extranet** is a private network that uses Internet technologies to share business information with select corporate partners or key customers. Companies and organizations can use an extranet to share product manuals, training modules, inventory status, and order information. An extranet also might allow retailers to purchase inventory directly or to pay bills online.

Companies use Web sites to advertise or sell their products and services worldwide, as well as to provide technical and product support for their customers. Many company Web sites also support **electronic commerce (e-commerce)**, which is the buying and selling of goods and services on the Internet. Using e-commerce technologies, these Web sites allow customers to browse product catalogs, comparison shop, and order products online. Figure 1–4 shows Amazon.com, which is a company that sells products only online. Many company Web sites also provide job postings and announcements, a frequently asked questions (FAQs) section, customer feedback links to solicit comments from their customers, and searchable technical support databases.



Intranets and Extranets

There are many Web sites that discuss ideas for intranets and extranets.

Many companies are already using these technologies and share their "best practice" techniques. The CIO Intranet/Extranet Research Center provides valuable information on building and maintaining an intranet or extranet, along with additional resources.



E-Commerce

Today, e-commerce is a standard part of doing business. E-commerce technologies, however, continue to change, offering new applications and potential uses. Several online magazines are dedicated to providing an in-depth look at e-commerce. Many print magazines also provide useful information about this important way to do business.

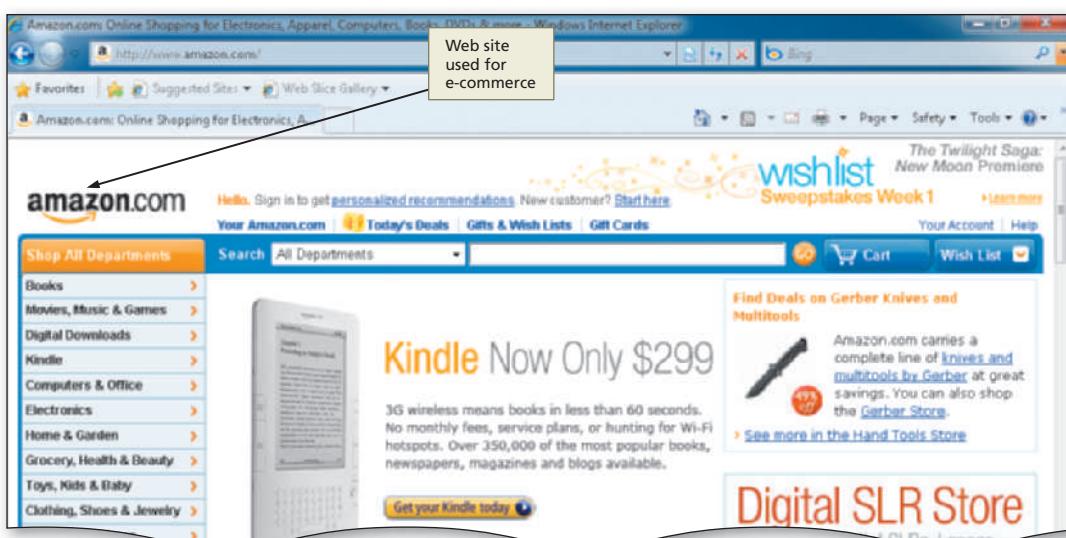


Figure 1–4 Amazon.com is a company that operates online only.

Colleges, universities, and other schools use Web sites to distribute information about areas of study, provide course information, or register students for classes online. Instructors use their Web sites to issue announcements, post questions on reading material, list contact information, and provide easy access to lecture notes and slides. Many instructors today use the course management software adopted by their respective schools to upload course content. Using a standard course management product across a university makes it easier for students to find information relative to their various courses. Many course management tools allow instructors to write their own Web content for courses. With many systems, instructors can use Web pages to provide further information for their students within the structure of the course management tool provided by the school. In addition to keeping in contact with current students via the Web, universities also utilize the Web to keep in touch with their alumni, as shown in Figure 1–5.



Figure 1–5 Many universities have alumni Web sites.

In addition to the use of the Internet by companies and educational institutions, individuals might create personal Web sites that include their résumés to make them easily accessible to any interested employers. Families can share photographs, video and audio clips, stories, schedules, or other information through Web sites (Figure 1–6). Many individual Web sites allow password protection, which makes a safer environment for sharing information.



Figure 1–6 Personal Web sites are used to communicate with family and friends.

Web Browsers

To display a Web page on any type of Web site, a computer needs to have a Web browser installed. A **Web browser**, also called a **browser**, is a program that interprets and displays Web pages and enables you to view and interact with a Web page. Microsoft Internet Explorer, Mozilla Firefox, and Apple Safari are popular browsers today. Browsers provide a variety of features, including the capability to locate Web pages, to link forward and backward among Web pages, to add a favorite or bookmark a Web page, and to choose security settings.

To locate a Web page using a browser, you type the Web page's Uniform Resource Locator (URL) in the browser's Address or Location bar. A **Uniform Resource Locator (URL)** is the address of a document or other file accessible on the Internet. An example of a URL on the Web is:

http://www.scsite.com/html6e/index.html

The URL indicates to the browser to use the HTTP communications protocol to locate the index.html Web page in the html6e folder on the scsite.com Web server. Web page URLs can be found in a wide range of places, including school catalogs, business cards, product packaging, and advertisements.

Hyperlinks are an essential part of the World Wide Web. A **hyperlink**, also called a **link**, is an element used to connect one Web page to another Web page that's located on the same server or used to link Web pages located on a different Web server located anywhere in the world. Clicking a hyperlink allows you to move quickly from one Web page to another, and the user does not have to be concerned about where the Web pages reside. You also can click hyperlinks to move to a different section of the same Web page.

With hyperlinks, a Web site user does not necessarily have to view information in a linear way. Instead, he or she can click the available hyperlinks to view the information in a variety of ways, as described later in this chapter. Many different Web page elements, including text, graphics, and animations, can serve as hyperlinks. Figure 1–7 shows examples of several different Web page elements used as hyperlinks.

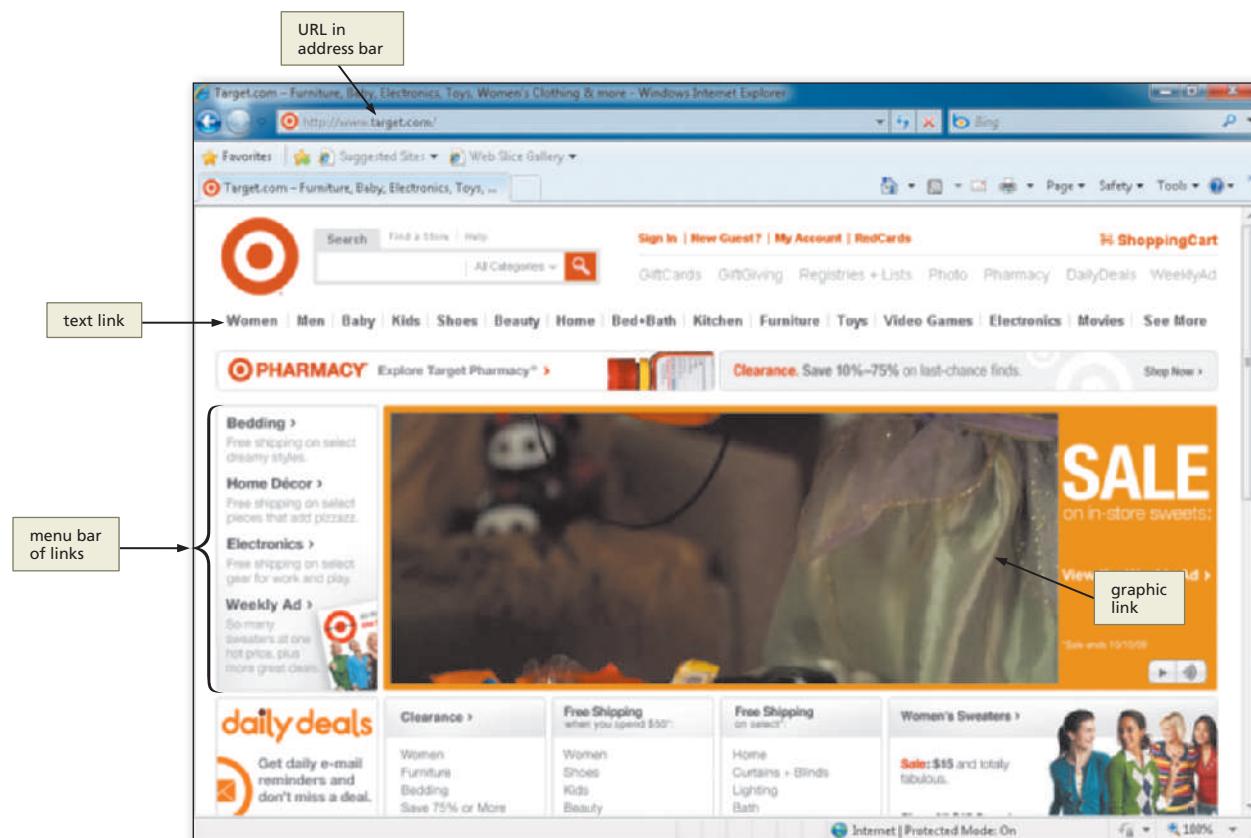


Figure 1–7 A Web page can use many different Web page elements as hyperlinks.

What Is Hypertext Markup Language?

Web pages are created using **Hypertext Markup Language (HTML)**, which is an authoring language used to create documents for the World Wide Web. HTML uses a set of special instructions called **tags** or **markup** to define the structure and layout of a Web document and specify how the page is displayed in a browser.

A Web page is a file that contains both text and HTML tags. HTML tags mark the text to define how it should appear when viewed as a page on the Web. HTML includes hundreds of tags used to format Web pages and create hyperlinks to other documents or Web pages. For instance, the HTML tags `<p>` and `</p>` are used to indicate a new paragraph with a blank line above it, `<table>` and `</table>` are used to indicate the start and end of a table, and `<hr />` is used to display a horizontal rule across the page. Figure 1–8a shows the HTML tags needed to create the Web page shown in Figure 1–8b. You also can enhance HTML tags by using attributes, as shown in Figure 1–8a. **Attributes** define additional characteristics such as font weight or style for the HTML tag.

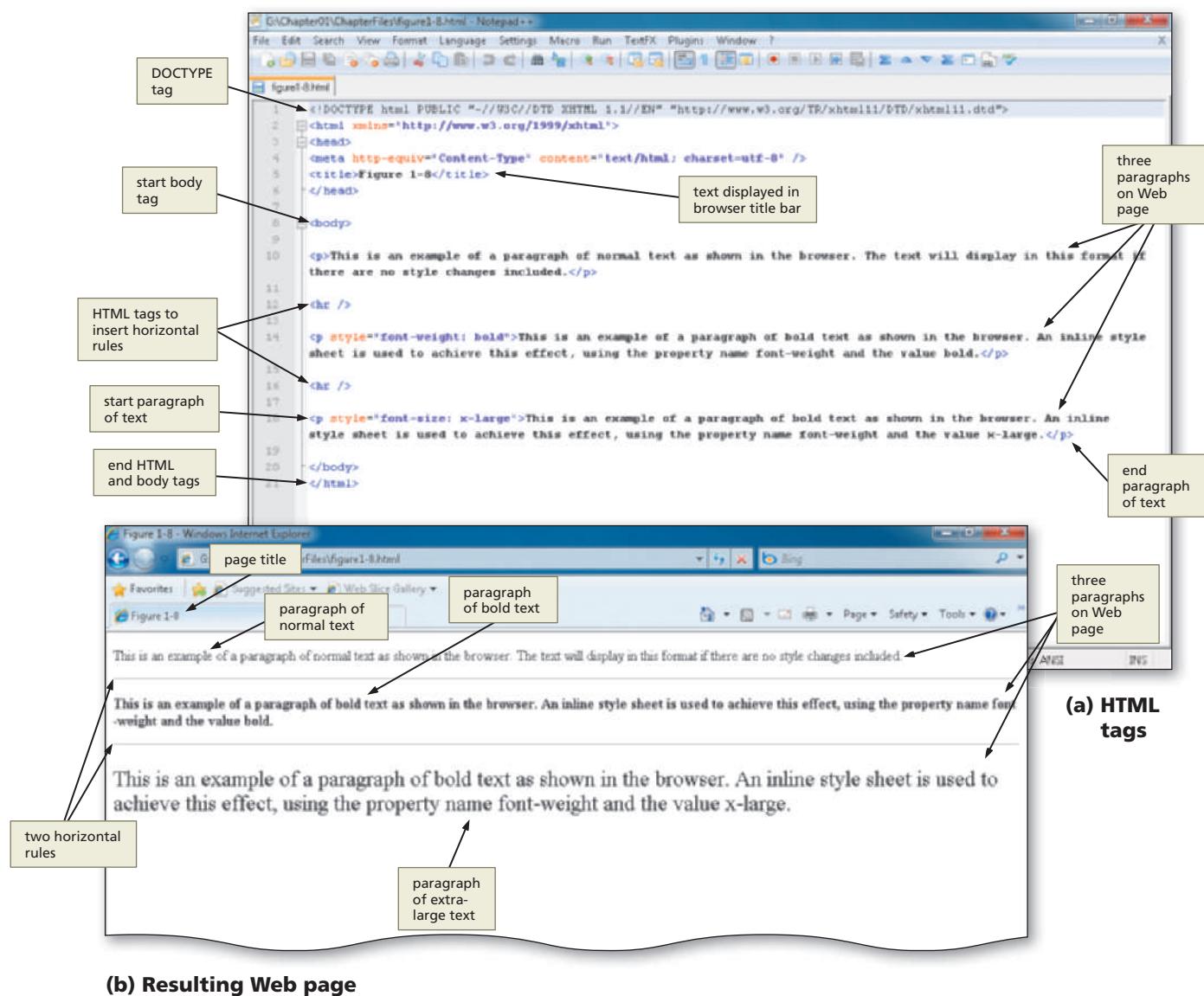


Figure 1-8 A Web page is a file that contains both text and HTML tags.

HTML is **platform independent**, meaning you can create, or code, an HTML file on one type of computer and then use a browser on another type of computer to view that file as a Web page. The page looks the same regardless of what platform you are using. One of the greatest benefits of Web technology is that the same Web page can be viewed on many different types of digital hardware, including mobile devices like smart phones.

HTML Elements

HTML combines descriptive tags with special tags that denote how a document should appear in a Web browser. HTML elements include headings, paragraphs, hyperlinks, lists, images, and more. Most HTML elements consist of three parts: a start tag, content, and an end tag. For example, to specify that certain text should appear bold in a Web page, you would enter the following HTML code:

```
<bold>this is bold text</bold>
```

where `<bold>` is the start bold code and `</bold>` is the end bold code. Table 1–2 on the next page shows examples of some HTML elements.

Table 1-2 HTML Elements

Element	Tag	Purpose
Title	<title>...</title>	Indicates title to appear on the title bar in the browser
Body	<body>...</body>	Specifies what appears on the Web page; all Web page content is inserted within the start <body> tag and end </body> tag
Paragraph	<p>...</p>	Inserts a blank line before paragraph text
Line Break	 	Inserts a line break before the next element without a blank line

BTW

HTML Elements

Numerous sources of information about HTML elements are available. The World Wide Web Consortium (w3.org) provides the most comprehensive list of tags and attributes together with examples of their use. One of the main goals of the W3C is to help those building Web sites understand and utilize standards that make the Web accessible to all.

HTML Coding Practices

Similar to all programming languages, HTML has a set of coding practices designed to simplify the process of creating and editing HTML files and ensure that Web pages appear correctly in different browsers.

When creating an HTML file, you should separate sections of the HTML code with spaces and by using the Tab key. Adding space between sections, either with blank lines or by tabbing, gives you an immediate view of the sections of code that relate to one another and helps you view the HTML elements in your document more clearly. HTML browsers ignore spaces that exist between the tags in your HTML document, so the spaces and indentations inserted within the code will not appear on the Web page. Figure 1–9 shows an example of an HTML file with code sections separated by blank lines and code section indentations. Another developer looking at this code can see immediately where the specific sections are located in the code.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Figure 1-9</title>
</head>
<body>
<table>
<tr>
<th>Table Example</th>
</tr>
<tr>
<td>First data cell in table</td>
<td>Second data cell in table</td>
<td>Third data cell in table</td>
</tr>
</table>
<p>This text displays as a new paragraph. It has a blank line between it and the bulleted list above.</p>
<ul>
<li>Item one in a bulleted list</li>
<li>Item two in a bulleted list</li>
<li>Item three in a bulleted list</li>
</ul>
</body>
</html>

```

Figure 1–9 Adding spaces to HTML code separates sections to make reading easier.

HTML Versions

HTML has gone through several versions, each of which expands the capabilities of HTML. Although HTML version 5.0 is under development, it has not yet been released. The most recent version of HTML is HTML 4.01, although most browsers still support HTML versions 3.2 and 2.0. To ensure that browsers can interpret each new version of HTML, the World Wide Web Consortium (W3C) maintains HTML standards, or specifications, which are publicly available on its Web site. As described later in this chapter, it is important to verify that Web pages are displayed as intended in a variety of browsers during the testing phase of development.

Cascading Style Sheets

This book has taken a new direction with the 6th edition by eliminating deprecated tags and attributes. **Deprecated** tags and attributes are tags and attributes that are no longer recommended in the latest W3C standard. In Appendix A, deprecated tags and attributes are highlighted with an asterisk. In an effort to eliminate deprecated HTML tags, the chapter projects utilize Cascading Style Sheets (CSS) to alter the style (or look) of a Web page. Although HTML allows Web developers to make changes to the structure, design, and content of a Web page, it is limited in its ability to define the appearance, or style, across one or more Web pages. **Cascading Style Sheets (CSS)** allow you to specify styles for various Web page elements. A **style** is a rule that defines the appearance of a Web page element. A **style sheet** is a series of rules that defines the style for a Web page or an entire Web site. With a style sheet, you can alter the appearance of a Web page or pages by changing characteristics such as font family, font size, margins, and link specifications, as well as visual elements such as colors and borders. CSS is not used to add any content to your Web site; it just makes your content look more stylish.

With CSS you can specify the style for an element within a single Web page or throughout an entire Web site. For example, if you want all text paragraphs on a Web page to be indented by five spaces, you can use a style sheet to handle the indenting, rather than coding each paragraph with an indentation. And, if you decided you wanted to change the indent to three spaces, you would change just one style sheet line rather than changing the coding for each paragraph. So you can see that using CSS saves a lot of time and makes it much easier to make style changes.

CSS is not HTML; it is a separate language used to enhance the display capabilities of HTML. The World Wide Web Consortium (W3C), the same organization that defines HTML standards, defines the specifications for CSS. Appendix A at the back of this book and available online provides a list of HTML tags and corresponding attributes that will allow you to alter the Web page elements as needed, and Appendix D has complete information on the properties and values associated with different CSS elements.

BTW

CSS, DHTML, and XHTML

The w3.org Web site has an extensive amount of information and tutorials about Cascading Style Sheets (CSS), Dynamic HTML (DHTML), and Extensible HTML (XHTML). The standards suggested in the W3C Web site are the ones that most Web developers follow.

Document Object Model (DOM)

HTML can be used with other Web technologies to provide additional Web page functionality. For example, the term **Document Object Model (DOM)** describes a combination of HTML tags, CSS, and a scripting language such as JavaScript. DOM allows JavaScript and other languages to manipulate the structure of the underlying document to create interactive, animated Web pages. This is a model in which the Web page (or document) contains objects (elements, links, etc.) that can be manipulated. This allows a Web developer to add, delete, or change an element or attribute. Web pages enhanced with

DOM can be more responsive to visitor interaction than basic HTML Web pages. Not all interactive Web pages require DOM, but if you have a need for extensive interactivity, then this might be a model to consider. CSS, JavaScript, and DOM are covered in later chapters in the Comprehensive (12-chapter) version of this book.

Extensible Hypertext Markup Language (XHTML)

As you have learned, HTML uses tags to describe how a document should appear in a Web browser, or the Web page format. **Extensible Markup Language (XML)** is a markup language that uses tags to describe the structure and content of a document, not the format. **Extensible Hypertext Markup Language (XHTML)** is a reformulation of HTML formatting so it conforms to XML structure and content rules. By combining HTML and XML, XHTML combines the display features of HTML and the stricter coding standards required by XML.

If you create a Web page in HTML and do not follow XHTML coding standards exactly (for example, by not using an end `</p>` tag), the Web browser on your computer can still interpret and display the Web page correctly. However, newer types of browsers, such as those for mobile phones or handheld computers, cannot interpret HTML code that does not meet XHTML standards. Because XHTML has such strict coding standards, it helps ensure that Web pages created in XHTML will be readable by many different types of applications. An important step in Web development is to check that your Web pages are XHTML compliant. You will validate your Web pages starting in Chapter 2 and continue that process throughout the book.

Table 1–3 lists some of the XHTML coding rules that Web developers should follow to ensure that their HTML code conforms to XHTML standards. All of the projects in this book follow XHTML standards (as discussed in Chapter 2) and adhere to the rules outlined in Table 1–3. The specifics of each rule are explained in detail when used in a project.

Table 1–3 XHTML Coding Practices

Practice	Invalid Example	Valid Example
HTML file must include a DOCTYPE statement	<pre><html> <head><title>sample Web page</title></pre>	<pre><!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/ TR/xhtml1/DTD/xhtml1-transitional.dtd"> <html> <head><title>sample Web page</title> </head></pre>
All tags and attributes must be written in lowercase	<pre><TABLE WIDTH="100%"></pre>	<pre><table width="100%"></pre>
All attribute values must be enclosed by single or double quotation marks	<pre><table width=100%></pre>	<pre><table width="100%"></pre>
All tags must be closed, including tags such as img, hr, and br, which do not have end tags, but which must be closed as a matter of practice	<pre>
 <hr> <p>This is another paragraph</pre>	<pre>
 <hr /> <p>This is another paragraph</p></pre>
All elements must be nested properly	<pre><p>This is a bold paragraph</p></pre>	<pre><p>This is a bold paragraph</p></pre>

Tools for Creating HTML Documents

You can create Web pages using HTML with a simple text editor, such as Notepad++, Notepad, TextPad, or SimpleText. A **text editor** is a program that allows a user to enter, change, save, and print text, such as HTML. Text editors do not have many advanced features, but they do allow you to develop HTML documents easily. For instance, if you want to insert the DOCTYPE tags into the Web page file, type the necessary text into any of the text editors, as shown in Figure 1–10a and Figure 1–10b. Although Notepad (Figure 1–10b) is an adequate text editor for Web development, note its differences from Notepad++. Notepad++ is a more robust text editor that uses color schemes for HTML code as it is entered.

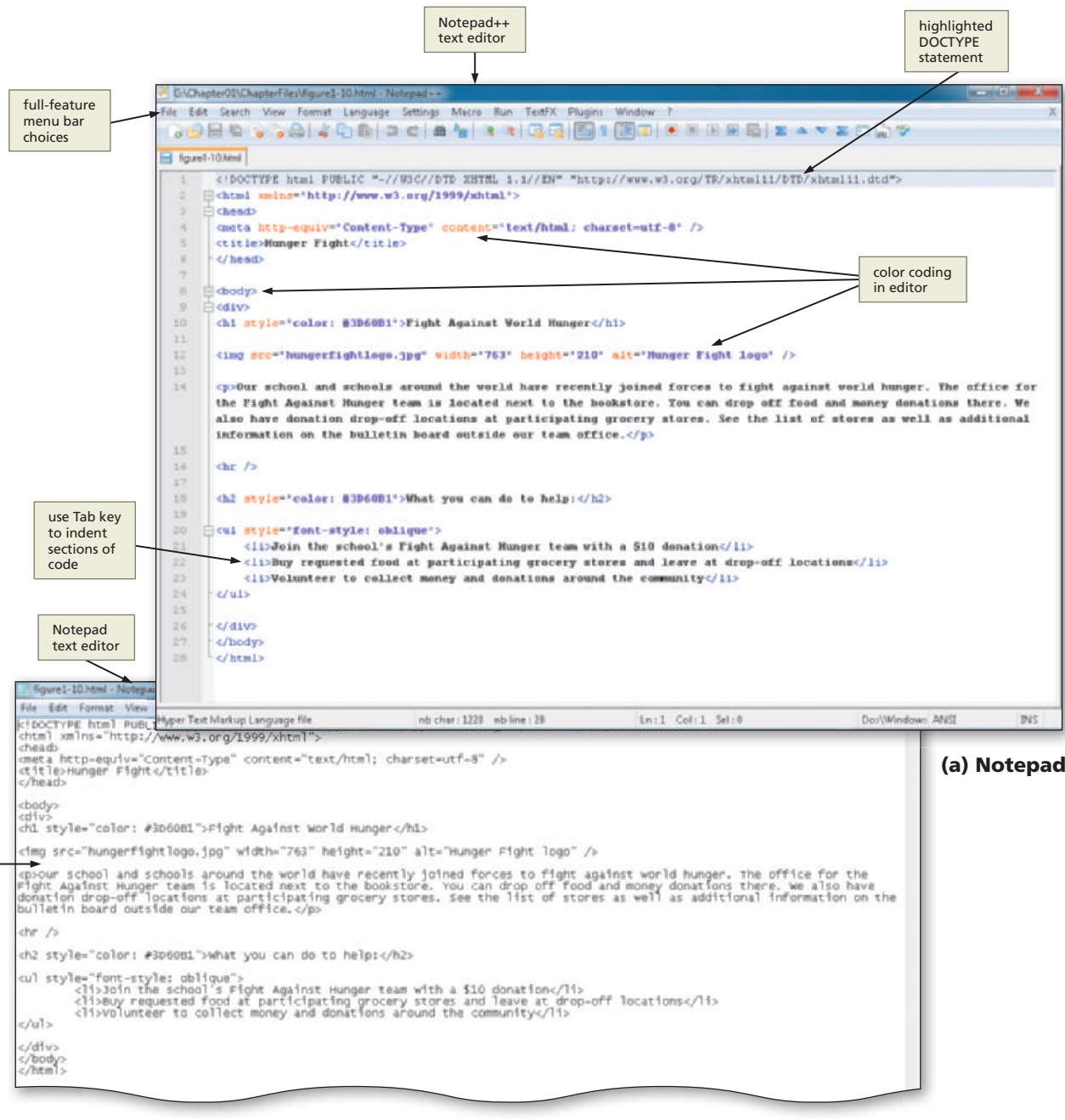


Figure 1-10 With a text editor such as Notepad++ or Notepad, you enter HTML tags and text that create a Web page.

You also can create Web pages using an HTML text editor, such as EditPlus or BBEdit (for Macintosh). An **HTML text editor** is a program that provides basic text-editing functions, as well as more advanced features such as color-coding for various HTML tags, menus to insert HTML tags, and spell checkers. An **HTML object editor**, such as Eiffel Software object editor, provides the additional functionality of an outline editor that allows you to expand and collapse HTML objects and properties, edit parameters, and view graphics attached to the expanded objects.

Many popular software applications also provide features that enable you to develop Web pages easily. Microsoft Word, Excel, and PowerPoint, for example, have a Save as Web Page feature that converts a document into an HTML file by automatically adding HTML tags to the document. Using Microsoft Access, you can create a Web page that allows you to view data in a database. Adobe Acrobat also has an export feature that creates HTML files. Each of these applications also allows you to add hyperlinks, drop-down boxes, option buttons, or scrolling text to the Web page.

These advanced Web features make it simple to save any document, spreadsheet, database, or presentation to display as a Web page. Corporate policy and procedures manuals and PowerPoint presentations, for example, easily can be saved as Web pages and published to the company's intranet. Extranet users can be given access to Web pages that allow them to view or update information stored in a database.

You also can create Web pages using a WYSIWYG editor such as Adobe Dreamweaver, Amaya, or CoffeeCup HTML Editor. A **WYSIWYG editor** is a program that provides a graphical user interface that allows a developer to preview the Web page during its development. WYSIWYG (pronounced wizzy-wig) is an acronym for What You See Is What You Get. A WYSIWYG editor creates the HTML code for you as you add elements to the Web page, which means that you do not have to enter HTML tags directly. The main problem with WYSIWYG editors is that they often create “puffed-up” HTML code (HTML tags with many lines of unnecessary additional code surrounding them).

Regardless of which type of program you use to create Web pages, it is important to understand the specifics of HTML so you can make changes outside of the editor. For instance, you may be able to create a Web page with Dreamweaver, but if you want to make some minor changes, it is very helpful to know the HTML tags themselves. It also is important to understand the Web development life cycle so the Web pages in your Web site are consistent and complete.

Web Development Life Cycle

For years, university and college information technology courses have stressed the importance of following the Systems Development Life Cycle when designing and implementing new software to ensure consistency and completeness. The Web development process should follow a similar cycle. Comprehensive planning and analysis ensure that developers will provide what the users want. If you start to code your Web pages without thorough planning and analysis, you run the risk of missing pertinent information. It is much less expensive to make corrections to a Web site in the early phases of project development than it is to alter Web pages that are completed.

The Web development life cycle outlined in this section is one that can be utilized for any type or size of Web development project. The **Web development life cycle** is a process that can be used for developing Web pages at any level of complexity. The Web development life cycle includes the following phases: planning, analysis, design and development, testing, and implementation and maintenance. Table 1–4 lists several questions that should be asked during each phase in the Web development life cycle. Throughout this book, you will follow this systematic cycle as you develop your Web pages.

Free HTML WYSIWYG Editors

There are a number of popular WYSIWYG editors that are being used by many novice Web developers to create well-designed, interactive Web sites. You can find these by searching for “WYSIWYG HTML editor” in most search engines.

Table 1–4 Web Development Phases and Questions

Web Development Phase	Questions to Ask
Planning	<ul style="list-style-type: none"> • What is the purpose of this Web site? • Who will use this Web site? • What are the users' computing environments? • Who owns and authors the information on the Web site? • Who decides if/where the information goes on the Web site?
Analysis	<ul style="list-style-type: none"> • What tasks do the users need to perform? • What information is useful to the users? • What process considerations must be made?
Design and Development	<ul style="list-style-type: none"> • How many Web pages will be included in the Web site? • How will the Web pages be organized? • What type of Web site structure is appropriate for the content? • How can I best present the content for ease of use? • What file naming convention will be employed for this Web site? • What folder structure will be used for the Web page files? • How do I apply XHTML standards throughout the development process? • What forms of multimedia contribute positively to the Web site? • How can accessibility issues be addressed without limiting usability? • Will there be an international audience?
Testing	<ul style="list-style-type: none"> • Do the Web pages pass the World Wide Web Consortium (W3C) validation process as XHTML compliant? • Is the Web site content correct? • Does the Web site function correctly? • Are users able to find the information they need to complete desired tasks? • Is navigation clear and easy to use?
Implementation and Maintenance	<ul style="list-style-type: none"> • How is the Web site published? • How can users be attracted to visit and revisit the Web site? • How is the Web site updated? • Who is responsible for content updates? • Who is responsible for structure updates? • How will users be notified about updates to the Web site? • Will the Web site be monitored?

Web Site Planning

Web site planning, which is the first phase of the Web development life cycle, involves identifying the goals or purpose of the Web site. The first step in the Web site planning phase is to answer the question “What is the purpose of this Web site?” As you have learned, individuals and groups design and publish Web sites for a variety of purposes. Individuals develop Web sites to share their hobbies, to post résumés, or just to share ideas on personal interests. Organizations create Web sites to keep members informed of upcoming events or to recruit new members. Businesses create Web sites to advertise and sell products or to give their customers 24-hour online support. Instructors publish Web sites, or add information to their courses using the school’s online course management software, to inform students of course policies, assignments, and due dates, as well as course requirements. Until you adequately can identify the intended purpose of the Web site, you should not proceed with the Web development project.

In addition to understanding the Web site’s purpose, you also should understand who will use the Web site and the computing environments of most of the users. Knowing the makeup of your target audience — including age, gender, general demographic background, and level of computer literacy — will help you design a Web site appropriate

for all users. Understanding users' computing environments will determine what types of Web technologies to use. For example, if most users have low-speed Internet connections, you would not want to create pages with large graphics or multimedia elements.

A final aspect to the Web site planning phase is to identify the content owners and authors. To determine this, you need to ask the questions:

- Who owns and authors the information on the Web site?
- Who decides if/where the information goes on the Web site?

Once you have identified who will provide and authorize the Web site content, you can include those individuals in all aspects of the Web development project.

Web Site Analysis

During the analysis phase, you make decisions about the Web site content and functionality. To help define the appropriate Web site content and functionality, you first should identify the tasks that users need to perform. Answering that question allows you to define necessary content to facilitate those tasks and determine useful information for the users. Extraneous content should be eliminated from the Web site because it does not serve any purpose.

In the analysis phase, it also is important to consider the processes required to support Web site features. For example, if you determine that users should be able to order products through the Web site, then you also need to define the processes or actions to be taken each time an order is submitted. For instance, after an order is submitted, how will that order be processed throughout the back-office business applications such as inventory control and accounts payable? Will users receive e-mail confirmations with details about their orders? The analysis phase is one of the more important phases in the Web development life cycle. Clearly understanding and defining the desired content and functionality of the Web site will direct the type of Web site that you design and reduce changes during Web site development.

BTW

Accessibility Standards
Creating a Web site that is accessible to all users allows your Web site to reach the widest audience. Further, under Section 508 law, any Web site or technology used by a U.S. federal agency must be usable by people with disabilities. See Appendix C for Section 508 guidelines.

Web Site Design and Development

After determining the purpose of the Web site and defining the content and functionality, you need to consider the Web site's design. Some key considerations in Web site design are defining how to organize Web page content, selecting the appropriate Web site structure, determining how to use multimedia, addressing accessibility issues, and designing pages for an international audience. One of the most important aspects of Web site design is determining the best way to provide navigation on the Web site. If users cannot easily find the information that they are seeking, they will not return to your Web site.

Many ways to organize a Web page exist, just as many ways to organize a report or paper exist. Table 1–5 lists some organizational standards for creating a Web page that is easy to read and navigate.

Table 1–5 Web Page Organizational Standards

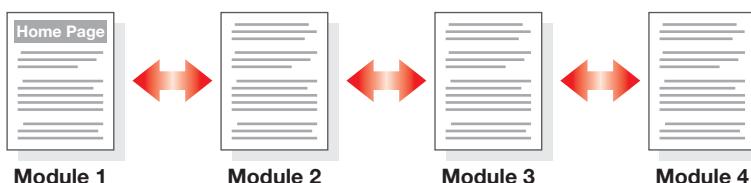
Element	Organizational Standard	Reason
Titles	Use simple titles that clearly explain the purpose of the page	Titles help users understand the purpose of the page; a good title explains the page in the search engine results lists
Headings	Use headings to separate main topics	Headings make a Web page easier to read; simple headlines clearly explain the purpose of the page

Table 1–5 Web Page Organizational Standards (continued)

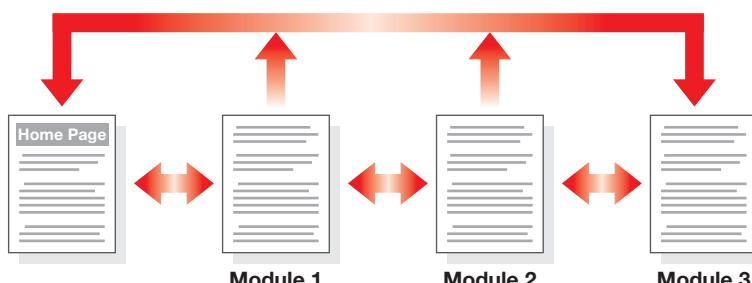
Element	Organizational Standard	Reason
Horizontal Rules	Insert horizontal rules to separate main topics	Horizontal rules provide graphical elements to break up Web page content
Paragraphs	Use paragraphs to help divide large amounts of text	Paragraphs provide shorter, more readable sections of text
Lists	Utilize bulleted or numbered lists when appropriate	Lists provide organized, easy-to-read text that readers can scan
Page Length	Maintain suitable Web page lengths	Web users do not always scroll to view information on longer pages; appropriate page lengths increase the likelihood that users will view key information
Information	Emphasize the most important information by placing it at the top of a Web page	Web users are quick to peruse a page; placing critical information at the top of the page increases the likelihood that users will view key information
Other	Incorporate a contact e-mail address; include the date of the last modification	E-mail addresses and dates give users a way to contact a Web site developer with questions; the date last modified helps users determine the timeliness of the site information

Web sites can use several different types of structures, including linear, hierarchical, and webbed. Each structure links, or connects, the Web pages in a different way to define how users navigate the site and view the Web pages. You should select a structure for the Web site based on how users will navigate the site and view the Web site content.

A **linear** Web site structure connects Web pages in a straight line, as shown in Figure 1–11. A linear Web site structure is appropriate if the information on the Web pages should be read in a specific order. For example, if the information on the first Web page, Module 1, is necessary for understanding information on the second Web page, Module 2, you should use a linear structure. Each page would have links from one Web page to the next, as well as a link back to the previous Web page. There are many cases in which Web pages need to be read one after the other.

**Figure 1–11 Linear Web site structure.**

A variation of a linear Web site structure includes the addition of a link to the home page of the Web site, as shown in Figure 1–12. For some Web sites, moving from one

**Figure 1–12 Linear Web site structure with links to home page.**

BTW

User Interface Design

The user interface design is an important aspect of a Web site. If a site is designed poorly, users may be unable to find the desired information or complete a task, which makes the Web site ineffective.

module to the next module is still important, but you also want to provide users with easy access to the home page at any time. In this case, you would still provide links from the module Web pages to the previous and next module, but each Web page would also have a link back to the home page. In this way, the user does not have to click the previous link multiple times in order to get back to the home page.

A **hierarchical** Web site structure connects Web pages in a treelike structure, as shown in Figure 1–13. A hierarchical Web site structure works well on a site with a main index or table of contents page that links to all other Web pages. With this structure, the main index page would display general information, and secondary pages would include more detailed information. Notice how logically the information in Figure 1–13 is organized. A Web page visitor can easily go from the home page to any of the three modules. In addition, the visitor can easily get to the Module 3 Quiz by way of the Module 3 link. One of the inherent problems with this structure, though, is the inability to move easily from one section of pages to another. As an example, to move from Module 1 Page 2 to the Module 3 Summary, the visitor would have to use the Back button to get to the Home Page and then click the Module 3 link. This is moderately annoying for a site with two Web pages, but think what it would be like if Module 1 had 100 Web pages!

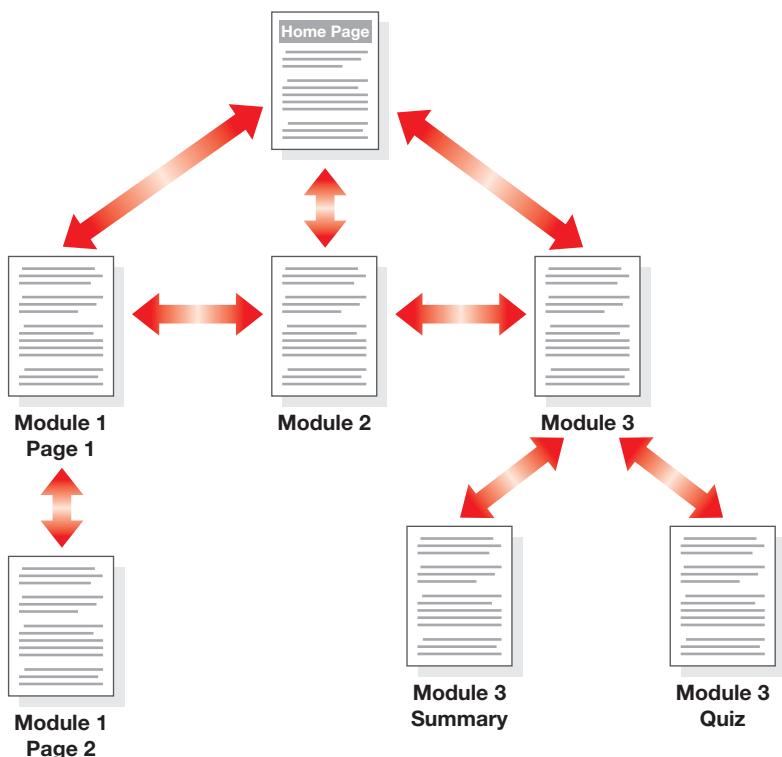


Figure 1–13 Hierarchical Web site structure.

To circumvent the problems with the hierarchical model, you can use a **webbed** model (Figure 1–14). A **webbed** Web site structure has no set organization, as shown in Figure 1–14. A webbed Web site structure works best on sites with information that does not need to be read in a specific order and with many navigation options. The World Wide Web uses a webbed structure, so users can navigate among Web pages in any order they choose. Notice how the Web site visitor can more easily move between modules or module summaries with this structure. With this model, you most often provide a link to the Home Page from each page, resulting in an additional arrow going from each individual Web page back to the home page (which is difficult to depict in these small figures). Many Web sites today utilize a graphical image (usually the company or institutional logo) in the top-left corner of each Web page as a link to the home page. You will use that technique later in the book.

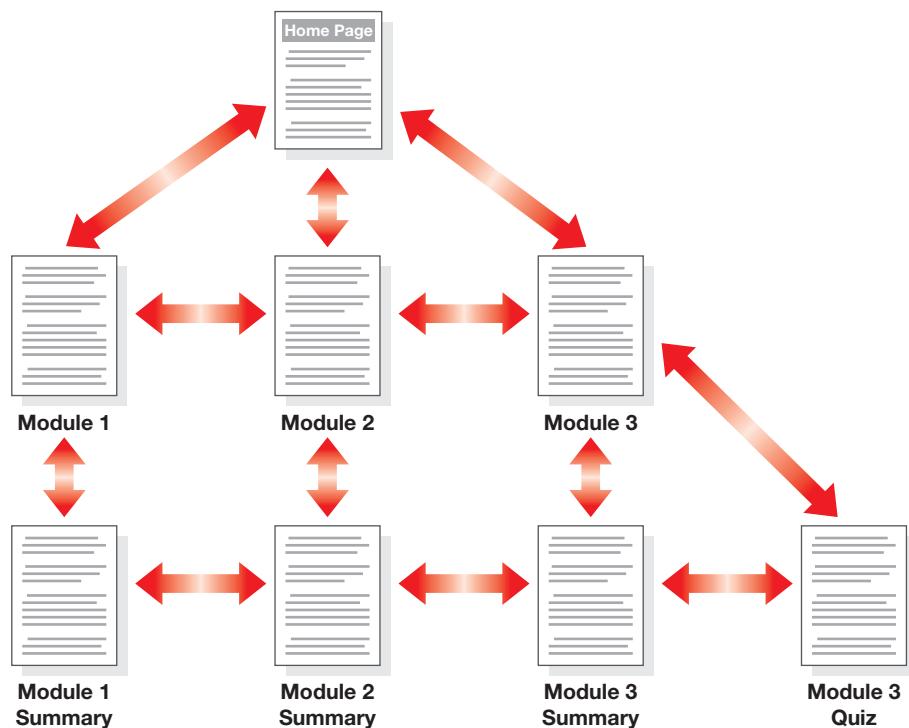


Figure 1-14 Webbed Web site structure.

Most Web sites are a combination of the linear, hierarchical, and webbed structures. Some information on the Web site might be organized hierarchically from an index page, other information might be accessible from all areas of the site, and still other information might be organized linearly to be read in a specific order. Using a combination of the three structures is appropriate if it helps users navigate the site easily. The key is to get the right information to the users in the most efficient way possible.

Regardless of the structure or structures that you use, you should balance the narrowness and depth of the Web site. A **broad Web site** is one in which the home page is the main index page, and all other Web pages are linked individually to the home page (Figure 1-15).

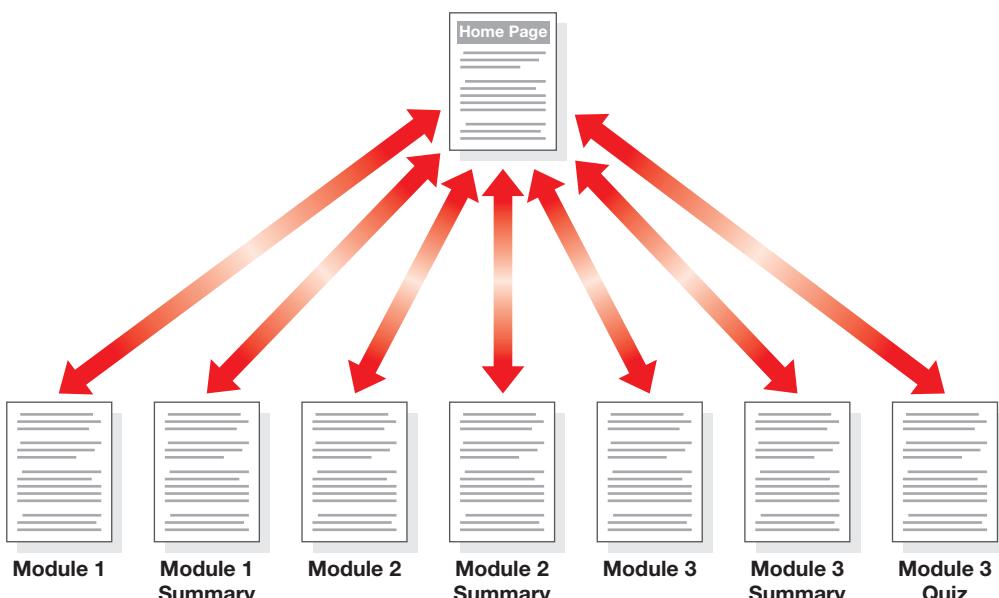


Figure 1-15 Broad Web site.

By making the other Web pages accessible only through the home page, a broad Web site forces the user to return to the home page to move from one Web page to another. The structure makes navigation time-consuming and limiting for users. A better structure would present a user with navigation alternatives that allow for direct movement between Web pages.

A **deep Web site** is one that has many levels of pages, requiring the user to click many times to reach a particular Web page (Figure 1–16). By requiring a visitor to move through several Web pages before reaching the desired page, a deep Web site forces a user to spend time viewing interim pages that may not have useful content. As an example, note the difference between finding the Module 3 Summary in Figure 1–13 on page HTML 18

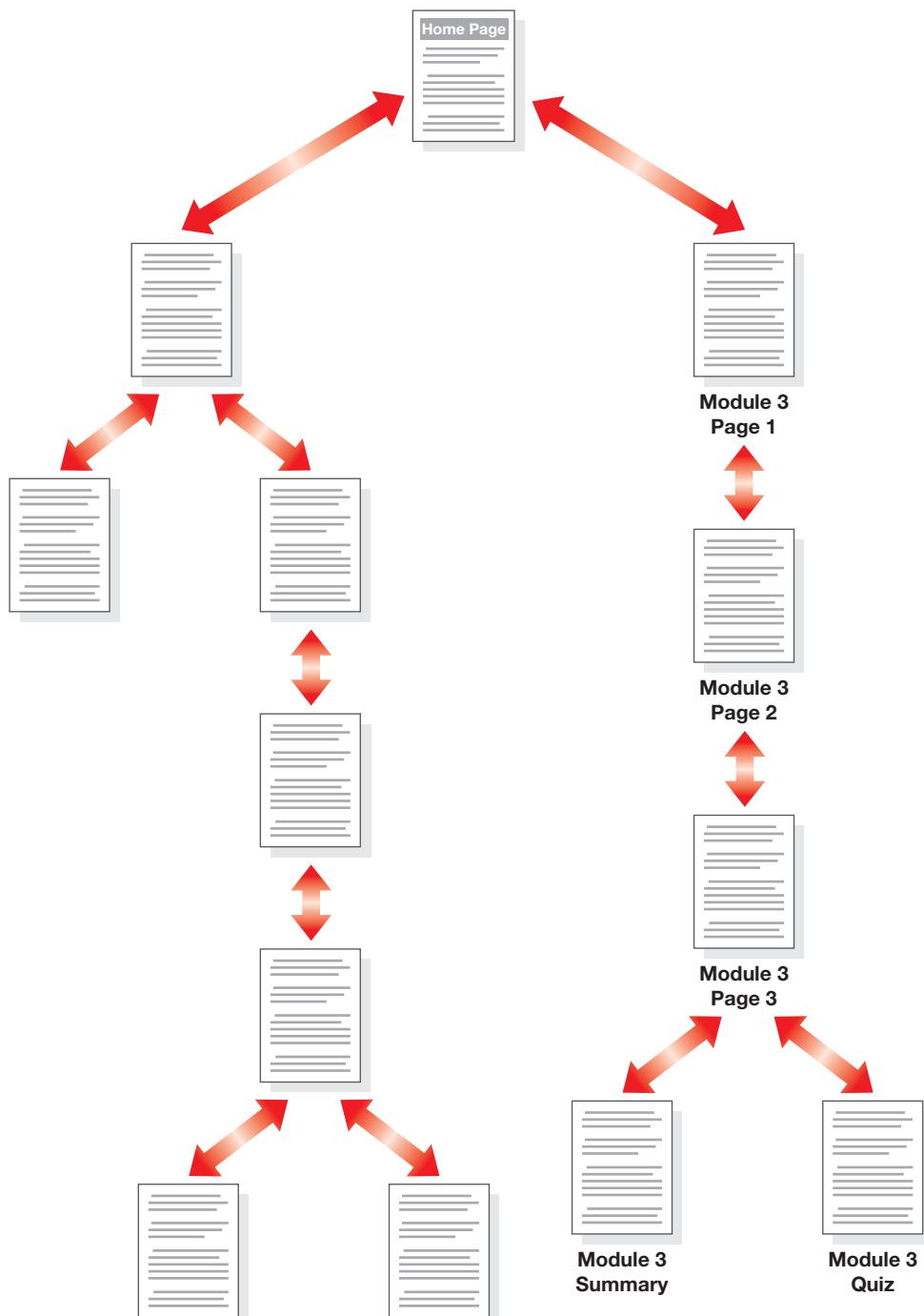


Figure 1–16 Deep Web site.

as compared to finding the same Web page (Module 3 Summary) in Figure 1–16. Assume that the user went through the Figure 1–13 Web site once to study the Module 3 material. When the user returns to the Web site using the Figure 1–16 structure, however, to review the Module 3 Summary Web page and then take the Module 3 Quiz, the user would have to go completely through the Module 3 material, Web page by Web page, in order to get to the Module 3 Summary page. You probably want to give users easier access to that Web page.

As a Web developer, you must select an appropriate structure for the Web site and work to balance breadth and depth. Users go to a Web site looking for information to complete a task. Good design provides ease of navigation, allowing users to find content quickly and easily. In addition to planning the design of the Web site itself, a Web developer should always plan the specifics of the file naming and storage conventions early on in the design phase. Once you determine the structure of the Web site and the approximate number of pages necessary to fulfill the site purpose, then you need to identify what standards to use with file naming and the folder structure. For instance, saving your Web pages with names such as page1.html and page2.html does not tell you the purpose of those Web pages. A better option would be to name the Sabatina Pizza Web site's home page sabatinahome.html or sabatinas.html, and the Web page with the user order form could be named orderform.html. Those file names tell the developer, as well as future developers maintaining the Web site, the purpose of those Web pages.

The same principle applies to the folder structure that you use in your Web development. The projects in this book have so few Web page files and graphic files that all content (Web pages and graphics) is stored together in one folder. With a large Web site, however, you may want to put the Web page files in a separate folder from the graphics files. Larger, more complex Web sites might also require a folder just to store video or audio clips. Where you store the files will affect how you access those files in your HTML code. Determining a good folder structure in the planning phase of the Web development life cycle is important. You'll learn more about effective folder structures in Chapter 3.

During the design and development phase, you also should consider what, if any, types of multimedia could contribute positively to the Web site experience. For instance, adding a video message from the company CEO might be useful, but if the computing environment of your users cannot accommodate video playback, then the video serves no purpose. In general, do not use advanced multimedia technologies in a Web site unless they make a positive contribution to the Web site experience. Today, more Web sites are using audio and video content. These new additions sometimes enhance the overall purpose of the Web site, but they sometimes detract from the message.

Finally, consider accessibility issues and internationalization. A Web developer should always design for viewing by a diverse audience, including physically impaired and global users. A key consideration is that the software used by physically impaired individuals does not work with some Web features. For instance, if you use graphics on the Web site, always include alternative text for each graphic. To support an international audience, use generic icons that can be understood globally, avoid slang expressions in the content, and build simple pages that load quickly over lower-speed connections.

The design issues just discussed are only a few of the basic Web page design issues that you need to consider. Throughout this book, design issues will be addressed as they relate to each project. Many excellent Web page design resources also are available on the Internet.

Once the design of the Web site is determined, Web development can begin. The rest of the chapters in this book discuss good Web page standards, in addition to the actual development of Web pages. You will learn many development techniques, including links, tables, graphics, image maps, and Web forms. The umbrella that covers all of the development techniques taught in this book is the use of Cascading Style Sheets (CSS).

BTW**Web Page Structure**

There are many resources available on the Web that further discuss Web site structures. In addition to general design information, there are a number of tools available for sale or free download that can help you design your Web sites. Enter the phrase "Web site structure" into a search engine to find many valuable design sources.

Web Site Testing

Testing should be done on all pages in a Web site. You should also test the links within the Web page, to other Web pages in the Web site, and to external Web sites. Testing is an important part of Web development and assures that your Web pages work as intended.

Web Site Testing

A Web site should be tested at various stages of the Web design and development processes. The testing process should be comprehensive and include a review of Web page content, functionality, and usability. Web sites with broken links, missing graphics, and incorrect content create a poor impression. You want to attract users to your Web site and maintain their interest. If visitors find that your Web site is poorly tested and maintained, they will be less likely to return. You cannot get your message out if users don't frequently visit the Web site. Some basic steps to test content and functionality include:

- Validating each Web page by running it through the W3C markup validation service
- Proofreading page content and titles to review for accurate spelling and grammar
- Checking links to ensure they are not broken and are linked correctly
- Checking graphics to confirm they appear properly and are linked correctly
- Ensuring that accessibility and internationalization issues are addressed
- Testing forms and other interactive page elements
- Testing pages to make sure they load quickly, even over lower-speed connections
- Printing each page to check how printed pages look

Usability is the measure of how well a product, such as a Web site, allows a user to accomplish his or her goals. **Usability testing** is a method by which users of a Web site or other product are asked to perform certain tasks in an effort to measure the product's ease-of-use and the user's perception of the experience. Usability testing for a Web site should focus on three key aspects: content, navigation, and presentation.

Usability testing can be conducted in several ways; one effective way is to directly observe users interfacing with (or using) the Web site. As you observe users, you can track the links they click and record their actions and comments. You even can ask the users to explain what tasks they were trying to accomplish while navigating the site. The information gained by observing users can be invaluable in helping identify potential problem areas in the Web site. For example, if you observe that users have difficulty finding the Web page that lists store locations and hours of operation, you may want to clarify the link descriptions or make the links more prominent on the home page.

Another way to conduct usability testing is to give users a specific task to complete (such as finding a product price list) and then observe how they navigate the site to complete the task. If possible, ask them to explain why they selected certain links. Both of these observation methods are extremely valuable, but require access to users.

Usability testing also can be completed using a questionnaire or survey. When writing a questionnaire or survey, be sure to write open-ended questions that can give you valuable information. For instance, asking the yes/no question "Is the Web site visually appealing?" will not gather useful information. If you change that question to use a scaled response, such as, "Rate the visual appeal of this Web site, using a scale of 1 for low and 5 for high," you can get more valuable input from the users. Make sure, however, that the scale itself is clear and understandable to the users. If you intend that a selection of 1 equates to a "low" rating, but the users think a 1 means "high," then your survey results are questionable. A usability testing questionnaire always should include space for users to write additional explanatory comments.

Figure 1–17 shows some examples of types of questions and organization that you might include in a Web site usability testing questionnaire.

In addition to content, functionality, and usability testing, there are other types of testing. For a newly implemented or maintained Web site, two other types of tests should be conducted: compatibility testing and stress testing. **Compatibility testing** is done to verify that the Web site works with a variety of browsers and browser versions. Initially, test using the browsers that your audience is most likely to use. Different browsers display some

Web Site Usability Questionnaire

The purpose of this questionnaire is to gather information regarding the usability of our company Web site.

Name (not required): _____

Your relationship to the company (circle your choice): customer employee no relationship

How easy was it to use our Web site navigation to find the following information? (circle your choice)

<i>Contact information</i>	very easy	somewhat easy	moderately difficult	very difficult
<i>Store hours/locations</i>	very easy	somewhat easy	moderately difficult	very difficult
<i>Product information</i>	very easy	somewhat easy	moderately difficult	very difficult
<i>FAQs</i>	very easy	somewhat easy	moderately difficult	very difficult

How would you rate our Web site? (circle one)

Browsing Experience	Strongly Agree	Agree	Neither Agree nor Disagree	Disagree	Strongly Disagree
The Web pages loaded in an acceptable time frame	5	4	3	2	1
The Web site was visually appealing	5	4	3	2	1
The navigation was clear and made it easy to find information	5	4	3	2	1
The graphics were used to convey pertinent information	5	4	3	2	1
I found what I was looking for on this Web site	5	4	3	2	1
My experience with this Web site was successful	5	4	3	2	1

Please use the area below to give us additional feedback. Our company goal is to provide an appealing and enjoyable Web browsing experience. We therefore are interested to hear your comments about our Web site.

area for additional comments

range of choices for answers

Figure 1–17 Web site usability testing questionnaire.

aspects of Web pages differently, so it is important to test Web pages in several different browsers to verify they appear correctly in each browser. If you have used technologies that are not supported by older browsers or that require plug-ins, consider changing the content or providing alternative Web pages for viewing in older browsers. If your audience uses both PC and Macintosh computers, you need to test the Web pages using browsers on both platforms. You also may want to test the Web pages in several versions of the same browser (usually the two most recent versions), in the event users have not yet upgraded.

Stress testing determines what happens on your Web site when greater numbers of users access the site. A Web site with 10 users accessing it simultaneously may be fine. When 100 users use the Web site at once, it may operate at an unacceptably slow speed. Stress testing verifies that a Web site runs at an acceptable speed with many users. There are many cases in which companies did not effectively stress test their Web sites. The results of this lack of testing have been disastrous, with Web sites locking up when too many users tried to access the same Web site function. Especially in the case of Web sites used for e-commerce, it is imperative for the Web site to stay online. A crashed or locked-up Web site will not sell products or services, and the company stands to lose a lot of money.

Web Site Implementation and Maintenance

Once Web site testing is complete and any required changes have been made, the Web site can be implemented. Implementation of a Web site involves the actual publishing of the Web pages to a Web server. Many HTML editors and WYSIWYG editors provide publishing capabilities. You also can use FTP software, such as WS_FTP or CuteFTP, to publish your Web pages to a Web server. After you publish a Web site, you should test the Web pages again to confirm no obvious errors exist such as broken links or missing graphics.

After a site is tested and implemented, you need to develop a process to maintain the Web site; users will undoubtedly request changes and timely content will require updates. You need to ensure, however, that updates to the Web site do not compromise the site's integrity and consistency. For example, if you have several different people updating various Web pages on a large Web site, you might find it difficult to maintain a consistent look on pages across the Web site. You should plan to update your Web site on a regular basis to keep content up-to-date. This could mean hourly, daily, weekly, or less often, depending on the site's purpose. Do not allow your content to become stale, outdated, or include broken links to Web pages that no longer exist. As a user looking for information related to a specific topic, how likely are you to believe the information found on a Web site that says "Last update on December 10, 1998" comes from a reliable source?

To help manage the task of Web site maintenance, first determine who is responsible for updates to content, structure, functionality, and so on. Then, limit update responsibilities to specific users. Be sure the implementation is controlled by one or more Web developers who can verify that the Web pages are tested thoroughly before they are published.

As updates and changes are made to a Web site, consider notifying users with a graphic banner or a "What's New" announcement, explaining any new features and how the features will benefit them. This technique not only keeps users informed, but also encourages them to come back to the Web site to see what is new.

Finally, Web site monitoring is another key aspect of maintaining a Web site. Usually, the Web servers that host Web sites keep logs of information about Web site usage. A **log** is the file that lists all of the Web pages that have been requested from the Web site. Web site logs are an invaluable source of information for a Web developer. Obtaining and analyzing the logs allow you to determine such things as the number of visitors, browser types and versions, connection speeds, pages most commonly requested, and usage patterns. With this information, you can design a Web site that is effective for your targeted audience, providing visitors with a rich and rewarding experience.

BTW Quick Reference

For a list of HTML tags and their associated attributes, see the HTML Quick Reference (Appendix A) at the back of this book, or visit the HTML Quick Reference Web page (scsite.com/HTML6e/qr).

Be an Observant Web User

As you embark on this course, and perhaps start your Web development career, one useful practice is to be an observant Web user. Most of us use the Web several times a day (or more often) to complete our daily tasks. As a Web developer, you should review the Web pages that you access with an eye on functionality and design. As described in the first In the Lab exercise at the end of the chapter, you can bookmark Web sites you think are effective and ineffective, good and bad, and use them as references for your own Web development efforts. Watch for trends on the Web as you search for information or make online purchases. For example, running banners used to be very popular on the Web, but now other design techniques have taken over. Being an observant Web user can help you become a more effective Web developer.

Chapter Summary

In this chapter, you have learned about the Internet, the World Wide Web, and associated technologies, including Web servers and Web browsers. You learned the essential role of HTML in creating Web pages and reviewed tools used to create HTML documents. You also learned that most Web development projects follow a five-phase life cycle. The items listed below include all the new concepts you have learned in this chapter.

1. Describe the Internet (HTML 2)
2. Describe the World Wide Web (HTML 3)
3. Define Web servers (HTML 4)
4. Describe the Internet, intranets, and extranets (HTML 5)
5. Discuss Web browsers (HTML 7)
6. Define Hypertext Markup Language (HTML 8)
7. Describe HTML elements (HTML 9)
8. List HTML coding practices (HTML 10)
9. Explain HTML versions (HTML 11)
10. Describe Cascading Style Sheets (HTML 11)
11. Define the Document Object Model (HTML 11)
12. Define Extensible Hypertext Markup Language (XHTML) (HTML 12)
13. Describe tools for creating HTML documents (HTML 13)
14. Discuss the Web development life cycle (HTML 14)
15. Describe steps in the Web development planning phase (HTML 15)
16. Explain the Web development analysis phase (HTML 16)
17. Discuss Web design and development (HTML 16)
18. Describe various Web site structures (HTML 17)
19. Discuss the importance of Web site testing, including usability testing, compatibility testing, and stress testing (HTML 22)
20. Discuss Web site implementation and maintenance (HTML 24)
21. Explain the importance of being an observant Web user (HTML 24)

Learn It Online

Test your knowledge of chapter content and key terms.

Instructions: To complete the Learn It Online exercises, start your browser, click the Address bar, and then enter the Web address scsite.com/html6e/learn. When the HTML Learn It Online page is displayed, click the link for the exercise you want to complete and read the instructions.

Chapter Reinforcement TF, MC, and SA

A series of true/false, multiple choice, and short answer questions that test your knowledge of the chapter content.

Flash Cards

An interactive learning environment where you identify chapter key terms associated with displayed definitions.

Practice Test

A series of multiple choice questions that test your knowledge of chapter content and key terms.

Who Wants to Be a Computer Genius?

An interactive game that challenges your knowledge of chapter content in the style of a television quiz show.

Wheel of Terms

An interactive game that challenges your knowledge of chapter key terms in the style of the television show *Wheel of Fortune*.

Crossword Puzzle Challenge

A crossword puzzle that challenges your knowledge of key terms presented in the chapter.

Apply Your Knowledge

Reinforce the skills and apply the concepts you learned in this chapter.

Understanding Web Page Structures

Instructions: Figure 1–18 shows the Web site of a popular retailer, Costco. As you learned in this chapter, three common Web site structures include linear, hierarchical, and webbed. Based on that information, determine the structure used in the Costco.com Web site. Review other similar Web sites and determine which Web site design features are beneficial to a user. Incorporate those ideas into a new Web site design for Costco.com. Use paper to sketch the new Web site design for the Costco.com Web site.

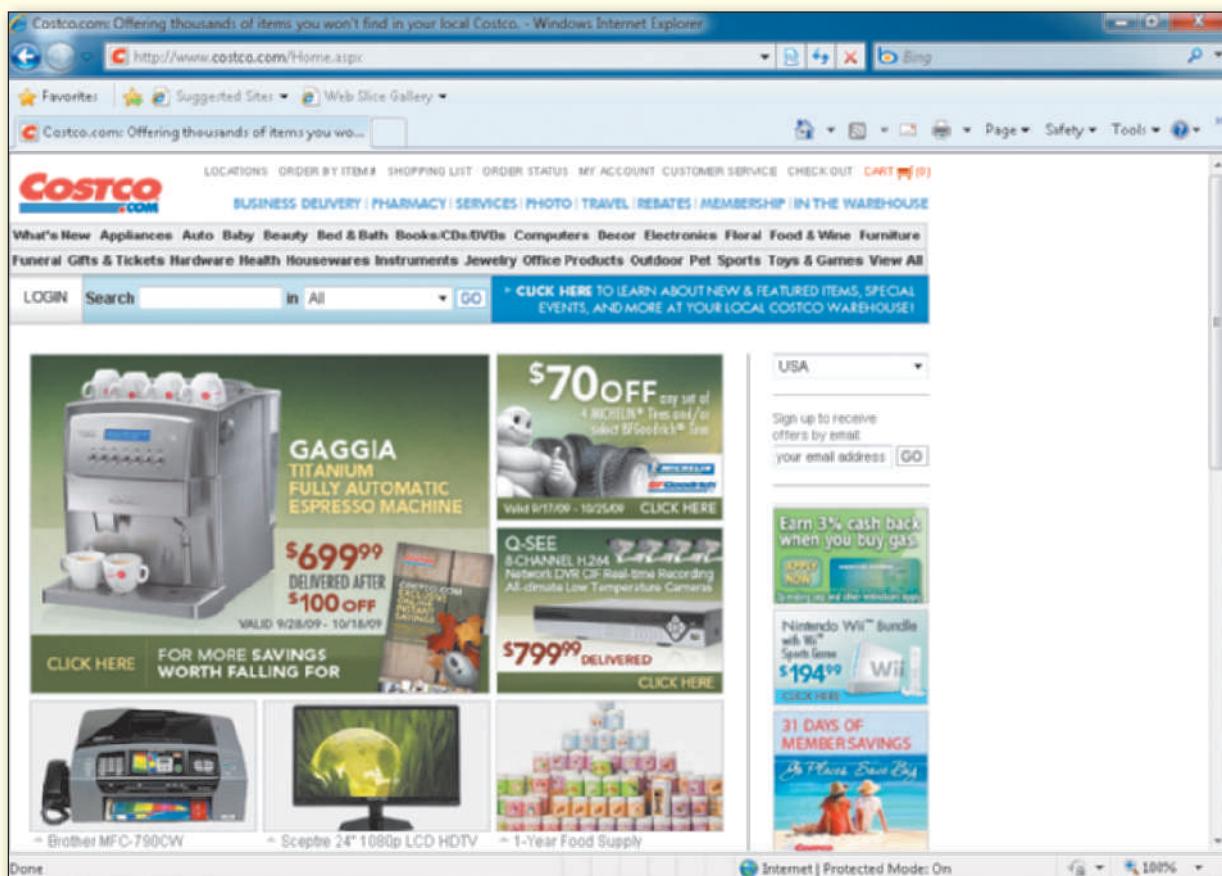


Figure 1–18

Perform the following tasks:

1. Start your browser. Open the Costco.com Web site in your browser. Print the home page by clicking Print on the File menu or by clicking the Print icon.
2. Explore the Costco.com Web site, determine the structure that the Web site utilizes (linear, hierarchical, or webbed), and then write that on the printout.
3. Find two other online retail store Web sites. Print the home pages for each of those sites. Navigate these Web sites to identify any design features that are beneficial to a user.
4. Using ideas from the online retail Web sites that you found in Step 3, sketch a new Web site structure and design for the Costco.com site on paper.
5. Write your name on the printouts and the sketch and hand them in to your instructor.

Extend Your Knowledge

Extend the skills you learned in this chapter and experiment with new skills.

Evaluating a User Survey

Instructions: Start your word-processing program. Open the document extend1-1.doc from the Chapter01\Extend folder of the Data Files for Students. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files. This sample Web site survey shows various questions that could be asked in gathering feedback on Web site usability. It is important to assess the usability of your Web site, as mentioned in the chapter.

You will evaluate the user survey and modify the questions or add new questions that apply to the Web site that you have chosen. You then will ask five people to take your survey.

Perform the following tasks:

1. Connect to the Internet and identify one Web site that you think is confusing, difficult to use, or unattractive.
2. Make changes to the user survey by following some of the guidelines provided in Figure 1–17 on page HTML 20 and save the file as extend1-1solution.doc. Add questions to the survey that will help you determine a user’s opinion of the selected Web site. Remember that the purpose of using surveys is to improve a Web site. Your questions therefore have to provide you with information that can help you achieve that goal.
3. Print five copies of your edited survey and distribute those printed copies to at least five family members or friends. Have them complete the survey relative to the Web site that you chose in Step 1. Collect their survey responses.
4. Determine what you learned from the results of the surveys. How helpful was the survey in determining needed improvements to the selected Web site? Were there other questions or other types of questions that you could ask that could have provided more valuable information about the Web site?
5. Identify what you can do to improve your Web site survey. Using a word processor, type your analysis at the bottom of your extend1-1solution.doc document, and then submit it in the format specified by your instructor.

Make It Right

Analyze a document and correct all errors and/or improve the design.

Correcting the Web Site Type Table

Instructions: Start your word-processing program. Open the file makeitright1-1.doc from the Chapter01\MakeItRight folder of the Data Files for Students. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files. The document, shown in Table 1–7 on the next page, is a modified version of Table 1–5 (on page HTML 17). The table, which intentionally contains errors, lists the Web page organizational standards discussed in Chapter 1. Without referring to Table 1–5, make the necessary corrections to Table 1–7 by identifying the correct organizational standard and reason for each of the seven elements listed. Save the revised document as makeitright1-1solution.doc and then submit it in the form specified by your instructor.

Continued >

Make It Right *continued***Table 1–7 Types of Web Sites**

Element	Organizational Standard	Reason
Titles	Use these to separate main topics	These provide graphical elements to break up Web page content
Headings	Use simple ones that clearly explain the purpose of the page	These provide shorter, more-readable sections of text
Horizontal Rules	Utilize these in bulleted or numbered format when appropriate	Web users do not always scroll to view information on longer pages; appropriate page lengths increase the likelihood that users will view key information
Paragraphs	Maintain suitable Web page lengths	Web users are quick to peruse a page; placing critical information at the top of the page increases the likelihood that users will view key information
Lists	Insert these graphical elements to separate main topics	These provide organized, easy-to-read text that readers can scan
Page Length	Use these to help divide large amounts of text	Titles help users understand the purpose of the page; a good title explains the page in the search engine results lists
Information	Emphasize the most important information by placing it at the top of a Web page	These make a Web page easier to read; simple headlines clearly explain the purpose of the page

In the Lab

Design and/or create a document using the guidelines, concepts, and skills presented in this chapter. Labs are listed in order of increasing difficulty.

Lab 1: Evaluating Web Sites

Problem: In Chapter 1, you learned the importance of being an observant Web user, which can help you become a more effective Web developer. To further develop that concept, find and then discuss “good” and “bad” (“effective” and “ineffective”) Web sites. Start your browser and your word-processing program. Open the file lab1-1.doc from the Chapter01\IntheLab folder of the Data Files for Students. See the inside back cover of this book for instructions for downloading the Data Files for Students, or contact your instructor for information on accessing the required files for this book.

Instructions: Perform the following steps using your browser and the file listed.

1. Browse the Internet and find one “good” (i.e., effective) or one “bad” (i.e., ineffective) Web site. Determine, based on your own opinion, what is “good” and what is “bad” in these Web sites. You will identify the specific reason for your opinion in Step 2 below.
2. Using the lab1-1.doc file, rate the usability of the good and bad Web sites that you selected. Be sure to add additional comments in the survey to specifically identify your positive or negative feelings about the Web site. Save the documents using the file names lab1-1goodsolution.doc and lab1-1badsolution.doc.
3. Team up with one other student and discuss your survey results while reviewing the Web sites that you selected. Also review your student partner’s Web site and surveys.
4. Open the word-processing document named lab1-1comparison.doc and note any differences of opinion in your survey results and the opinion of your student partner. Make sure to include the

URLs of the four Web sites that you and your partner reviewed in this new document. Save the document using the file name lab1-1comparison.doc.

- Submit your own solutions (lab1-1goodsolution.doc and lab1-1badsolution.doc) and the team document (lab1-1comparison.doc) in the format specified by your instructor.

In the Lab

Lab 2: Designing a Web Site for the School Counselors

Problem: Your school's counseling department wants you to design a Web site to link to/from the school's main Web site. To do this, you must complete the planning and analysis phases by answering such questions as:

- What tasks do students want to complete on the Web site?
- What tasks will the school counselors want to complete on the Web site?
- What types of information should be included?
- Who will provide information on the Web site content?

Interview several students of the school and determine the answers to these questions. Based on that information, you will draw a sketch of a design for the home page of the counseling department's Web site, such as the design shown in Figure 1–19.

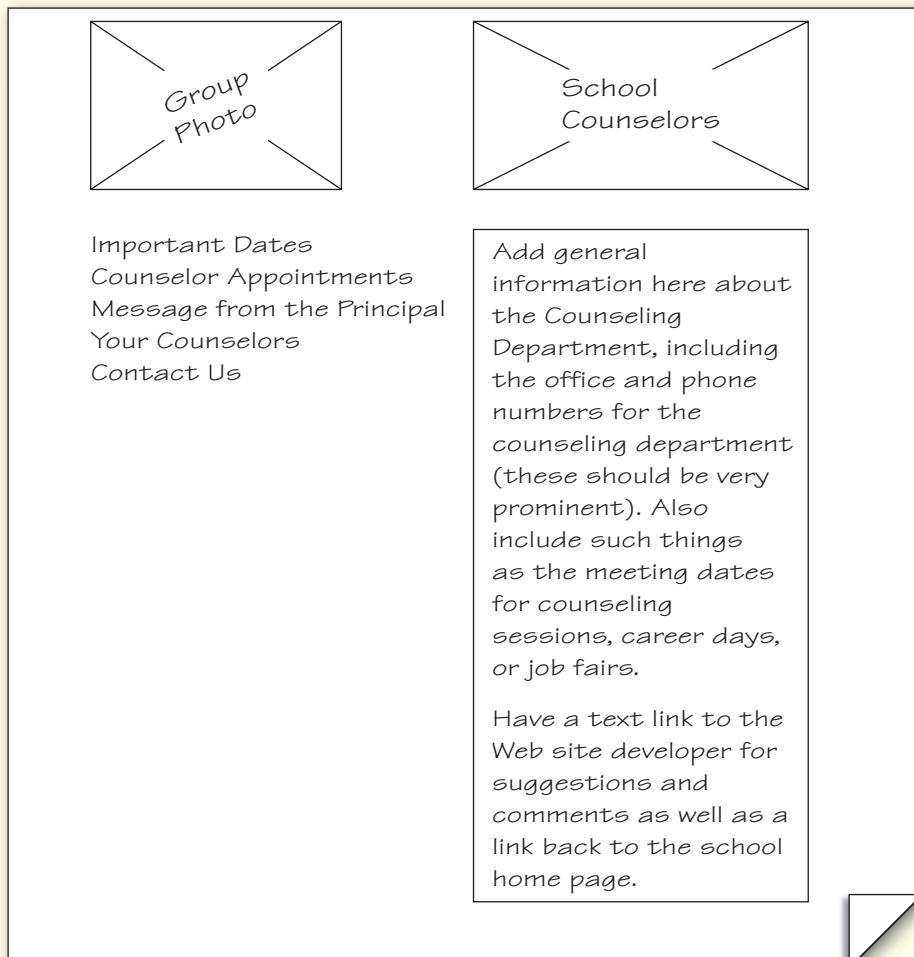


Figure 1–19

Continued >

In the Lab *continued*

Instructions: Perform the following tasks using your word-processing program and paper.

1. Review the questions in the planning and analysis phases of the Web development life cycle, as shown in Table 1–4 on page HTML 15.
2. Assess the value of those questions listed in the table. Add other questions that you think are relevant to the planning and analysis of a counseling department Web site.
3. Start your word-processing program. If necessary, open a new document. Enter the questions you will use for planning and analysis. Save the document using the file name lab1-2solution.doc. Print the document.
4. Using the questions that you developed, interview school counselors to determine what information should be included in the Web site, who will provide the information, and so on.
5. After gathering the required information, sketch a design for the home page of the Web site on paper.
6. Share your design sketch with students of the school and some of the counselors to get their opinions on your design.
7. Redraw the design on paper, making any changes based on the input from the students and counselors.
8. Write Original Design on the first design sketch.
9. Write Second Design on the second design sketch.
10. Write your name on the lab1-2solution printout and sketches and hand them in to your instructor.

In the Lab

Lab 3: Asking Planning Phase Questions: Internet, Intranet, and Extranet Designs

Problem: Three different types of Web sites were discussed in this chapter — Internet, intranet, and extranet. Each type of Web site is designed for a different target audience. Think of a business (for example, a restaurant, library, or card store) that you frequently visit and how that business might use an Internet, intranet, and extranet site. The Planning phase questions found in Table 1–4 on page HTML 15 have been reproduced in Table 1–8. Determine the answers to these questions and enter your ideas in the table. If there are questions that are difficult/impossible to answer directly (for example, What are users' computing environments?), list ways that you can find the answers to those questions.

Table 1–8 Planning Phase Questions

Type Of Business	Planning Question	Internet	Intranet	Extranet
What is the purpose of this Web site?				
Who will use this Web site?				
What are users' computing environments?				
Who owns and authors the information on the Web site?				
Who decides if/where the information goes on the Web site?				

Instructions: Start your word-processing program. Open the file lab1-3.doc from the Chapter01\IntheLab folder of the Data Files for Students. See the inside back cover of this book for instructions for downloading the Data Files for Students, or contact your instructor for information on accessing the required files. Perform the following tasks using your word-processing program.

1. Enter the type of business in the first row of the table. Determine the answers to the first question for all three types of Web sites and then enter the answers in the appropriate table cells. If the business you choose has no reason to maintain one of the three types of Web sites (Internet, intranet, or extranet), thoroughly identify in your answer why they would not need it.
2. Continue answering the other four questions.
3. Save the file using the file name lab1-3solution.doc and then submit it in the format specified by your instructor.

Cases and Places

Apply your creative thinking and problem-solving skills to design and implement a solution.

• EASIER •• MORE DIFFICULT

• 1: Learn More About Web Access Issues

Your company wants to offer online courses to employees. Several employees have physical challenges, and it is imperative that the online courses be accessible to everyone. Your manager has asked you to learn more about accessibility guidelines to determine what changes are needed to make the company's online courses accessible to those with physical challenges. Research accessibility issues on the Web and determine what needs should be considered to satisfy accessibility requirements. Make sure to visit the W3.org Web site. Consider the following questions when doing your research: What types of physical challenges do you have to consider when developing Web pages? What recommendations do the Web sites make for accessibility? Why is this important to you as a Web developer?

• 2: Determine Web Site Structure

You have recently started a job as a Web developer at Triple-Tom Design. In your new job, you often are asked to restructure clients' existing Web sites to make them more user friendly and easier to navigate. Find a Web site that utilizes more than one Web site structure (linear, hierarchical, and/or webbed). Is the information conveyed on the Web site displayed in the appropriate structure? Does the structure effectively support the information communicated? Print the home page of the Web site that you found. On a blank sheet of paper, sketch a design that you think might be more appropriate for the message. Use a word-processing program to create a document that explains why your new design is more effective.

•• 3: Learn More About CSS

This chapter introduced the use of Cascading Style Sheets (CSS) in Web development. You will utilize CSS throughout this book, so it is important that you become familiar with it. Visit the W3Schools Web site (w3schools.com) to learn more about CSS. Find three other sources of information about CSS on other Web sites. Using a word-processing program, create a document that briefly describes CSS and how you can best utilize it for Web development.

Continued >

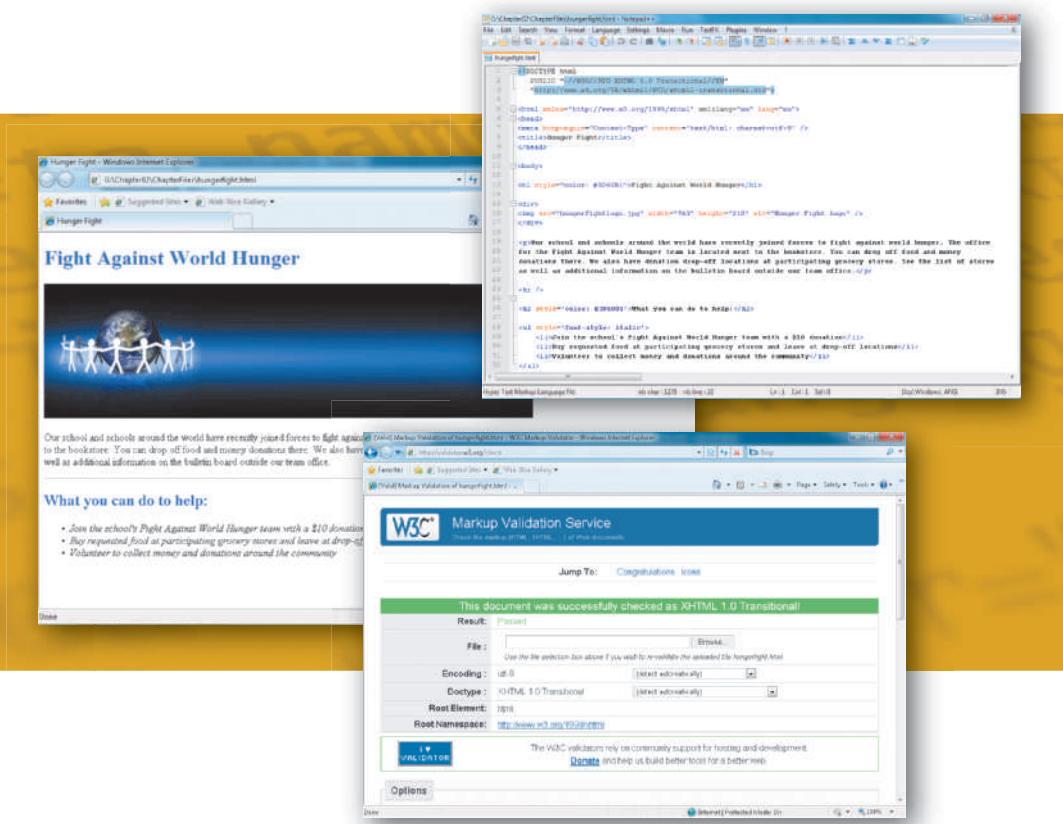
Cases and Places *continued***•• 4: Design a Web Site for Your Sister****Make It Personal**

Your sister would like to sell the jewelry that she makes online. You would like to develop a Web site for her that can display her jewelry and be a means for sales transactions. Thoroughly investigate the Web sites of other online jewelers and print home pages from three of them. Before starting on the design, you decide to create a list of Web design principles to which the Web site will adhere. Search the Web for more information about Web site design. Find three Web sites that give information about Web design principles. In a word-processing document, take the ideas presented in this chapter together with the ideas presented in the other Web sites and make a comprehensive list of Web design principles. Determine whether or not the Web sites that you printed comply with good design principles. Where appropriate, identify any conflicting design principles discussed in the Web sites.

•• 5: Create a Usability Survey**Working Together**

Your school recently updated its Web site. The school administration has selected a team to develop a usability survey or questionnaire that you can give to a group of users (including students, parents, and teachers) to evaluate the new Web site. What types of information do you hope to gain by distributing this survey or questionnaire? How can you convey information on the survey or questionnaire so it clearly identifies what you are asking? Create a usability survey using your word-processing program. Give the survey or questionnaire to at least five people, including at least one from each group identified above. Allow participants to complete the survey or questionnaire and then look at the results. If possible, ask the users what they thought the various questions conveyed. Is that what you wanted to convey? If not, think of clearer, more relevant questions and redistribute the survey to another group of participants.

2 Creating and Editing a Web Page Using Inline Styles



Objectives

You will have mastered the material in this chapter when you can:

- Identify elements of a Web page
- Start Notepad++ and describe the Notepad++ window
- Enable word wrap in Notepad++
- Enter HTML tags
- Enter a centered heading and a paragraph of text
- Create an unordered, ordered, or definition list
- Save an HTML file
- Use a browser to view a Web page
- Activate Notepad++
- Identify Web page image types and attributes
- Add an image, change the color of headings on a Web page, change bulleted list style, and add a horizontal rule using inline styles
- View the HTML source code in a browser
- Print a Web page and an HTML file
- Quit Notepad++ and a browser

2 | Creating and Editing a Web Page Using Inline Styles

Introduction

With an understanding of the Web development life cycle, you should have a good idea about the importance of proper Web site planning, analysis, and design. After completing these phases, the next phase is the actual development of a Web page using HTML. As discussed in Chapter 1, Web pages are created employing HTML, which uses a set of special tags to define the structure, layout, and appearance of a Web page. In this chapter, you create and edit a Web page using basic HTML tags.

Project — Fight Against World Hunger Web Page

Chapter 2 illustrates how to use HTML to create a Web page for the school's Fight Against World Hunger team, as shown in Figure 2–1a. The team is joining other students around the world to collect donations and money. Because you are the only Web development major in the group, team members have asked for your help in developing a Web page to advertise the team's cause. The Fight Against World Hunger Web page will include general information about the team, along with information on how to donate and participate.

To enter text and HTML tags used to create the Web page, you will use a program called Notepad++, as shown in Figure 2–1b. **Notepad++** is a basic text editor you can use for simple documents or for creating Web pages using HTML. Previous editions of this book used Notepad, a text editor that is a part of the Windows operating system. Notepad worked well to enter the HTML elements and Web page content, but Notepad++ is a more sophisticated text editor with more features. Notepad++ has line numbering, which is very helpful when reading code. It also is color-coded for the text that is entered, as you will see later in the chapter. Because of this added versatility, Notepad++ is the chosen text editor for this edition. You will use the Microsoft Internet Explorer browser to view your Web page as you create it. By default, Internet Explorer is installed with Windows, and Notepad++ can be downloaded for free on the Web. If you do not have Notepad++ or Internet Explorer available on your computer, other text editor or browser programs will work.

Overview

As you read this chapter, you will learn how to create the Web page shown in Figure 2–1 by performing these general tasks:

- Enter HTML code into the Notepad++ window.
- Save the file as an HTML file.
- Enter basic HTML tags and add text to the file.
- Organize the text by adding headings and creating a bulleted list.
- Enhance the Web page's appearance with inline styles.
- View the Web page and HTML code in your browser.
- Validate the Web page.
- Print the Web page.

Hunger Fight - Windows Internet Explorer
G:\Chapter02\ChapterFiles\hungerfight.html

Favorites Suggested Sites Web Slice Gallery
Hunger Fight

Fight Against World Hunger

Our school and schools around the world have recently joined forces to fight against world hunger. The office for the Fight Against World Hunger team is located next to the bookstore. You can drop off food and money donations there. We also have donation drop-off locations at participating grocery stores. See the list of stores as well as additional information on the bulletin board outside our team office.

What you can do to help:

- Join the school's Fight Against World Hunger team with a \$10 donation
- Buy requested food at participating grocery stores and leave at drop-off locations
- Volunteer to collect money and donations around the community

Done Computer | Protected Mode: Off 100% ▾

(a) Fight Against World Hunger Web page.

```
G:\Chapter02\ChapterFiles\hungerfight.html - Notepad++
File Edit Search View Format Language Settings Macro Run TestFx Plugins Window ?
hungerfight.htm
1 <!DOCTYPE html
2   PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3   "http://www.w3.org/TR/xhtml/DTD/xhtml1-transitional.dtd">
4
5 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
6   <head>
7     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
8     <title>Hunger Fight</title>
9   </head>
10
11 <body>
12
13   <h1 style="color: #3D60B1">Fight Against World Hunger</h1>
14
15   <div>
16     
17   </div>
18
19   <p>Our school and schools around the world have recently joined forces to fight against world hunger. The office for the Fight Against World Hunger team is located next to the bookstore. You can drop off food and money donations there. We also have donation drop-off locations at participating grocery stores. See the list of stores as well as additional information on the bulletin board outside our team office.</p>
20
21   <hr />
22
23   <h2 style="color: #3D60B1">What you can do to help:</h2>
24
25   <ul style="font-style: italic">
26     <li>Join the school's Fight Against World Hunger team with a $10 donation</li>
27     <li>Buy requested food at participating grocery stores and leave at drop-off locations</li>
28     <li>Volunteer to collect money and donations around the community</li>
29   </ul>
30
31
32
Hyper Text Markup Language file nb char:1279 nb line:32 Ln:1 Col:1 Sel:0 Dos/Windows: ANSI Ins
```

(b) HTML code used to create the Web page.

Figure 2-1

**Plan
Ahead****General Project Guidelines**

When creating a Web page, the actions you perform and decisions you make will affect the appearance and characteristics of the finished page. As you create a Web page, such as the project shown in Figure 2–1 on the previous page, you should follow these general guidelines:

1. **Complete Web page planning.** Before developing a Web page, you must know the purpose of the Web site, identify the users of the site and their computing environments, and decide who owns the information on the Web page.
2. **Analyze the need for the Web page.** In the analysis phase of the Web development life cycle, you should analyze what content to include on the Web page. In this phase, you determine the tasks and the information that the users need. Refer to Table 1–4 on page HTML 15 for information on the phases of the Web development life cycle.
3. **Choose the content for the Web page.** Once you have completed the analysis, you need to determine what content to include on the Web page. Follow the *less is more* principle. The less text, the more likely the Web page will be read. Use as few words as possible to make a point.
4. **Determine the file naming convention that you will use for this Web page.** Before you start creating and saving files, you should decide on a standard way of naming your files. Should you use the .htm or .html extension? As explained later in the chapter, you use the .htm extension when the host Web server only allows short file names. You use .html when the host Web server allows long file names. What name should you give your file to indicate the file's content or purpose? For instance, naming a Web page page1.html does not describe what that Web page is; a more descriptive name is helpful in development of the Web site.
5. **Determine where to save the Web page.** You can store a Web page permanently, or save it, on a variety of storage media, including a hard disk, USB flash drive, CD, or DVD. Your instructor or the company for whom you are developing the Web page may have specific storage media requirements.
6. **Determine what folder structure to use on your storage device.** Once you have determined the storage media to use, you should also determine folder location, structure, and names on which to save the Web page. This should be done before you start to save any of your files.
7. **Identify how to format various elements of the Web page.** The overall appearance of a Web page significantly affects its ability to communicate clearly. Examples of how you can modify the appearance, or **format**, of the Web page include adding an image, color to headings, and horizontal rules.
8. **Find appropriate graphical images.** Eye-catching graphical images help convey the Web page's overall message and add visual interest. Graphics can be used to show a product, service, result, or benefit, or visually convey a message that is not expressed easily with words.
9. **Establish where to position and how to format the graphical images.** The position and format of the graphical images should grab the attention of passersby and draw them into reading the Web page.
10. **Test the Web page for XHTML compliance.** An important part of Web development is testing to assure that your Web page follows XHTML standards. The World Wide Web Consortium (W3C) has an online validator that allows you to test your Web page and clearly explains any errors.

When necessary, more specific details concerning the above guidelines are presented at appropriate points in the chapter. The chapter also will identify the actions performed and decisions made regarding these guidelines during the creation of the Web page shown in Figure 2–1a.

Elements of a Web Page

Today, many people — individuals, students, teachers, business executives, Web developers, and others — are developing Web pages for personal or professional reasons. Each person has his or her own style and the resulting Web pages are as diverse as the people who create them. Most Web pages, however, include several basic features, or elements, as shown in Figure 2–2.

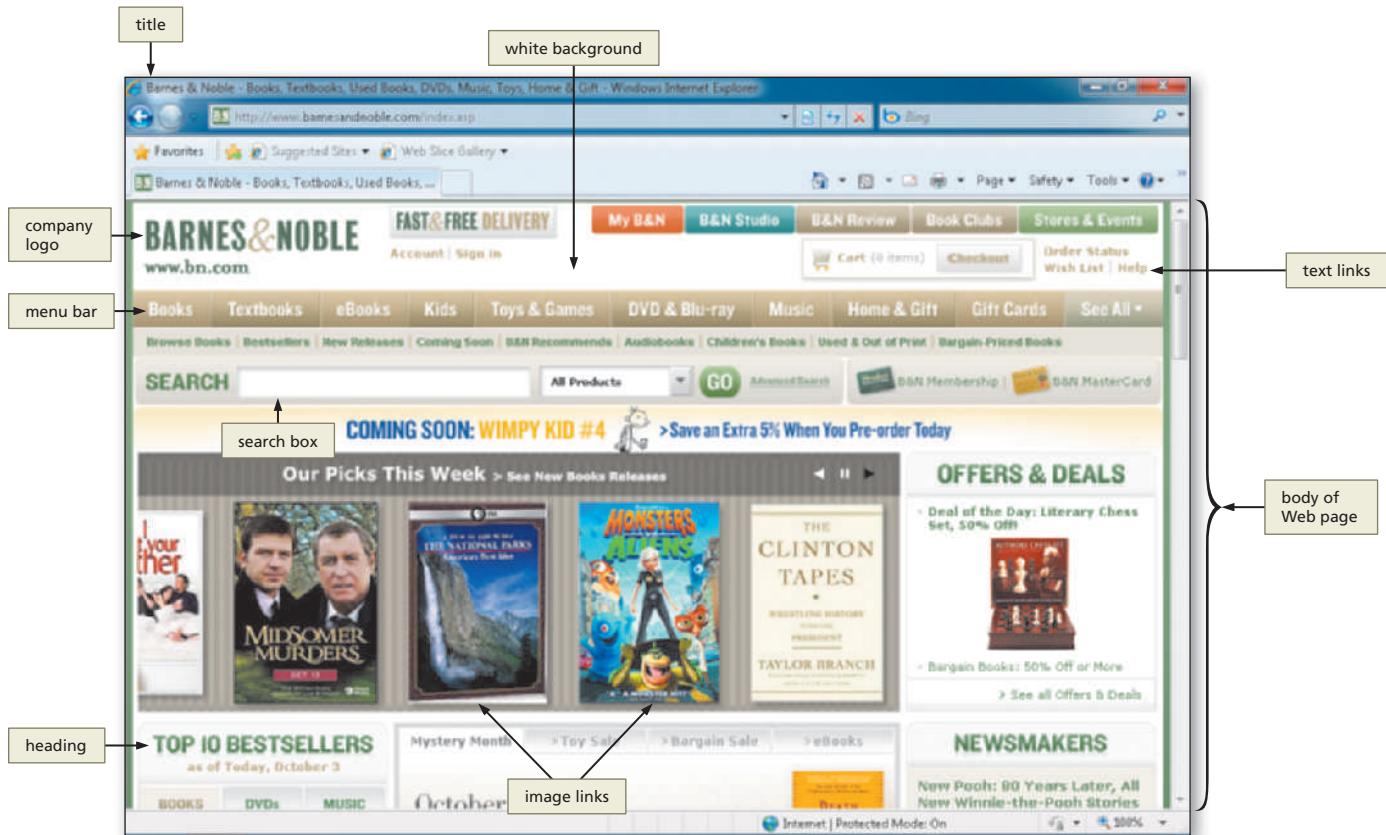


Figure 2–2 Elements of a Web page.

Window Elements

The **title** of a Web page is the text that appears on the title bar and taskbar of the browser window when the Web page appears. The title also is the name assigned to the page if a user adds the page to the browser's list of **favorites**, or **bookmarks**. Because of its importance, you always should include a title on your Web page. The title, which usually is the first element you see, should identify the subject or purpose of the page. The title should be concise, yet descriptive, and briefly explain the page's content or purpose to the visitor.

The **body** of the Web page contains the information that is displayed in the browser window. The body can include text, graphics, and other elements. The Web page displays anything that is contained within the `<body>` (start body) and `</body>` (end body) tags. The **background** of a Web page is a solid color, a picture, or a graphic against which the other elements on the Web page appear. When choosing your background, be sure it does not overpower the information on the Web page. As you surf the Web, watch for background colors or images that do not allow the content of the Web page to show through. This is certainly a “what not to do” guideline for Web developers.



Favorites and Bookmarks

Internet Explorer and Mozilla Firefox have a feature that allows you to add Web pages to a list so you quickly can access them in the future. Internet Explorer refers to these as Favorites, while Firefox calls them Bookmarks. Web developers need to make sure that they include a descriptive title on their Web pages because that is the title that is shown in the bookmark or favorite.

HTML Resources

The Web has many wonderful sources of information on HTML and Web page development. One of the better sources is the HTML Goodies Web site, which has primers and tutorials on a variety of topics as well as free downloads and discussion areas. To learn more about this Web site, search for the term "HTML Goodies" in a search engine.

Text Elements

Normal text is the default text format used for the main content of a Web page. Normal text can be used in a standard paragraph or formatted to appear as: bold, italic, or underlined; in different colors; and so on. You can also use inline cascading style sheets (CSS) to alter the format of the text, an approach used throughout this book. Normal text also can be used in a series of text items called a **list**. Typically, lists are bulleted or numbered. Various attributes of lists can be altered. For example, you might want to have square bullets rather than the default round bullets, or to have your list text in italic or bold.

Headings are used to set off paragraphs of text or different sections of a page. Headings are a larger font size than normal text and often are bold or italic or a different color than normal text. Heading sizes run from 1 (the largest) to 6 (the smallest). You generally go from one heading size to the next smallest when setting up a Web page.

Image Elements

Web pages typically use several different types of graphics, or images, such as an icon, bullet, line, photo, illustration, or other picture. An image used in a Web page also is called an **inline image**, which means the image or graphic file is not part of the HTML file. Instead, the Web browser merges the separate graphic file into the Web page as it is displayed in the browser window. The HTML file contains `` tags that tell the browser which graphic file to request from the server, where to insert it on the page, and how to display it.

Web pages typically use several different types of inline images. An **image map** is a special type of inline image in which you define one or more areas as hotspots. A **hotspot** is an area of an image that activates a function when selected. For example, each hotspot in an image map can link to a different Web page. Some inline images are **animated**, meaning they include motion and can change in appearance.

Horizontal rules are lines that are displayed across a Web page to separate different sections of the page. Although the appearance of a horizontal rule can vary, many Web pages use an inline image as a horizontal rule. Alternatively, you can use the horizontal rule tag (`<hr />`) to add a simple horizontal rule, such as the one used in this chapter project.

Hyperlink Elements

One of the more important elements of a Web page is a hyperlink, or link. A **link** is text, an image, or another Web page element that you click to instruct the browser to go to a location in a file or to request a file from a server. On the Web, links are the primary way to navigate between Web pages and among Web sites. Links point not only to Web pages, but also to graphics, sound, video, program files, e-mail addresses, and parts of the same Web page. Text links, also called hypertext links, are the most commonly used hyperlinks. For example, the text "Order Status" in Figure 2–2 links to the status of the user's orders. When text identifies a hyperlink, it usually appears as underlined text, in a color different from the rest of the Web page text. Image links are also very common. For example, there are two image links identified in Figure 2–2. Clicking either of those image links sends (or links) the user to another Web page that contains further information about those items. A company logo often serves as an image link to a home page or corporate information.

Defining Web Page Structure

To create an HTML document, you use a text editor to enter information about the structure of the Web page, the content of the Web page, and instructions for how that content should be displayed. This book uses the Notepad++ text editor to enter the HTML elements and content for all projects and exercises.

Before you begin entering the content for this project, you must start by entering tags that define the overall structure of the Web page. You do this by inserting a `<!DOCTYPE>` tag and five tags (`<html>`, `<head>`, `<meta />`, `<title>`, and `<body>`) together with the closing tags (`</html>`, `</head>`, `</title>`, and `</body>`). These tags define the structure of a standard Web page and divide the HTML file into its basic sections: header information and the body of the page that contains text and graphics.

The `<!DOCTYPE>` tag is used to tell the browser which HTML or XHTML version and type the document uses. The World Wide Web Consortium (W3C) supports three document types for HTML or XHTML: strict, transitional, and frameset. The **strict** document type is specified when you want to prohibit the use of deprecated tags. **Deprecated tags** are tags that the W3C has earmarked for eventual removal from their specifications, because those tags have been replaced with newer, more functional tags. The **transitional** document type allows the use of deprecated tags. The **frameset** document type, which is used to support frames on a Web page, also allows the use of deprecated tags. The `<!DOCTYPE>` tag includes a URL that references a Document Type Definition found on the w3.org Web site. Although this book does not use deprecated tags, the projects do use the XHTML 1.0 transitional document type, because the projects use a few tags (e.g., `name`) that are not supported by the XHTML 1.1 standard. A **Document Type Definition (DTD)** is a file containing definitions of tags and instructions for how they should be used in a Web page. The project in this chapter uses the transitional document type.

BTW

The DOCTYPE Tag

The W3Schools Web site provides additional information about the DOCTYPE tags used for the strict, transitional, and frameset document types. To learn more about the DOCTYPE tag, visit the W3C Web site. It provides a wealth of information on this and other HTML tags.

Defining the HTML Document

The first set of tags beyond the `<!DOCTYPE>` tag, `<html>` and `</html>`, indicates the start and end of an HTML document. This set of tags contains all of the content of the Web page, the tags that format that content, and the tags that define the different parts of the document. Software tools, such as browsers, use these tags to determine where the HTML code in a file begins and ends.

The Header The next set of tags, `<head>` and `</head>`, contains the Web page title and other document header information. One of the tags inserted into the `<head>` `</head>` container is the meta tag. The `<meta />` tag has several functions. In this chapter, it is used to declare the character encoding UTF-8. The **Unicode Transformation Format (UTF)** is a compressed format that allows computers to display and manipulate text. When the browser encounters this meta tag, it will display the Web page properly, based on the particular UTF-8 encoding embedded in the tag. UTF-8 is the preferred encoding standard for Web pages, e-mail, and other applications. The encoding chosen also is important when validating the Web page. The meta tag has other purposes that are described in subsequent chapters of the book. The `<title>` tag is another tag inserted into the `<head>` `</head>` container. The `<title>` and `</title>` tags indicate the title of the Web page, which appears on the browser title bar and taskbar when the Web page is displayed in the browser window. The title also is the name given to the page when a user adds the page to a favorites or bookmarks list.

BTW

XHTML Compliance

To make your HTML files compliant with XHTML standards, always enter tags in lowercase (with the exception of the `<!DOCTYPE>` tag, which is always uppercase). Throughout this book, the project directions follow these standards to help you learn good HTML and XHTML coding practices.

The Body The final set of tags, `<body>` and `</body>`, contains the main content of the Web page. All text, images, links, and other content are contained within this final set of tags. Table 2–1 on the next page lists the functions of the tags described so far, as well as other tags that you use in this chapter.

Table 2–1 Basic HTML Tags and Their Functions

HTML Tag	Function
<!DOCTYPE>	Indicates the version and type of HTML used; includes a URL reference to a DTD
<html> </html>	Indicates the start and end of an HTML document
<head> </head>	Indicates the start and end of a section of the document used for the title and other document header information
<meta />	Indicates hidden information about the Web page
<title> </title>	Indicates the start and end of the title. The title does not appear in the body of the Web page, but appears on the title bar of the browser.
<body> </body>	Indicates the start and end of the body of the Web page
<hn> </hn>	Indicates the start and end of the text section called a heading; sizes range from <h1> through <h6>. See Figure 2–8a on page HTML 46 for heading size samples.
<p> </p>	Indicates the start and end of a new paragraph; inserts a blank line above the new paragraph
 	Indicates the start and end of an unordered (bulleted) list
 	Indicates that the item that follows the tag is an item within a list
<hr />	Inserts a horizontal rule
 	Inserts a line break at the point where the tag appears

BTW**WordPad**

WordPad is a text editor included with Windows that you can also use to create HTML files. To start WordPad, click the Start button on the taskbar, point to All Programs on the Start menu, point to Accessories on the All Programs submenu, and then click WordPad on the Accessories submenu. WordPad help provides tips on how to use the product.

Most HTML start tags, such as <html>, <head>, <title>, and <body>, have corresponding end tags, </html>, </head>, </title>, and </body>. Note that, for tags that do not have end tags, such as <meta />, <hr />, and
, the tag is closed using a space followed by a forward slash.

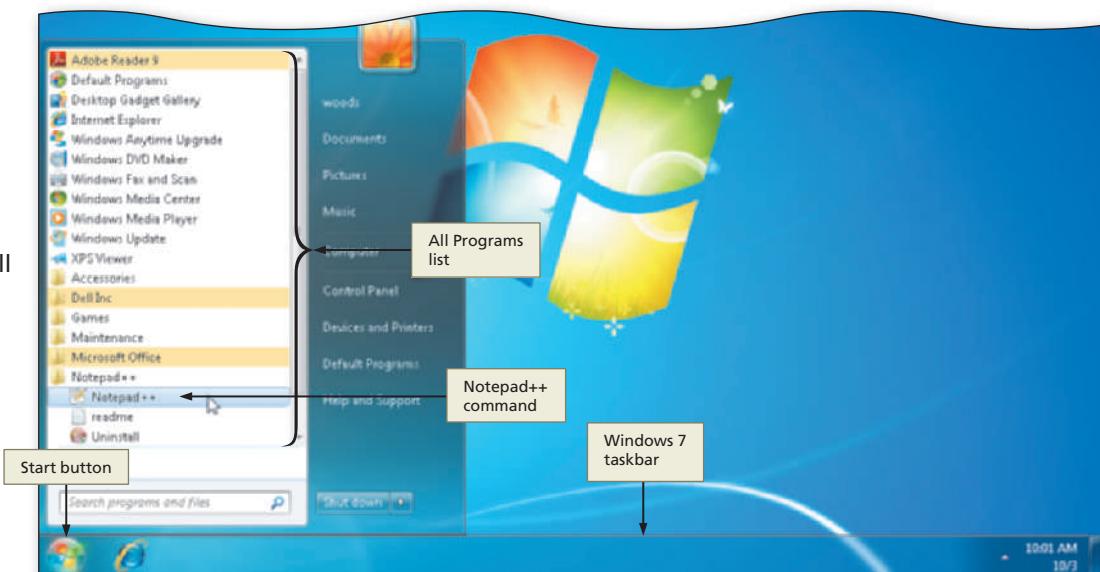
To Start Notepad++

With the planning, analysis, and design of the Web page complete, you can begin developing the Web page by entering HTML using a text editor.

The following steps, which assume Windows 7 is running and Notepad++ is installed, start Notepad++ based on a typical installation. You may need to ask your instructor how to download, install, and start Notepad++ for your computer.

1

- Click the Start button on the Windows taskbar to display the Start menu.
- Click All Programs at the bottom of the left pane on the Start menu to display the All Programs list.
- Click the Notepad++ folder in the All Programs list (Figure 2–3).

**Figure 2–3**

2

- Click Notepad++ in the list to display a blank Notepad++ window (Figure 2–4).
- If the Notepad++ window is not maximized, click the Maximize button on the Notepad++ title bar to maximize it.

Q&A

What is a maximized window?

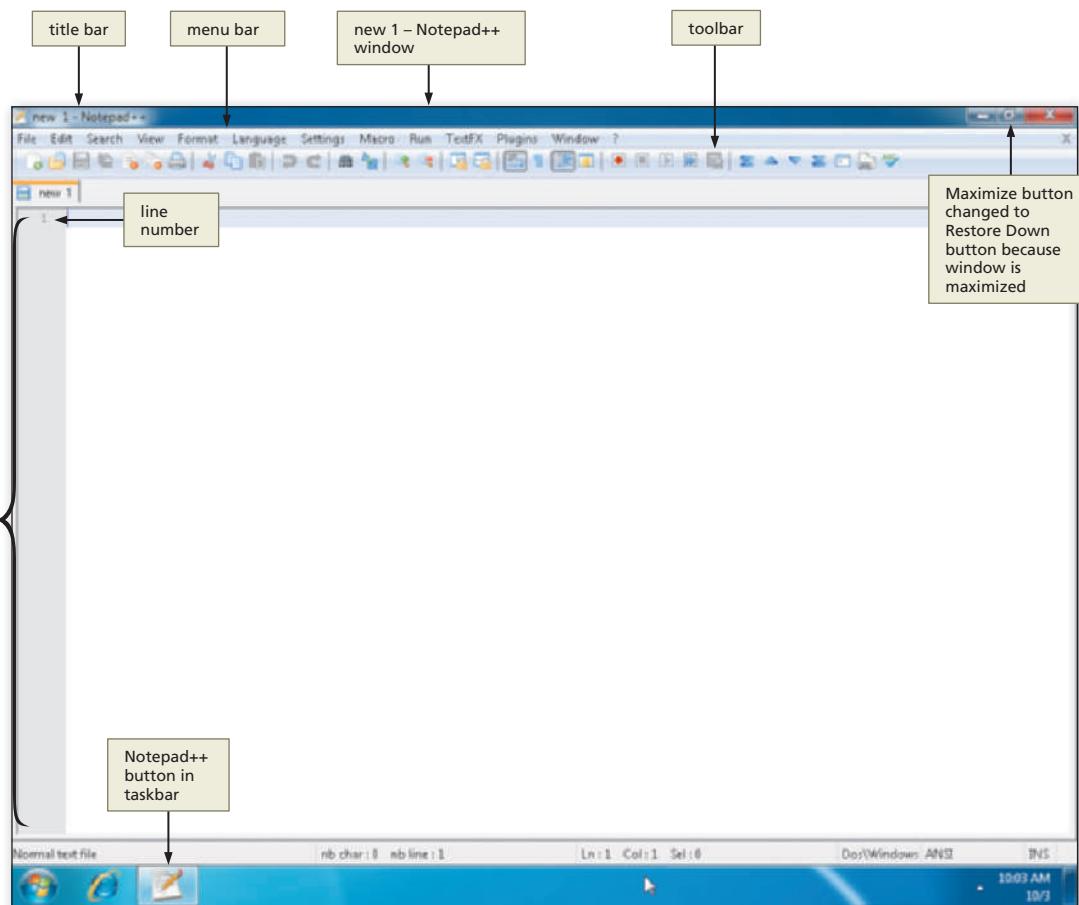


A maximized window fills the entire screen. When you maximize a window, the Maximize button changes to a Restore Down button.

Q&A

How can I add Notepad++ to my Start menu or the taskbar?

To add Notepad++ to the Start menu or taskbar, complete Step 1 above. Right-click Notepad++, then click Pin to Start menu or Pin to taskbar.

**Figure 2–4****BTW****Notepad++ Help**

Notepad++ has a wealth of help information available. There is Notepad++ Help internal to the program as well as help facilities online. NpWiki is the wiki for Notepad++ that you can access. The primary aim of this wiki is to be a storehouse for all information about Notepad++.

Other Ways

- | | |
|--|--|
| 1. Double-click Notepad++ icon on desktop, if one is present | 2. Click Notepad++ on Start menu, if it is present |
|--|--|

To Enable Word Wrap in Notepad++

In Notepad++, the text entered in the text area scrolls continuously to the right unless the word wrap feature is enabled, or turned on. **Word wrap** causes text lines to break at the right edge of the window and appear on a new line, so all entered text is visible in the Notepad++ window. With paragraphs of text and word wrap enabled, the text all appears on one logical line number even though it may display on multiple physical lines in Notepad++. Word wrap does not affect the way text prints. The following step shows how to enable word wrap in Notepad++.

1

- Click View on the menu bar (Figure 2–5).
- If Word wrap does not have a check mark next to it, click Word wrap.

Q&A

How do I know if Word wrap is enabled?

When Word wrap is enabled, a check mark precedes the Word wrap command on the View menu, and when you type, your words remain on the screen.

Q&A

What happens to the text if Word wrap is not enabled?

The text of a paragraph would appear all on one line in Notepad++ and scroll off the screen, though the Web page would still be displayed correctly in the browser. For readability in Notepad++, you should enable Word wrap.

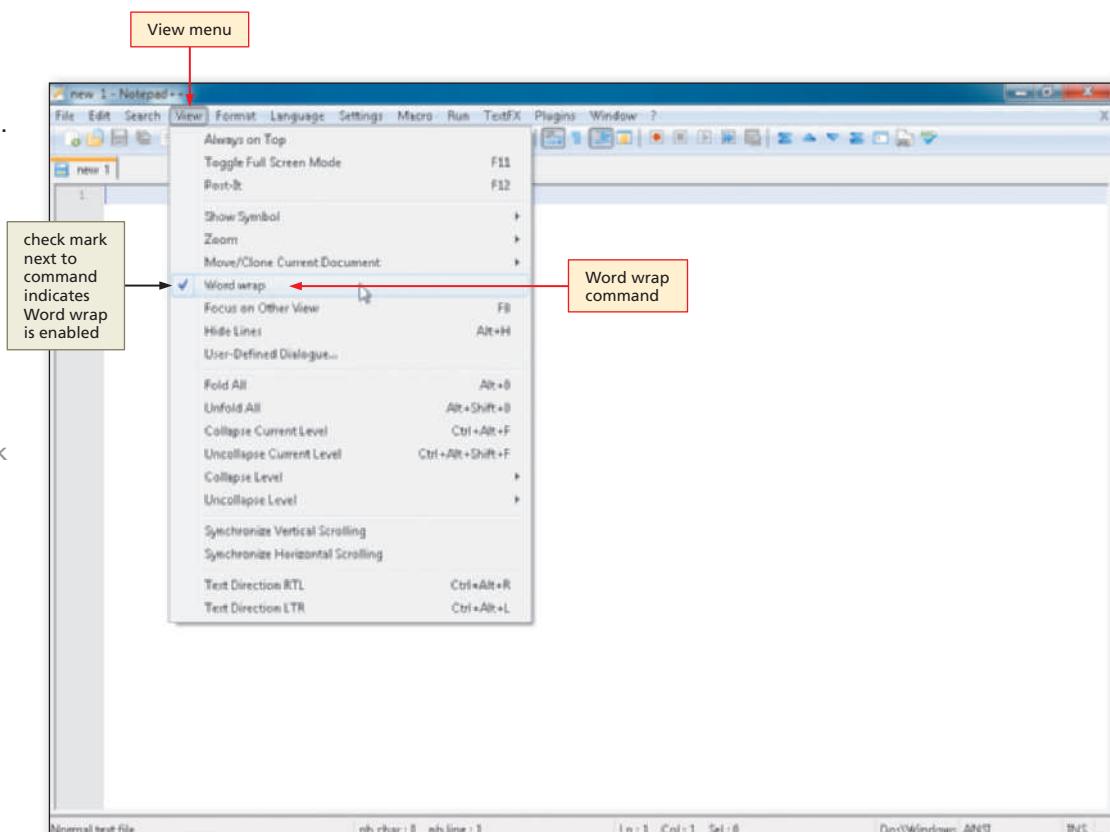


Figure 2–5

To Define the Web Page Structure Using HTML Tags

The first task is to enter the initial tags that define the Web page structure. Table 2–2 contains the HTML tags and text used to create the Web page shown in Figure 2–1a on page HTML 35. In this chapter and throughout this book, where large segments of HTML code or text are to be entered, you will find this code or text in tables with line number references, rather than within the steps. The steps will direct you to enter the text shown in the tables.

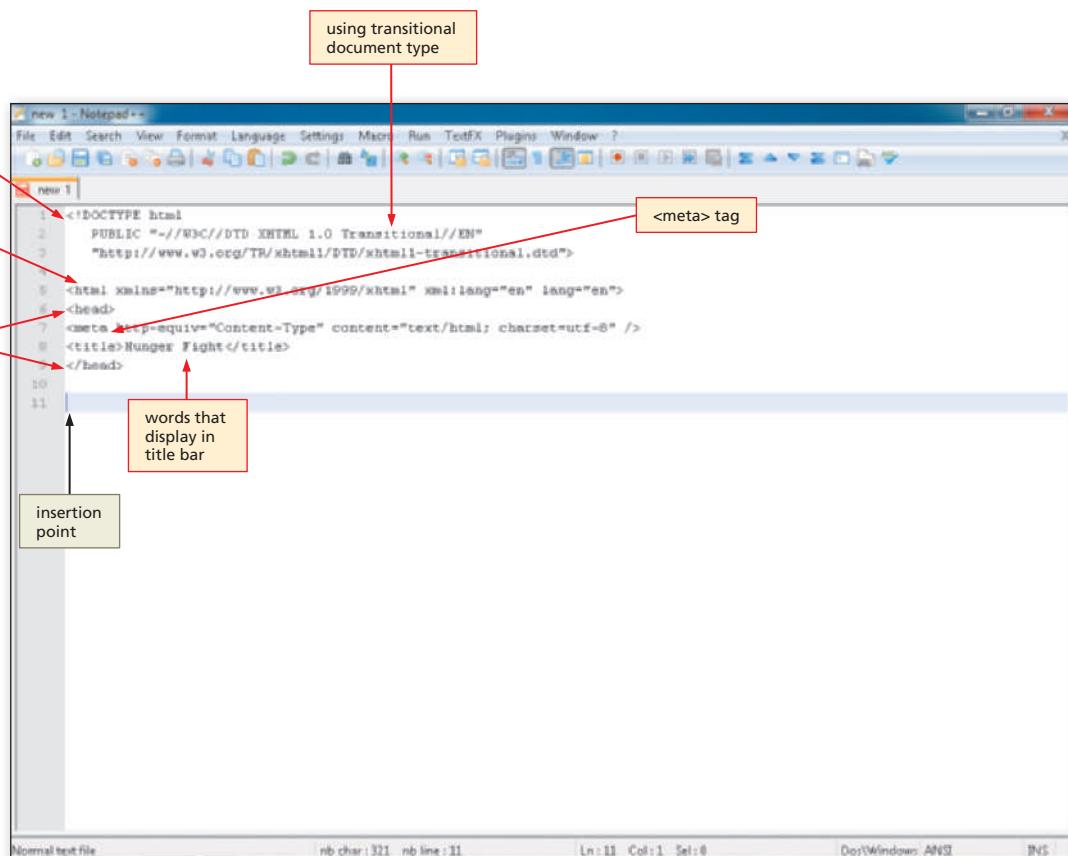
Table 2–2 Initial HTML Tags

Line	HTML Tag and Text
1	<!DOCTYPE html
2	PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3	" http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd ">
4	
5	<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
6	<head>
7	<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
8	<title>Hunger Fight</title>
9	</head>

The following steps illustrate how to enter the initial tags that define the structure of the Web page.

1

- Enter the HTML code shown in Table 2–2 (Figure 2–6). Press ENTER at the end of each line. If you make an error as you are typing, use the BACKSPACE key to delete all the characters back to and including the incorrect characters, and then continue typing.

**Figure 2–6**

- Press the ENTER key twice to start the next line of code, leaving one blank line after the </head> tag.
- Compare what you typed to Figure 2–6. If you notice errors, use your mouse pointer or arrow keys to move the insertion point to the right of each error and use the BACKSPACE key to correct the error.

2

- On line 11, type <body> and then press the ENTER key twice.
- Type </body> and then press the ENTER key.
- Type </html> as the end tag (Figure 2–7).
- Compare what you typed to Figure 2–7 and correct errors in your typing if necessary.

```

<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Hunger Flight</title>
</head>
<body>
</body>
</html>

```

Figure 2–7

Q&A

Why am I using the transitional document type? Doesn't that allow the use of deprecated tags?

Yes, you can use deprecated tags with a transitional document type. Some valid tags (e.g., name) cannot be used with the XHTML 1.1 standard, so this text uses XHTML 1.0 with the transitional document type.

Q&A

What is the difference between the <title> and <body> tags?

The text contained within the <title> </title> tags is what appears on the browser title bar when the Web page is displayed in the browser window. The text and graphics contained within the <body> </body> tags are what is displayed in the browser window.

Q&A

Do I have to type the initial HTML tags for every Web page that I develop?

The same initial HTML tags are used in many other chapters. To avoid retyping these tags, you can save the code that you just typed, and give it a new file name, something like structure.html or template.html. If you save this file at the root level of your folders, you will have easy access to it for other chapters.

Q&A

Can I use either uppercase or lowercase letters for my HTML code?

To make your HTML files compliant with XHTML standards, always enter tags in lowercase (with the exception of the <!DOCTYPE> tag, which is always uppercase). In this book, the project directions follow these guidelines to help you learn acceptable HTML and XHTML coding standards.

Identify how to format various elements of the text.

By formatting the characters and paragraphs on a Web page, you can improve its overall appearance. On a Web page, consider the following formatting suggestions.

- **Effectively utilize headings.** The main heading is generally the first line of text on the Web page. It conveys the purpose of the Web page, such as asking for help with the food drive. Heading size standards should be followed, as shown in Figure 2–8 on the next page. The main heading should be size 1, and subtopics or subheadings should be size 2. It is generally not a good idea to jump from one heading size to a heading two sizes smaller. For instance, if your main heading is size 1, then the next heading down should be heading size 2, not heading size 4.
- **Use default text size when appropriate.** The body text consists of all text between the heading and the bottom of the Web page. This text highlights the key points of the message in as few words as possible. It should be easy to read and follow. While emphasizing the positive, the body text must be realistic, truthful, and believable. The default font size and style are appropriate to use for the body of text.
- **Highlight key paragraphs with bullets.** A **bullet** is a dot or other symbol positioned at the beginning of a paragraph. The bulleted list contains specific information that is more clearly identified by a list versus a paragraph of text.

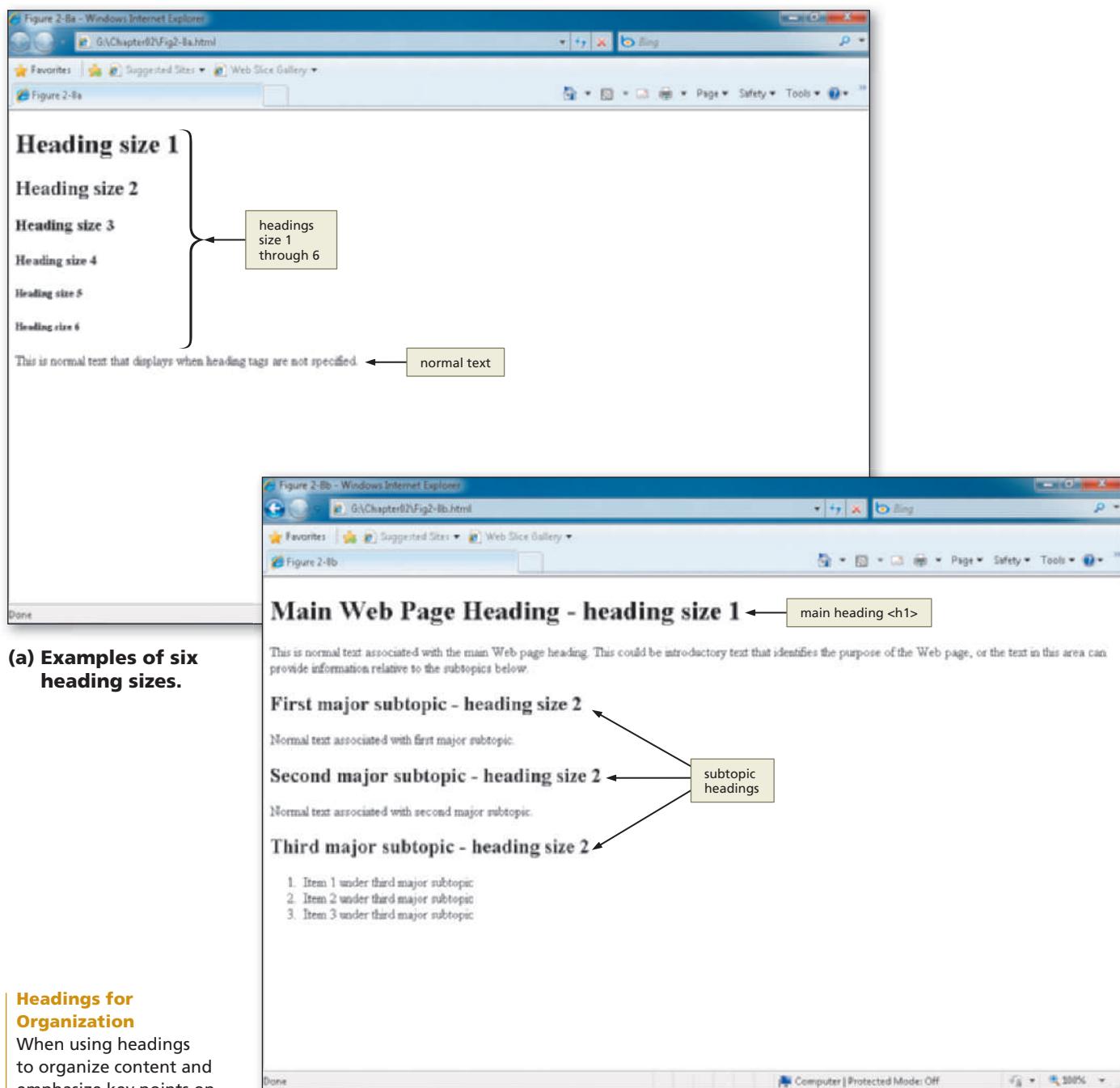
Plan Ahead

Entering Web Page Content

Once you have established the Web page structure, it is time to enter the content of the Web page, including headings, informational text paragraphs, and a bulleted list.

Headings are used to separate text or add new topics on the Web page. Several styles and sizes of headings exist, indicated by the tags `<h1>` through `<h6>`, with `<h1>` being the largest. Generally, you use the Heading 1 style for the main heading. Figure 2–8a on the next page shows a Web page using various sizes of headings. A Web page usually has only one main heading; therefore, the HTML file for that Web page usually has only one set of `<h1>` `</h1>` tags. One method of maintaining a consistent look on a Web page is to use the same heading size for headings at the same topic level (Figure 2–8b). Notice that the paragraphs of text and the bulleted lists are all separated by size 2 headings in Figure 2–8b. This separation indicates that the text (i.e., two paragraphs plus one bulleted list) is all at the same level of importance on the Web page.

Web pages generally contain a significant amount of text. Because you turned Word wrap on (Figure 2–5 on page HTML 42) in Notepad++, you will see all of the text that you type in one Notepad++ window. Breaking the text into paragraphs helps to separate key ideas and makes the text easier to read. Paragraphs are separated with a blank line by using `<p>` (start paragraph) and `</p>` (end paragraph) tags. Putting too much text on one Web page is not a good choice. Your audience can get lost in large amounts of text. If you find that you have to press the Page Down key dozens of times to get to the bottom of the Web page, you need to think about restructuring your Web page. You can split up large pieces of information under more headings, which will be more manageable and more readable.

**BTW****Headings for Organization**

When using headings to organize content and emphasize key points on a Web page, be sure to use them consistently.

That is, if you use a Heading 2 (`<h2>`) style for a specific level of text, you always should use a Heading 2 style to break up information at that level. Also, do not skip levels of headings in your document. For example, do not start with a Heading 1 (`<h1>`) style and then use a Heading 3 (`<h3>`) style.

Sometimes text on a Web page is easier for users to read and understand when it is formatted as a list, instead of as a paragraph. HTML provides several types of lists, but the most popular are unordered (bulleted) and ordered (numbered) lists. During the design phase of the Web development life cycle, you decide on the most effective way to structure the Web content and format the text on the Web page. Your main goal is to give Web page visitors an effective way to find the information that they need. If users cannot easily find what they need, they will not revisit your Web site.

Figure 2-8

To Enter a Heading

The heading, Fight Against World Hunger, is the main heading and indicates the main message of the Web page. To draw attention to this heading, you will use the `<h1>` tag for the heading. The following step illustrates how to enter a heading on the Web page.

1

- With the insertion point on line 12, press the ENTER key once, leaving a blank line after the `<body>` tag.
- Type `<h1>Fight Against World Hunger</h1>` in the text area, and then press the ENTER key twice (Figure 2–9).

Q&A

Why did you put an additional line in the HTML code after the `<body>` tag and the heading?

An additional line space was inserted for readability. This blank line will not be displayed on the Web page.

```

new 1 - Notepad - 
File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ? 
new 1 | 
1 <!DOCTYPE html 
2 PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" 
3 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"> 
4 
5 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en"> 
6 <head> 
7 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" /> 
8 <title>Hunger Fight</title> 
9 </head> 
10 <body> 
11 
12 <h1>Fight Against World Hunger</h1> ← main heading aligned left as default 
13 ← insertion point 
14 
15 </body> 
16 
17 </html> 
18

```

Figure 2–9

To Enter a Paragraph of Text

After you enter the heading, the next step is to add a paragraph of text using the `<p>` tag. When the browser finds a `<p>` tag in an HTML file, it starts a new line and inserts a blank line above the new paragraph. The `</p>` end tag indicates the end of the paragraph. When you enter this paragraph of text, do not press the ENTER key at the end of each line. Because Word wrap is turned on, your text will wrap to the next line even without pressing the ENTER key. Table 2–3 contains the HTML tags and text used in the paragraph.

Table 2–3 Adding a Paragraph of Text

Line HTML Tag and Text

- | | |
|----|--|
| 15 | <code><p>Our school and schools around the world have recently joined forces to fight against world hunger. The office for the Fight Against World Hunger team is located next to the bookstore. You can drop off food and money donations there. We also have donation drop-off locations at participating grocery stores. See the list of stores as well as additional information on the bulletin board outside our team office.</p></code> |
|----|--|

The following step illustrates how to enter a paragraph of text in an HTML file.

1

- With the insertion point on line 15, enter the HTML code, as shown in Table 2–3 on the previous page. Do not press ENTER at the end of each line when entering this text and use only one space after periods.
- Press the ENTER key twice to position the insertion point on line 17 (Figure 2–10).

Q&A

Why do you not press the ENTER key after each line of code in Table 2–3?

Because you turned on Word wrap right after you started Notepad++, the text that you enter as the paragraph will automatically wrap to the next line. The text goes to the end of the Notepad++ window and then wraps. If you had not turned on Word wrap, your text would continue scrolling to the right as you type, and text to the left would scroll off the screen. With Word wrap on, all text remains visible in the Notepad++ window.

```

1 <!DOCTYPE html
2 PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
6 <head>
7 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
8 <title>Hunger Fight</title>
9 </head>
10 <body>
11
12 <h1>Fight Against World Hunger</h1>
13
14
15 <p>Our school and schools around the world have recently joined forces to fight against world hunger. The office for the Fight Against World Hunger team is located next to the bookstore. You can drop off food and money donations there. We also have donation drop-off locations at participating grocery stores. See the list of stores as well as additional information on the bulletin board outside our team office.</p>
16
17 | insertion point
18 </p>
19 </body>
20 </html>

```

Figure 2–10

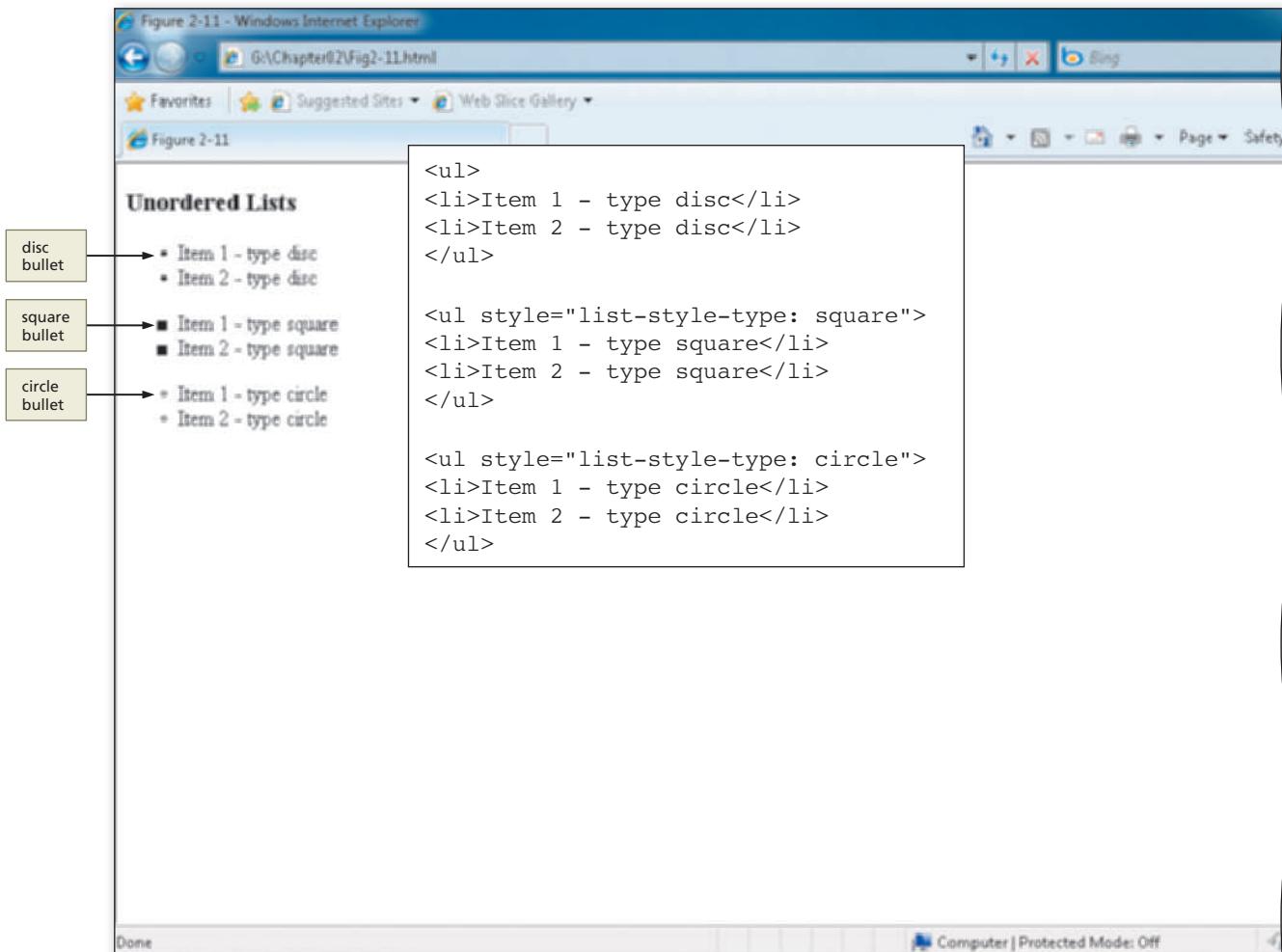
BTW

List Styles

It is helpful sometimes to structure the text of a Web page in a list. There are several list options that you can use. The Web page purpose determines which would be more effective. See Appendix D List Styles for style options that can be used with lists.

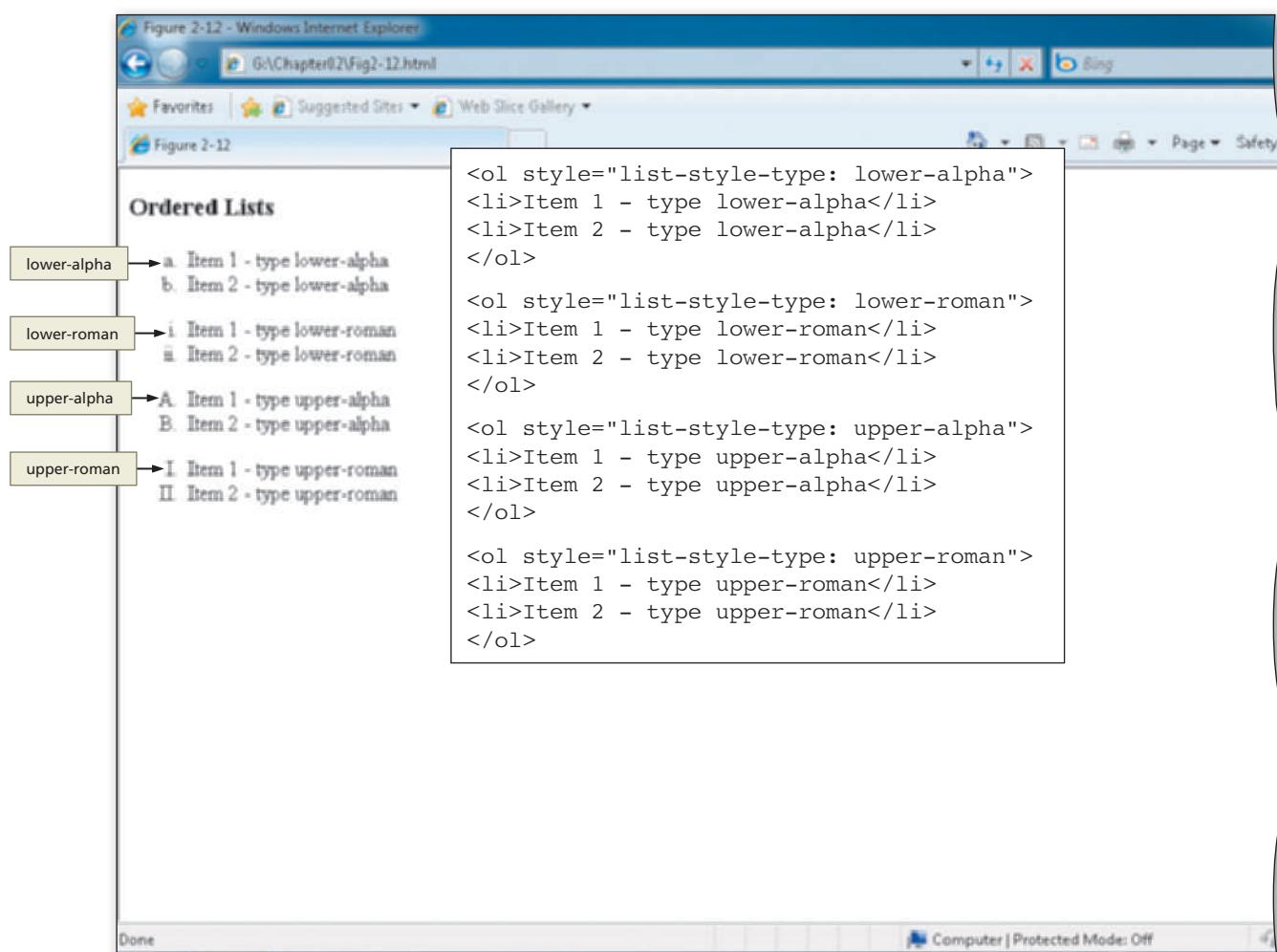
Using Lists to Present Content

Lists structure text into an itemized format. Typically, lists are bulleted (unordered) or numbered (ordered). An **unordered list**, which also is called a **bulleted list**, formats information using small images called bullets. Figure 2–11 shows Web page text formatted as unordered, or bulleted, lists and the HTML code used to create the lists.

**Figure 2-11**

An **ordered list**, which also is called a **numbered list**, formats information in a series using numbers or letters. An ordered list works well to organize items where order must be emphasized, such as a series of steps. Figure 2–12 on the next page shows Web page text formatted as ordered, or numbered, lists and the HTML tags used to create the lists.

The **** and **** tags must be at the start and end of an unordered or bulleted list. The **** and **** tags are used at the start and end of an ordered or numbered list. Unordered and ordered lists have optional bullet and number types. As shown in Figure 2–11, an unordered list can use one of three different bullet options: disc, square, or circle. If no type is identified, the default, disc, is used. An ordered list can use numbers, letters, or Roman numerals, as shown in Figure 2–12. The default option is to use Arabic numbers, such as 1, 2, and 3.

**Figure 2-12**

After the `` or `` tag is entered to define the type of list, the `` and `` tags are used to define a list item within an ordered or unordered list.

To Create an Unordered List

To highlight what Web site visitors can do to help the Fight Against World Hunger, you will create a bulleted (unordered) list using the HTML tags and text shown in Table 2–4. Remember that each list item must start with `` and end with ``.

Table 2–4 Adding an Unordered List

Line	HTML Tag and Text
17	<code><h2>What you can do to help:</h2></code>
18	
19	<code></code>
20	<code>Join the school's Fight Against World Hunger team with a \$10 donation</code>
21	<code>Buy requested food at participating grocery stores and leave at drop-off locations</code>
22	<code>Volunteer to collect money and donations around the community</code>
23	<code></code>

The following step illustrates how to create an unordered, or bulleted, list using the default bullet style.

1

- With the insertion point on line 17, enter the HTML code, as shown in Table 2–4. When you enter the text on lines 20, 21, and 22, make sure to press the TAB key at the start of the line, and then enter the text. Press ENTER at the end of each line.

- Press the ENTER key after typing line 23, leaving a blank line on 24 (Figure 2–13).

```

<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Hunger Fight</title>
</head>
<body>
<h1>Fight against World Hunger</h1>
<p>Our school and schools around the world have recently joined forces to fight against world hunger. The office for the Fight Against World Hunger team is located next to the bookstore. You can drop off food and money donations there. We also have donation drop-off locations at participating grocery stores. See the list of stores as well as additional information on the bulletin board outside our team office.</p>
<h2>What you can do to help:</h2>
<ul>
<li>Join the school's Fight Against World Hunger team with a $10 donation</li>
<li>Buy requested food at participating grocery stores and leave at drop-off locations</li>
<li>Volunteer to collect money and donations around the community</li>
</ul>
</body>
</html>

```

Figure 2–13

Q&A

Why do you press the TAB key at the start of the lines with the (list item) code?

Using the TAB key (to indent) when you enter list items helps format the text so that you can easily see that this text is different from the paragraph of text. Indenting text helps the Web developer see that certain segments of code are related to each other.

Q&A

What types of bullets will this list contain?

Because the code does not specify a type attribute, the list uses the default disc bullet. Other bullet options are square, circle, numbers, letters, and Roman numerals. You can also use a graphical image as a bullet, which you will do in later chapters.

More About List Formats

If you use the or start tags without attributes, you will get the default bullet (disc) or number style (Arabic numerals). To change the bullet or number type, the **list-style-type** property is entered within the or tags. To create a list with square bullets, you would type the line

```
<ul style="list-style-type: square">
```

as the inline style (CSS) code. You can find other list style properties and values in Appendix D.

In addition to ordered and unordered lists, there is a third kind of list, called a **definition list**, which offsets information in a dictionary-like style. Although they are used less often than unordered or ordered lists, definition lists are useful to create a glossary-like list of terms and definitions, as shown in Figure 2–14a on the next page. Figure 2–14b shows the HTML code used to create the definition list.

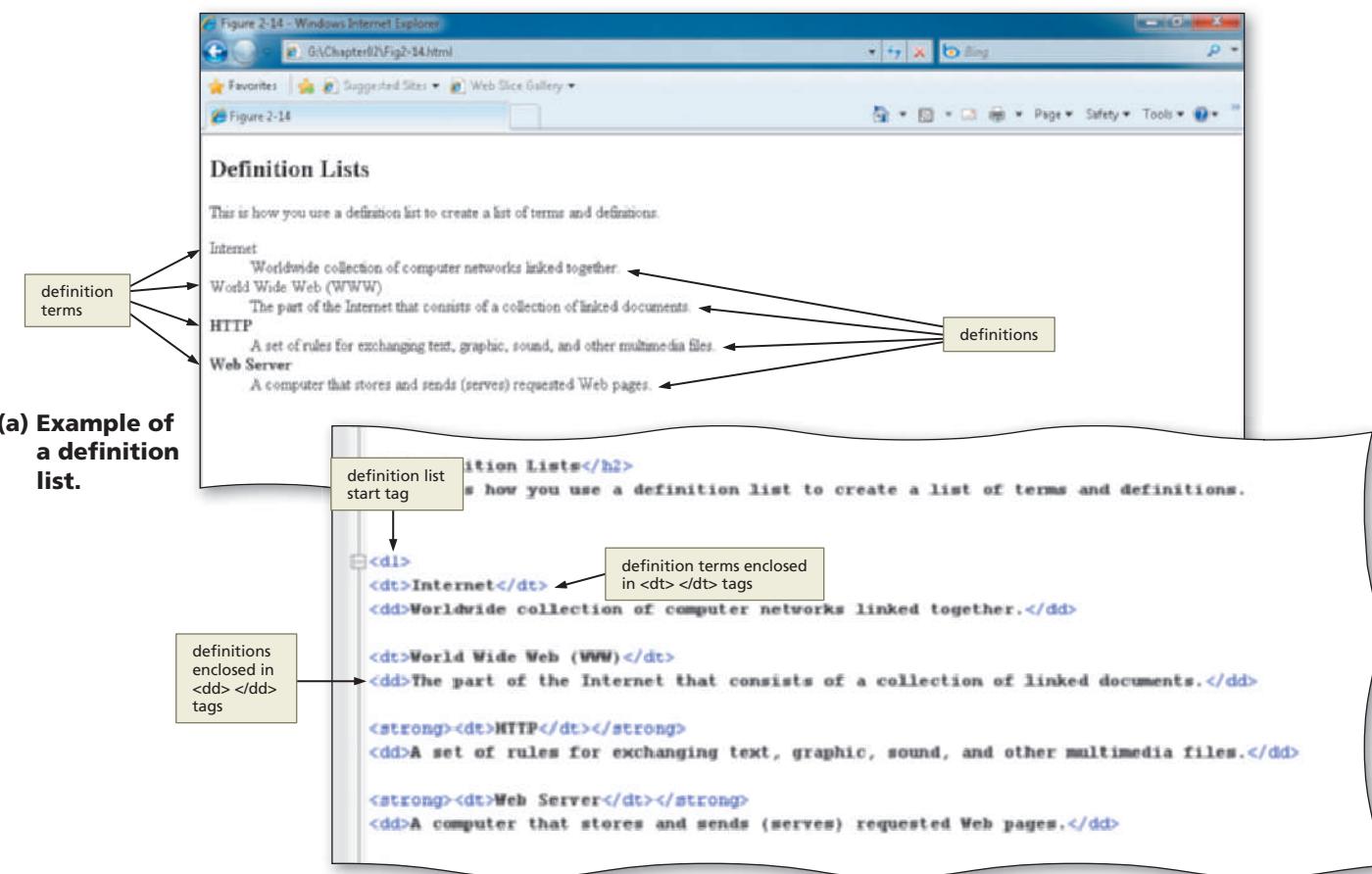


Figure 2-14

The syntax for definition lists is not as straightforward as the ``, ``, or `` structure that is used in the unordered and ordered list styles. With definition lists, you use the `<dl>` and `</dl>` tags to start and end the list. A `<dt>` tag indicates a term, and a `<dd>` tag identifies the definition of that term by offsetting the definition from the term. Table 2-5 lists the elements of a definition list and their purposes.

Table 2-5 Definition List Elements and Purposes

Definition List Element	Purpose
<code><dl> </dl></code>	Start and end a definition list
<code><dt> </dt></code>	Identify a term
<code><dd> </dd></code>	Identify the definition of the term directly above

As shown in Figure 2-14, by default, the definition term is left-aligned on the line and the definition for each term is indented so it is easily distinguishable as the definition for the term above it. In order to more clearly identify the definition term, you may want to make the term bold, as shown in the last two definitions (HTTP and Web Server) in Figure 2-14. You could do this by using an inline style that includes the following code:

```
<dt style="font-weight: bold">
```

Saving and Organizing HTML Files

Before you can see how your HTML file looks in a Web browser, you must save it. It also is important to save your HTML file for the following reasons:

- The document in memory will be lost if the computer is turned off or you lose electrical power while the text editor is open.
- If you run out of time before completing your project, you may finish your document at a future time without starting over.

To save your file, you use the Notepad++ File, Save command. When you save a file, you give your file a name and follow that with the file extension. As mentioned earlier in the book, file names should always make sense relative to their purpose. For instance, naming a file page1 does not indicate the purpose of that file. Naming the file hungerfight immediately identifies that this file has something to do with that topic. The Web page files in this book are always named with all lowercase letters and with no spaces. This is a standard that is followed throughout the book.

HTML files must end with an extension of **.htm** or **.html**. Many older Web page servers can only display pages with the .htm extension, or short file names (i.e., file names that are only up to eight characters in length). HTML files with an extension of .html can be viewed on Web servers running an operating system that allows long file names (i.e., file names that can be up to 255 characters in length). Almost all current operating systems allow long file names, including Windows 7, Windows Vista, Windows XP, Windows Server 2003/2008, Windows 2000, Macintosh, and Linux. For Web servers that run an operating system that does not accept long file names, you need the .htm extension. In this book, all files are saved using the .html extension.

You will use a very simple folder structure with all the projects in this book. It is therefore important to organize your files in folders so that all files for a project or end-of-chapter exercise, including HTML code and graphical images, are saved in the same folder. If you correctly downloaded the files from the Data Files for Students (see the inside back cover of this book), you will have the required file structure. When you initially save the hungerfight.html file, you will save it in the ChapterFiles subfolder of the Chapter02 folder. The graphical image used in Chapter 2, hungerfightlogo.jpg, will be stored in that same folder — Chapter02\ChapterFiles. Because the chapter projects in this book are relatively simple and use few images, images and HTML code are stored in the same folder. In real-world applications, though, hundreds or thousands of files might exist in a Web site, and it is more appropriate to separate the HTML code and graphical images into different subfolders. You will learn more about organizing HTML files and folders in Chapter 3.

Determine where to save the Web page.

When saving a Web page, you must decide which storage medium to use.

- If you always work on the same computer and have no need to transport your projects to a different location, then your computer's hard drive will suffice as a storage location. It is a good idea, however, to save a backup copy of your projects on a separate medium in case the file becomes corrupted or the computer's hard disk fails.
- If you plan to work on your projects in various locations or on multiple computers, then you should save your projects on a portable medium, such as a USB flash drive or CD. The projects in this book use a USB flash drive, which saves files quickly and reliably and can be reused. CDs are easily portable and serve as good backups for the final versions of projects because they generally can save files only one time.
- The above are general guidelines about saving your files. Your instructor may give you specific instructions for saving your work that differ from the steps that follow.

BTW

HTML File Names

HTML files have an extension of .html or .htm. The home page of a Web site is often called index.html, index.htm, default.html, or default.htm. Check with the service provider to find out which name they use.

BTW

Saving Your Work

It is a good idea to save your HTML file periodically as you are working to avoid the risk of losing your work completed thus far. You could get into the habit of saving your file after any large addition (i.e., a paragraph) of information. You might also want to save the file after typing in several HTML tags that would be difficult to re-do.

Plan Ahead

To Save an HTML File

You have performed many steps in creating this project and do not want to risk losing the work you have done so far. Also, to view HTML in a browser, you must save the file. The following steps show how to save an HTML file.

1

- With a USB flash drive connected to one of the computer's USB ports, click File on the Notepad++ menu bar (Figure 2–15).

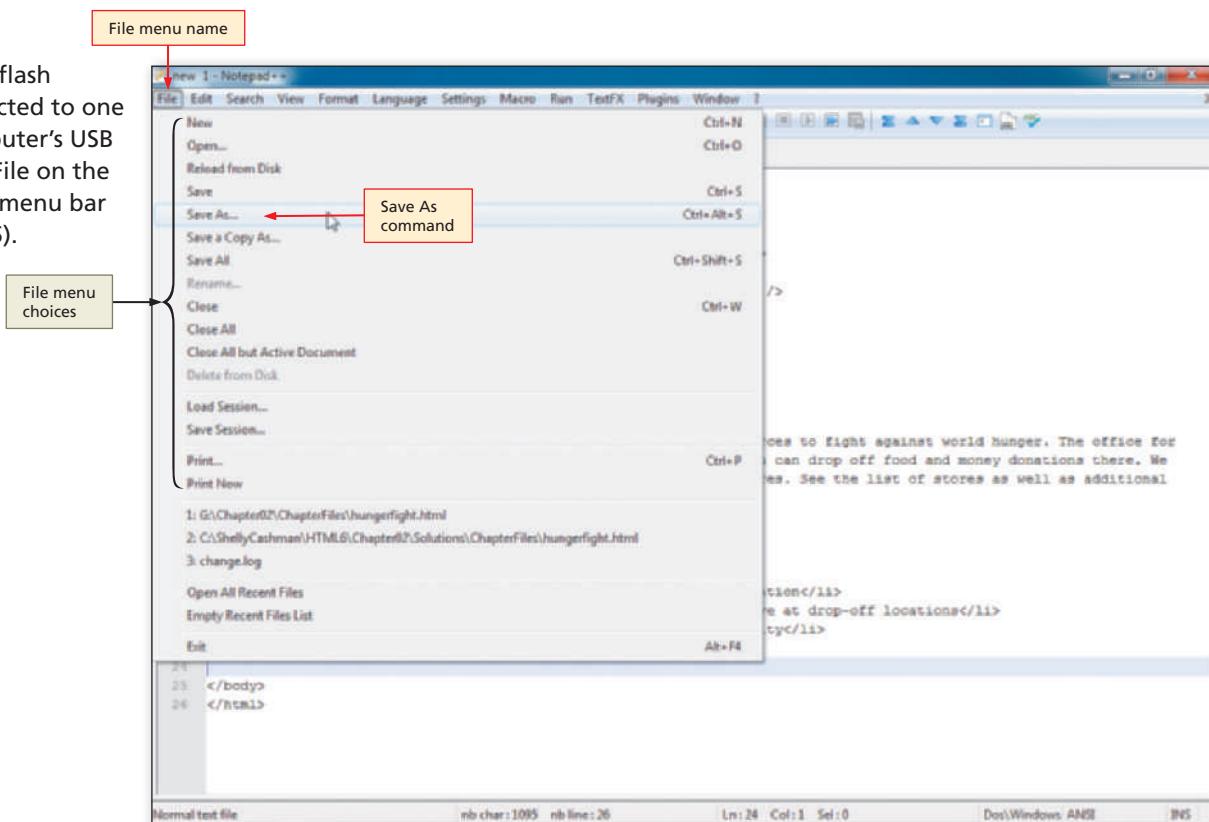


Figure 2–15

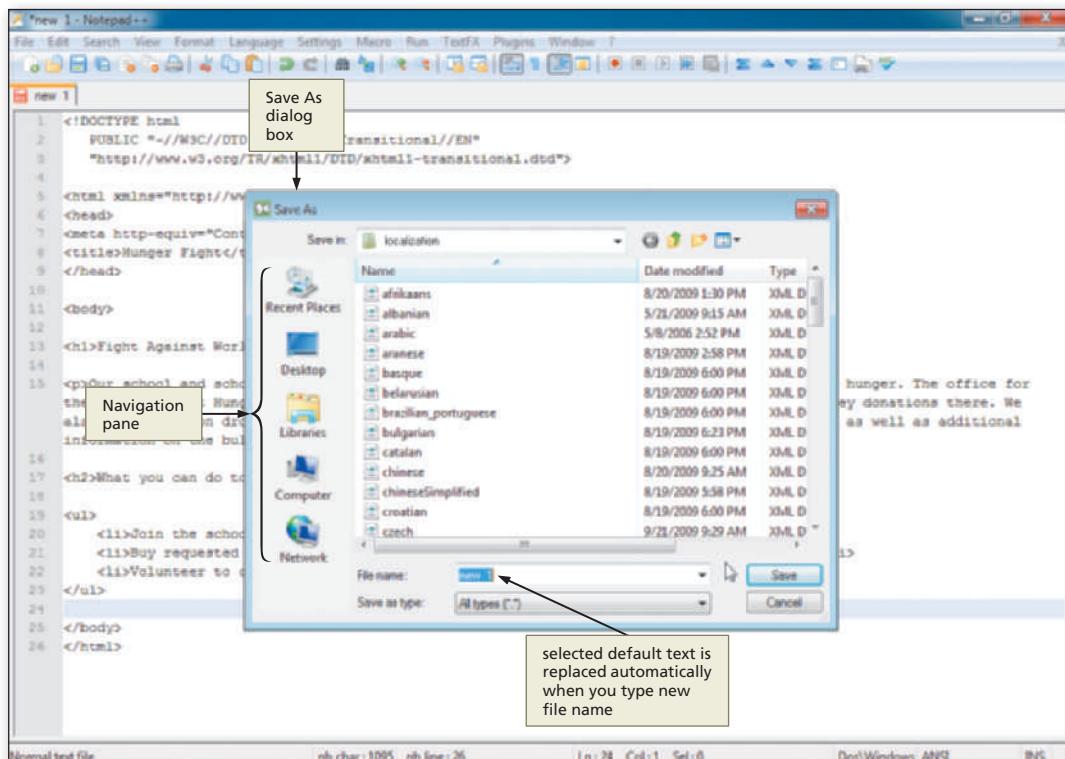
2

- Click Save As on the File menu to display the Save As dialog box (Figure 2–16).

Q&A

Do I have to save to a USB flash drive?

No. You can save to any device or folder. A folder is a specific location on a storage medium. Use the same process, but select your device or folder.



3

- Type hungerfight.html in the File name text box to change the file name. Do not press ENTER after typing the file name.
- Click Computer in the left side of the dialog box to display a list of available drives (Figure 2-17).
- If necessary, scroll until your USB flash drive, such as UDISK 2.0 (G:), appears in the list of available drives.

Q&A

Why is my list of files, folders, and drives arranged and named differently from those shown in the figure?

Your computer's configuration determines how the list of files and folders is displayed and how drives are named.

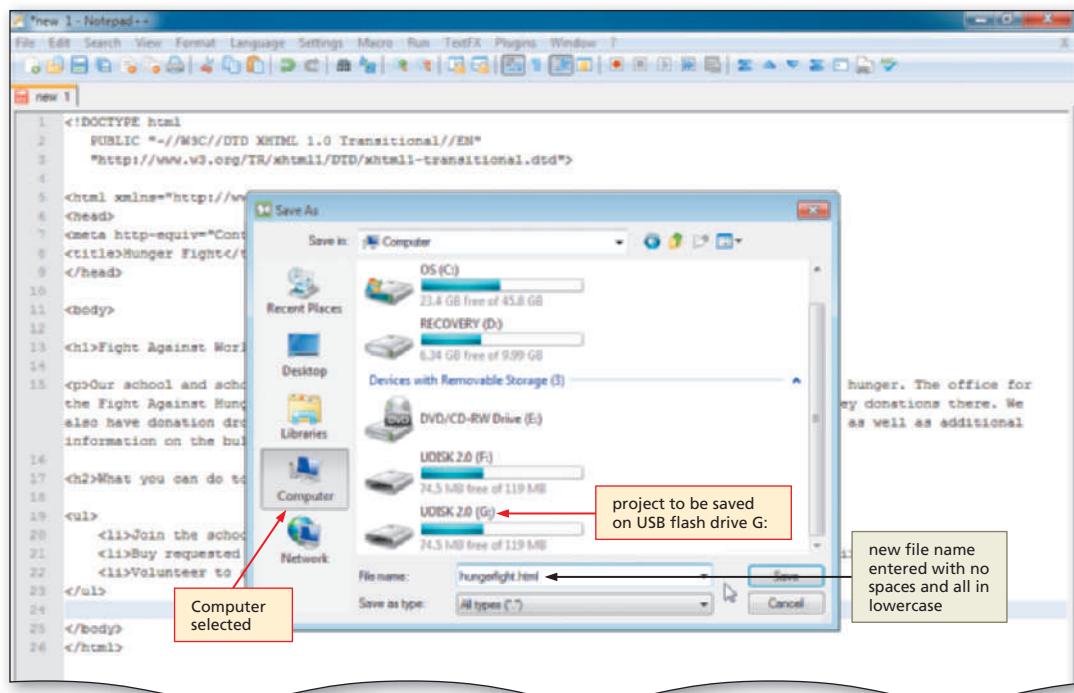
Q&A

How do I know the drive and folder in which my file will be saved?

Notepad++ displays a list of available drives and folders. You then select the drive and/or folder into which you want to save the file.

BTW**Storing Your Files**

Many schools provide students with space on a Web server to store their Web pages. However, saving your Web pages to another medium (e.g., a USB flash drive) assures that you have a backup copy of the files that you created. Saving Web page files to the hard drive on a computer in a school lab runs the risk of it not being there the next time you are in that lab. Many schools delete all files other than those installed by the lab technician at the start-up of each computer.

**Figure 2-17**

4

- Double-click UDISK 2.0 (G:) (or your storage device) in the Computer list to select the USB flash drive, drive G in this case, as the new save location.

Q&A

What if my USB flash drive has a different name or letter?

It is likely that your USB flash drive will have a different name and drive letter and be connected to a different port. Verify that the device in your Computer list is correct.

- If necessary, open the Chapter02\ChapterFiles folder (Figure 2-18).

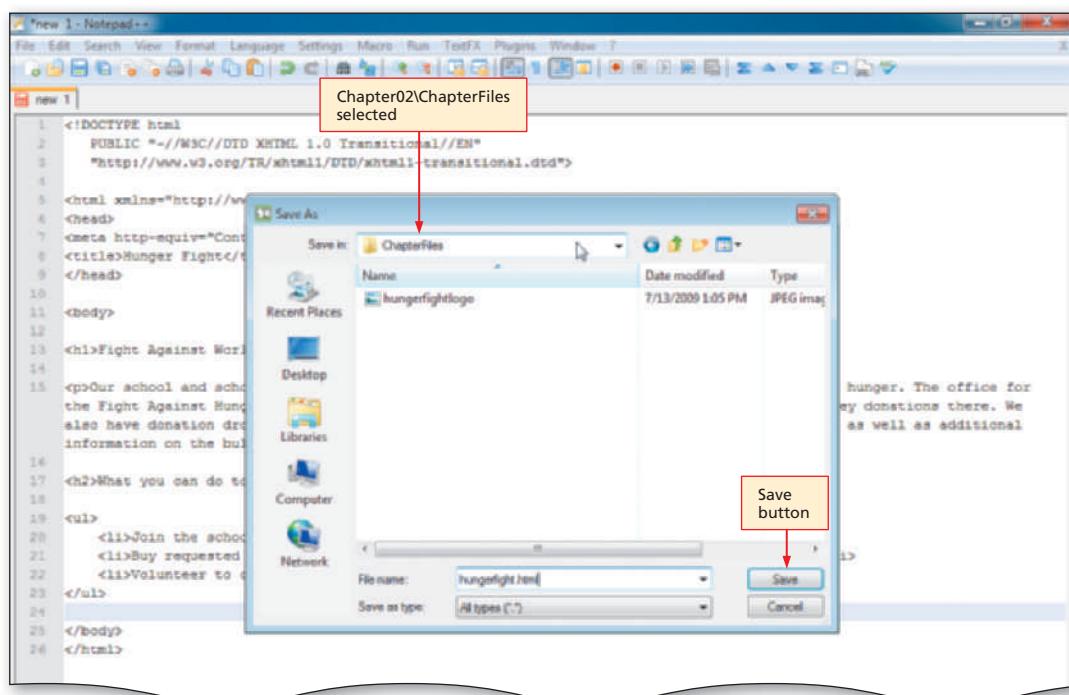


Figure 2-18

Q&A

What if my USB flash drive does not have a folder named Chapter02\ChapterFiles?

If you followed the steps to download the chapter files from the Data Files for Students, you should have a folder named Chapter02\ChapterFiles. If you do not, check with your instructor.

5

- Click the Save button in the Save As dialog box to save the file on the USB flash drive with the name hungerfight.html (Figure 2-19).

Q&A

Is my file only on the USB drive now?

No, although the HTML file is saved on a USB drive, it also remains in memory and is displayed on the screen (Figure 2-19). Notepad++ displays the new file name on the title bar and on the document tab.

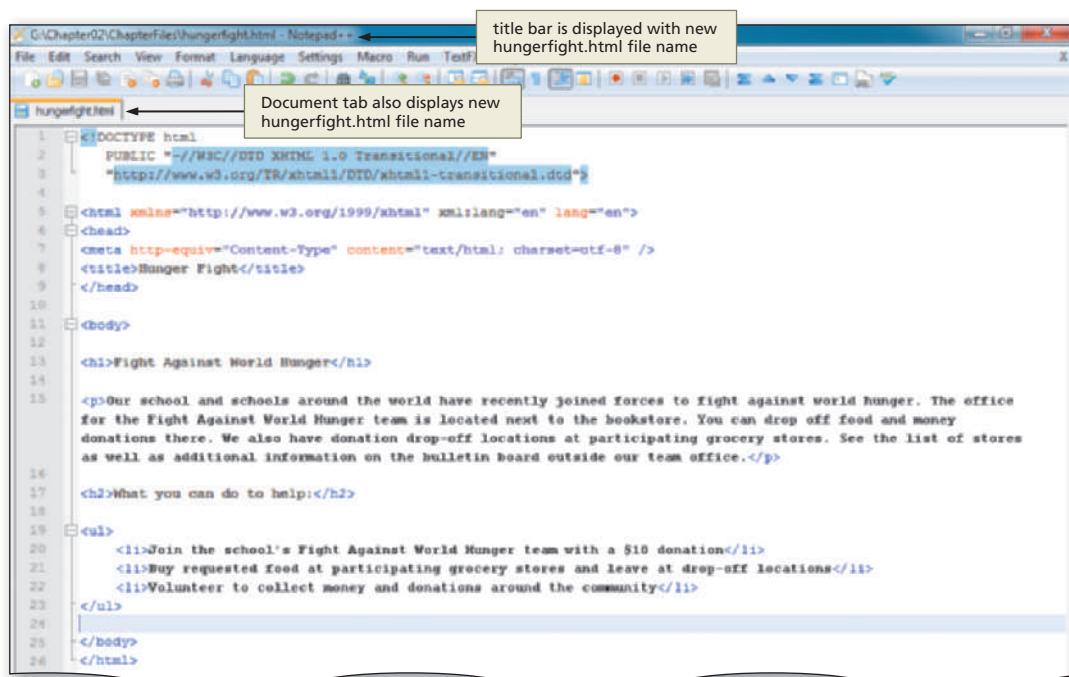


Figure 2-19

Other Ways

- Press CTRL+ALT+S, type the file name, click Computer,

select drive or folder, click the Save button

Using a Browser to View a Web Page

After entering code in the HTML file and saving it, you should view the Web page in a browser to see what the Web page looks like up to this point. The HTML file is displayed in the browser as if the file were available on the Web. In general, viewing the Web page periodically during development is good coding practice, because it allows you to see the effect of various HTML tags on the text and to check for errors in your HTML file.

If your computer is connected to the Internet when the browser window opens, it displays a **home page**, or **start page**, which is a Web page that appears each time Internet Explorer starts.

To Start a Browser

With the HTML file saved on the USB drive, the next step is to view the Web page using a browser. Because Windows is **multitasking**, you can have more than one program running at a time, such as Notepad++ and your browser. The following steps illustrate how to start a browser to view a Web page.

1

- Click the Internet Explorer icon on the taskbar (Figure 2–20).

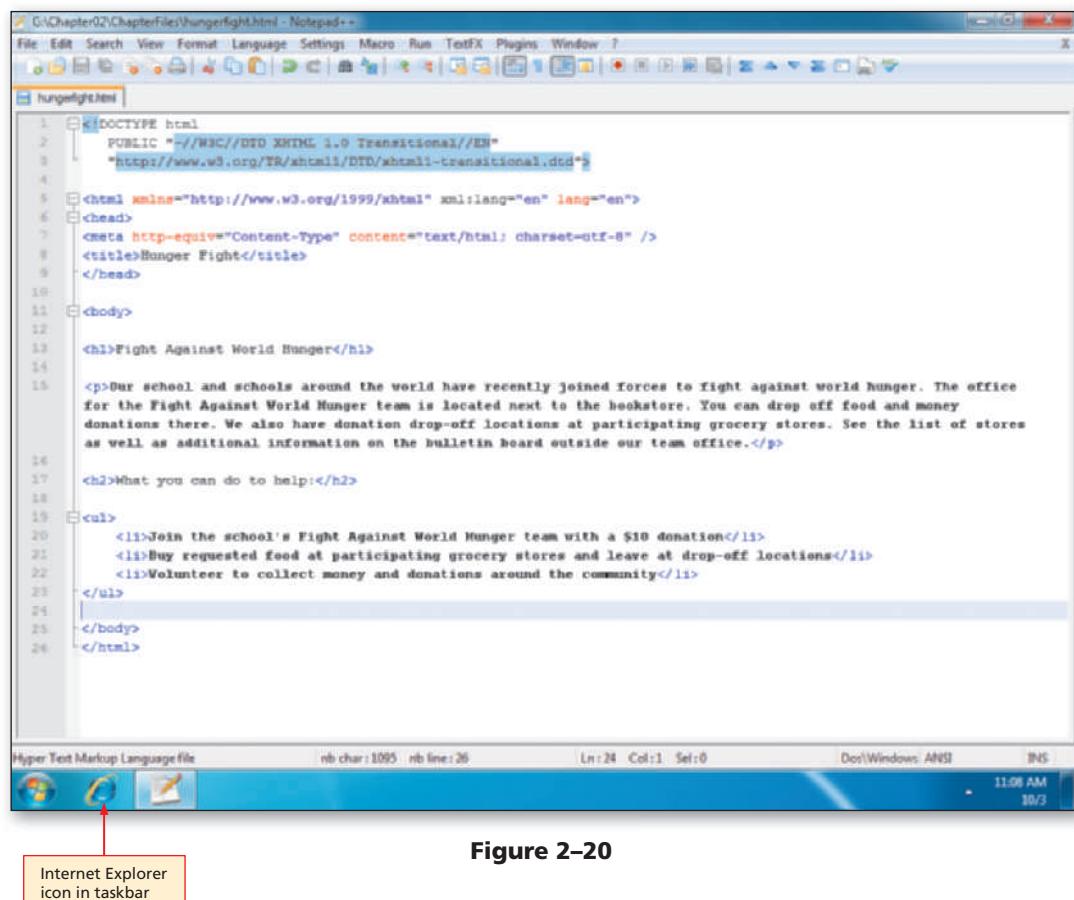


Figure 2–20

2

- If necessary, click the Maximize button to maximize the browser window (Figure 2–21).

Q&A

Why does my browser display a different window?

Because it is possible to

MSN home page; your home page may differ

customize browser settings to change the Web page that appears as the home page, the home page that is displayed by your browser may be different. Schools and organizations often set a main page on their Web sites as the home page for browsers installed on lab or office computers.

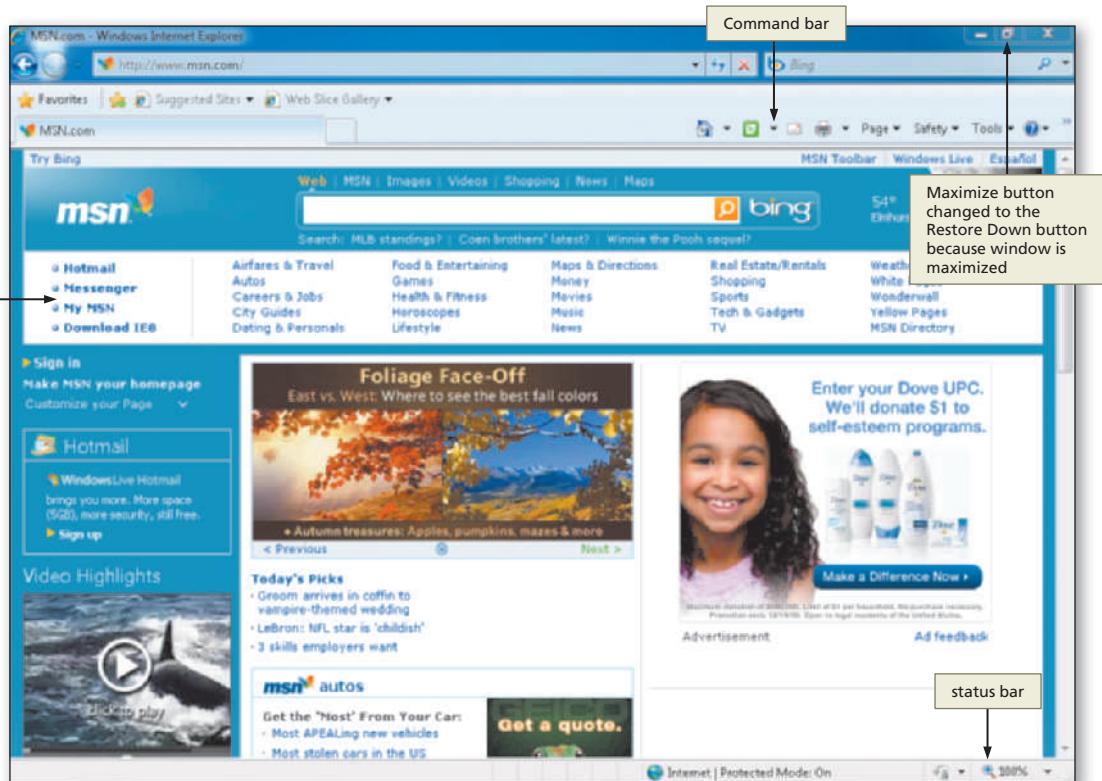


Figure 2–21

Other Ways

1. Click Start, All Programs, Internet Explorer
2. Double-click Internet Explorer icon on desktop, if one is present

BTW

Developing Web Pages for Multiple Browsers

When developing Web pages, you must consider the types of browsers visitors will use, including Internet Explorer and Mozilla Firefox for Windows or Safari or Internet Explorer for Mac OS. The Apple Web site provides suggestions for creating Web pages that will work in a wide range of browsers. Part of thorough testing includes bringing your Web pages up in different versions of different browsers.

To View a Web Page in a Browser

A browser allows you to open a Web file located on your computer and have full browsing capabilities, as if the Web page were stored on a Web server and made available on the Web. The following steps use this technique to view the HTML file, hungerfight.html, in a browser.

1

- Click the Address bar to select the URL on the Address bar.
- Type g:\Chapter02\ChapterFiles\hungerfight.html to display the new URL on the Address bar (Figure 2–22). The Web page is not displayed until you press the ENTER key, as shown in the next step.

Q&A

How can I correct the URL on the Address bar?

The URL is displayed on the Address bar. If you type an incorrect letter or symbol on the Address bar and notice the error before moving to the next step, use the BACKSPACE key to erase all the characters back to and including the one that is incorrect, and then continue typing.

Q&A

What if my file is in a different location?

You can type in the path to your file in the Address bar, or browse to your file, as shown in Other Ways on the next page.

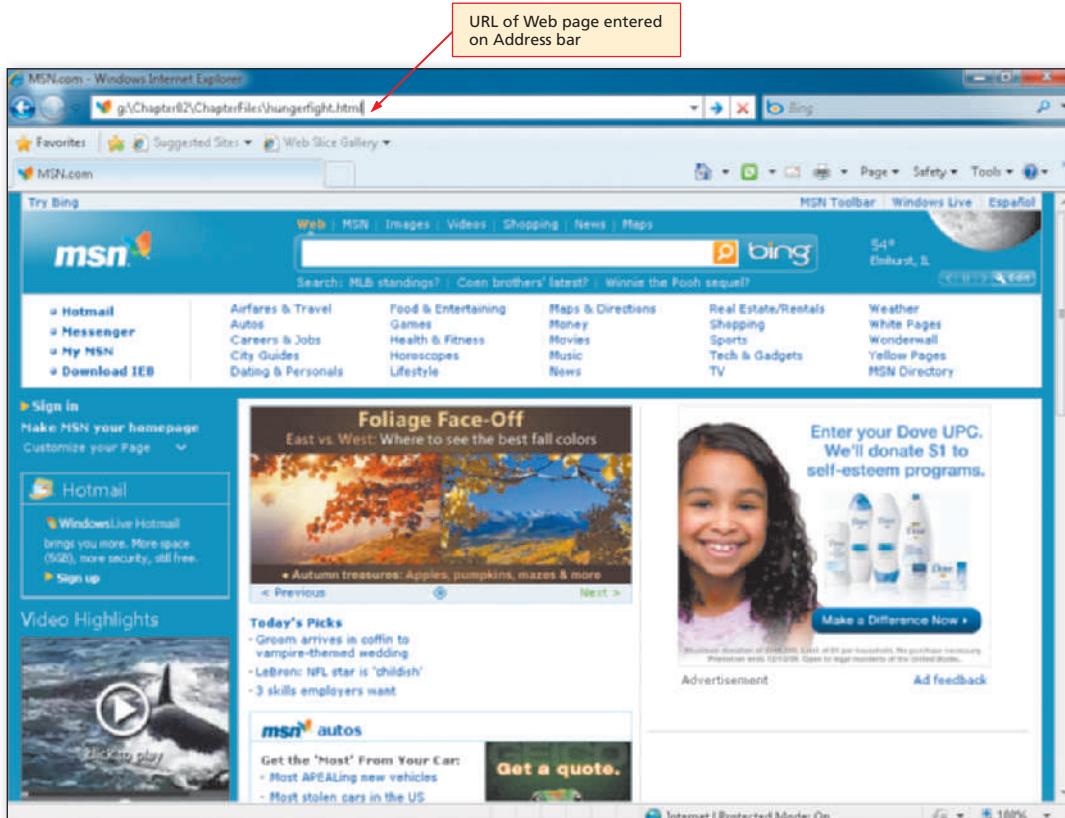


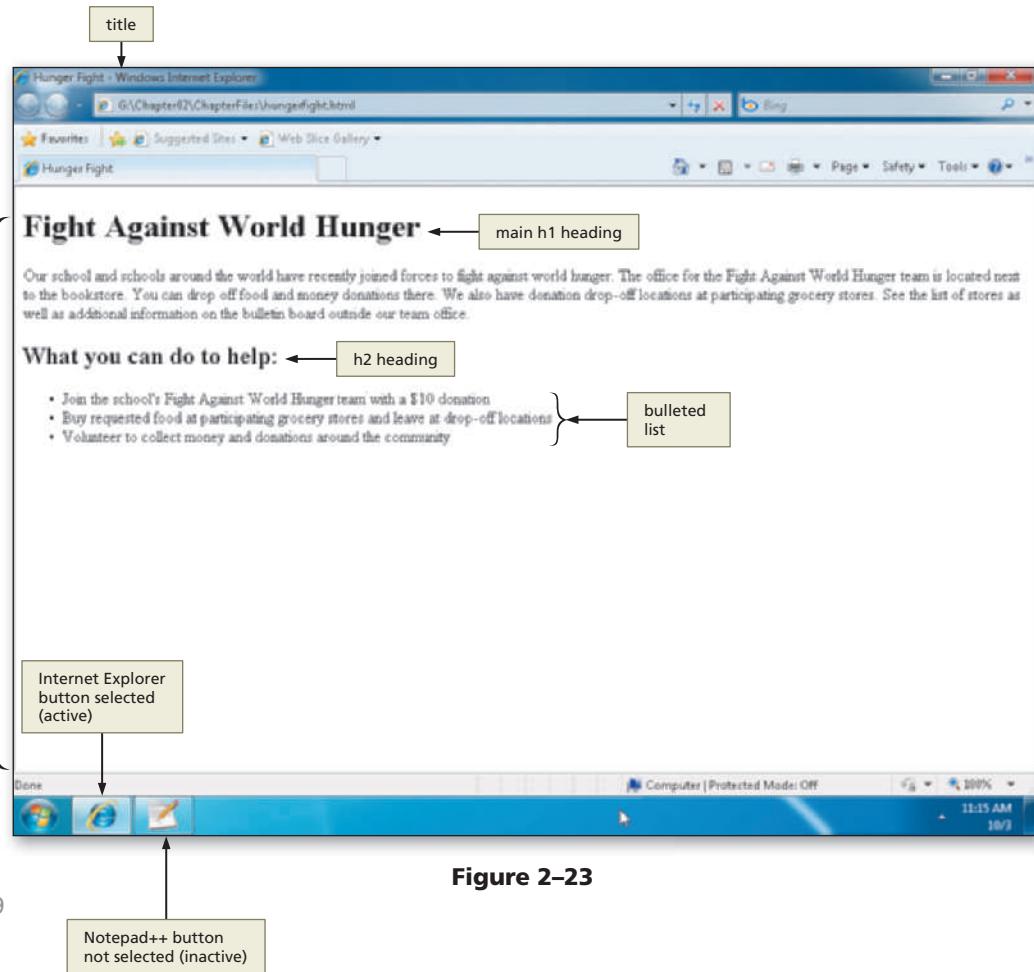
Figure 2–22

2

- Press the ENTER key to display the hungerfight.html page as if it were available on the Web (Figure 2-23).

Q&A
What if I get a warning from Internet Explorer (IE) that says, "Internet Explorer needs to open a new window to display this webpage."? If this happens, you should click the OK button to continue. You then will see your Web page displayed in another IE window.

Q&A
What if my page is not displayed correctly? Check your entries for hungerfight.html carefully against Figure 2-19 to make sure you have not made any typing errors or left anything out. Correct the errors, resave the file, and try again.

**Figure 2-23****Other Ways**

- In Windows Explorer, double-click HTML file name to open in default browser
- In Windows Explorer, right-click HTML file name, click Open with, click browser name
- Click Tools, Toolbars, Menu Bar if menu is not displayed; on Menu bar click File, Open, and browse to desired file

BTW

User Interface Design
The user interface design is a very important aspect of a Web site. If a site is designed poorly, users may be unable to find the desired information or complete a task, which makes the Web site ineffective. There are many good Web sites available that discuss Web design principles.

To Activate Notepad++

After viewing the Web page, you can modify the Web page by adding additional tags or text to the HTML file. To continue editing, you first must return to the Notepad++ window. The following step illustrates how to activate Notepad++.

1

- Click the Notepad++ button on the task-bar to maximize Notepad++ and make it the active window (Figure 2–24).

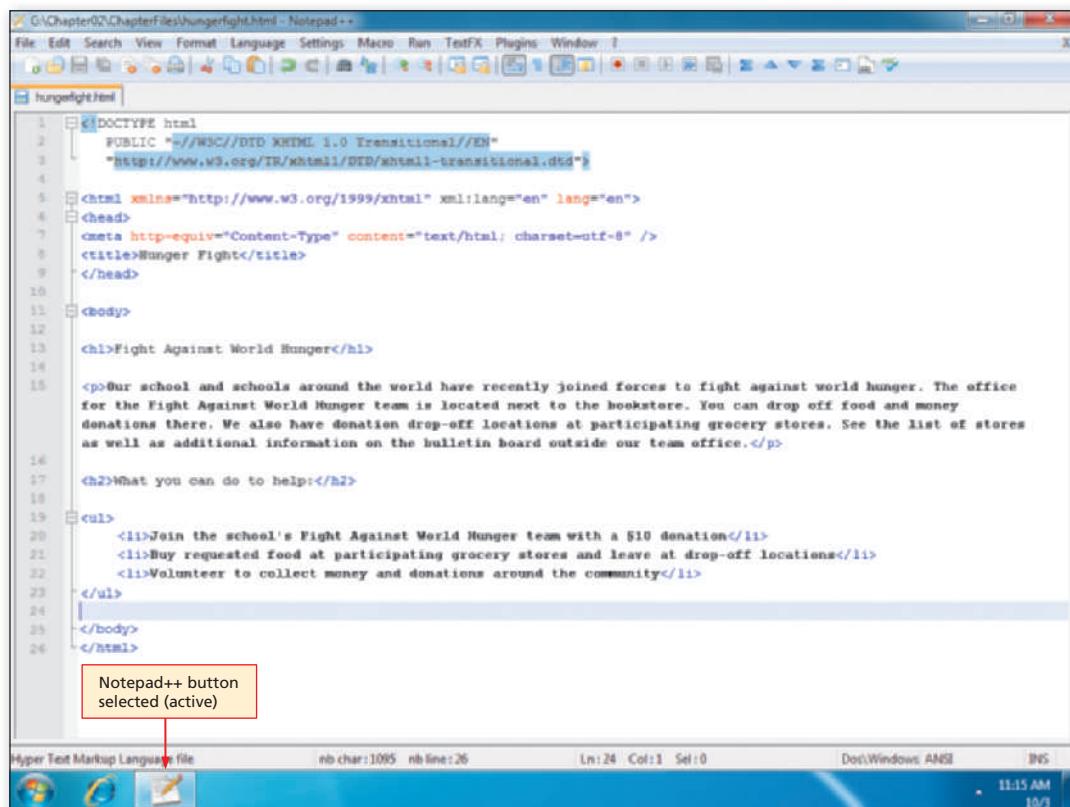


Figure 2–24

Improving the Appearance of Your Web Page

One goal in Web page development is to create a Web page that is visually appealing and maintains the interest of the visitors. The Web page developed thus far in the chapter is functional, but lacks visual appeal. In this section, you will learn how to improve the appearance of the Web page from the one shown in Figure 2–25a to the one shown in Figure 2–25b by adding an image, adding a color to the headings, adding a horizontal rule, and changing the font style of the bulleted list. Many of these tasks can be accomplished by using style sheets.

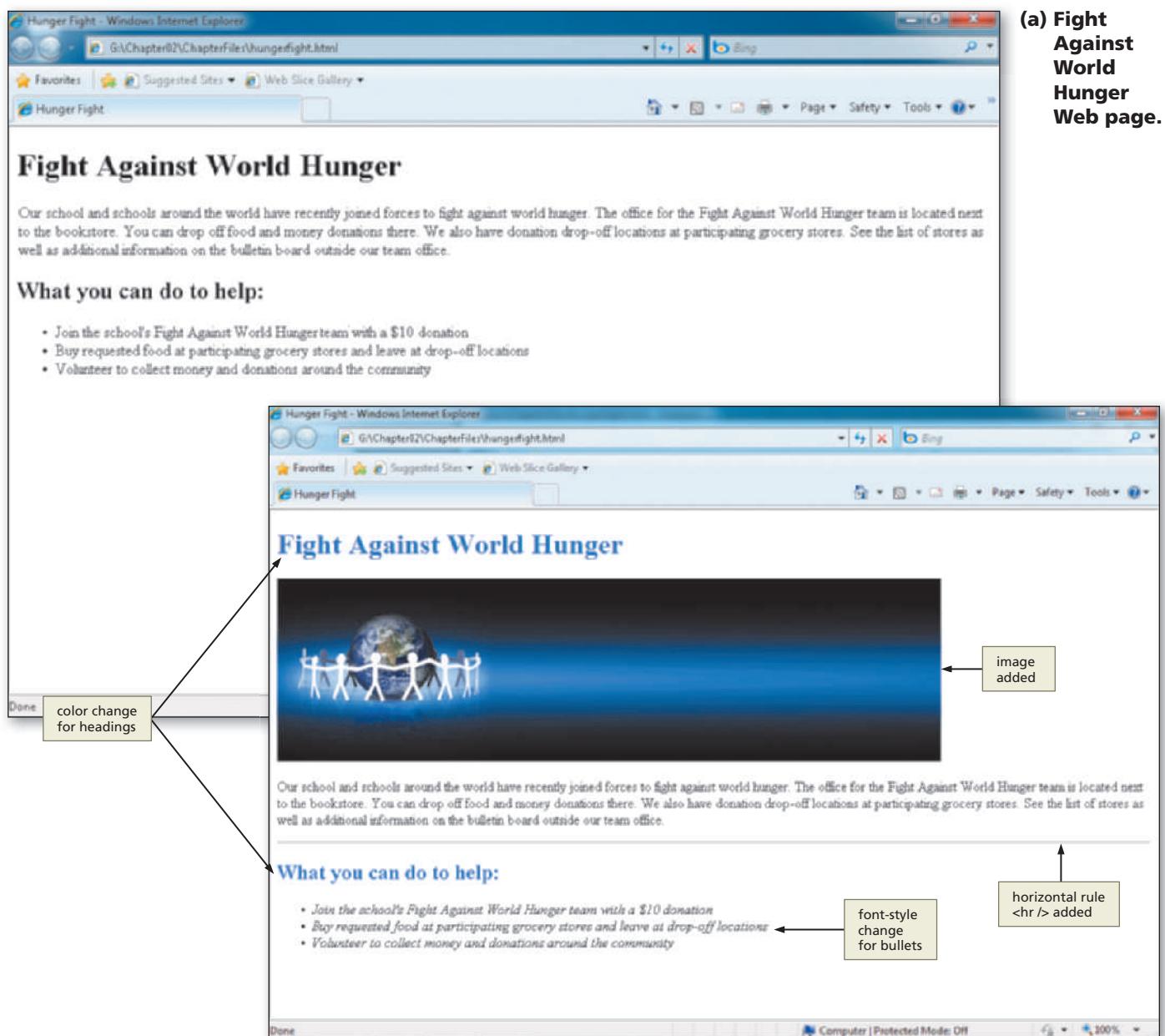


Figure 2-25

Using Style Sheets

Although HTML allows Web developers to make changes to the structure, design, and content of a Web page, HTML is limited in its ability to define the appearance, or style, across one or more Web pages. As a result, style sheets were created.

As a review, a **style** is a rule that defines the appearance of an element on a Web page. A **style sheet** is a series of rules that defines the style for a Web page or an entire Web site. With a style sheet, you can alter the appearance of a Web page or pages by changing characteristics such as font family, font size, margins, and link specifications.

CSS supports three types of style sheets: inline, embedded, and external (or linked). With an **inline style**, you add a style to an individual HTML tag, such as a heading

or paragraph. The style changes that specific tag, but does not affect other tags in the document. With an **embedded style sheet**, you add the style sheet within the <head> tags of the HTML document to define the style for an entire Web page. With a linked style sheet, or **external style sheet**, you create a text file that contains all of the styles you want to apply, and save the text file with the file extension .css. You then add a link to this external style sheet on any Web page in the Web site. External style sheets give you the most flexibility and are ideal to apply the same formats to all of the Web pages in a Web site. External style sheets also make it easy to change formats quickly across Web pages. You will use inline styles in this chapter's project to enhance the styles of the headings (change the color) and the bulleted list (change the font style).

Style Sheet Precedence As shown in Table 2–6, the three style sheets supported by CSS control the appearance of a Web page at different levels. Each style sheet type also has a different level of precedence or priority in relationship to the others. An external style sheet, for example, is used to define styles for multiple pages in a Web site. An embedded style sheet is used to change the style of one Web page, but overrides or takes precedence over any styles defined in an external style sheet. An inline style is used to control the style within an individual HTML tag and takes precedence over the styles defined in both embedded and external style sheets.

BTW**Inline Styles**

Using an inline style is helpful when you want to alter the appearance (or style) of a single HTML element. Appendix D contains the Cascading Style Sheet Properties and Values supported by most browsers. The inline styles used in this chapter can be found in the appendix. For more information on inline styles, look at w3.org.

Table 2–6 Style Sheet Precedence

Type	Level and Precedence
Inline	<ul style="list-style-type: none"> To change the style within an individual HTML tag Overrides embedded and external style sheets
Embedded	<ul style="list-style-type: none"> To change the style of one Web page Overrides external style sheets
External	<ul style="list-style-type: none"> To change the style of multiple pages in a Web site

Because style sheets have different levels of precedence, all three types of style sheets can be used on a single Web page. For example, you may want some elements of a Web page to match the other Web pages in the Web site, but you also may want to vary the look of certain sections of that Web page. You can do this by using the three types of style sheets.

Style Statement Format No matter what type of style sheet you use, you must use a **style statement** to define the style. The following code shows an example of a style statement used in an inline style:

```
<h1 style="font-family: Garamond; font-color: navy">
```

A style statement is made up of a selector and a declaration. The part of the style statement that identifies the page elements is called the **selector**. In this example, the selector is h1 (header size 1). The part of the style statement that identifies how the element(s) should appear is called the **declaration**. In this example, the declaration is everything between the quotation marks: the font-family and font-color properties and their values (Garamond and navy, respectively). A declaration includes at least one type of style, or **property**, to apply to the selected element. Examples of properties include color, text-indent, border-width, and font-style. For each property, the declaration includes a related **value**, which specifies the display parameters for that specific property.

Each property accepts specific values, based on the styles that property can define. The property, font-color, for example, can accept the value, navy, but cannot accept the value, 10%, because that is not a valid color value. In the next section of this chapter, you will change the heading color to blue for both the h1 and h2 headings. Using inline styles in this case is appropriate because there are only two headings to change on the Web page. If you had many headings to change, an embedded or external style sheet would be more appropriate. This will be discussed in later chapters.

Inline Styles An inline style is used to define the style of an individual HTML tag. For example, to change the style of a single paragraph, you could add an inline style with the <p> (paragraph) tag as the selector and a declaration that defines new font style and style values for that paragraph, as shown here:

```
<p style="font-style: italic; font-size: 8pt">
```

Because they take precedence over the other types of style sheets and affect the style for individual HTML tags, inline styles are helpful when one section of a Web page needs to have a style different from the rest of the Web page. In this chapter's project, an inline style is used to change the color of the headings on the Web page and to change the font style in the bulleted list.

Now that you understand how style sheets and inline styles function, it is time to think about adding an image to enhance the appearance of your Web page.

Plan Ahead

Find appropriate graphical images.

To use graphical images, also called graphics, on a Web page, the image must be stored digitally in a file. Files containing graphical images are available from a variety of sources:

- Microsoft has free digital images on the Web for use in a document. Other Web sites also have images available, some of which are free, while others require a fee.
- You can take a picture with a digital camera and **download** it, which is the process of copying the digital picture from the camera to your computer.
- With a scanner, you can convert a printed picture, drawing, or diagram to a digital file.

If you receive a picture from a source other than yourself, do not use the file until you are certain it does not contain a virus. A **virus** is a computer program that can damage files and programs on your computer. Use an antivirus program to verify that any files you use are virus free.

Establish where to position and how to format the graphical image. The content, size, shape, position, and format of a graphic should capture the interest of passersby, enticing them to stop and read the Web page. Often, the graphic is the center of attraction and visually the largest element on a page. If you use colors in the graphical image, be sure they are part of the Web page's color scheme.

Identify the width and height of the image. The width and height (measured in pixels) of an image should always be identified in the tag. These dimensions are used by the browser to determine the size to display the image. If you do not identify those attributes, the browser has to determine the size. This slows the process down for the browser.

Provide alternate text for the image. Alternate text should always be used for each image. This text is especially useful to users with visual impairments who use a screen reader, which translates information on a computer screen into audio output. The length of the alternate text should be reasonable.

Web Page Images

Images are used in many ways to enhance the look of a Web page and make it more interesting and colorful. Images can be used to add background color, to help organize a Web page, to help clarify a point being made in the text, or to serve as links to other Web pages. Images also are often used to break up Web page sections (such as with a horizontal rule) or as directional elements that allow a visitor to navigate a Web site.

Using Web Page Divisions

It is sometimes helpful to break up your Web page into divisions (or sections), which allows you to apply styles to different Web page elements. Throughout this book, you always use the start `<div>` and end `</div>` division tags as a container in which to insert images. Utilizing division tags allows you to add styles such as centering your image or adding background color to your images.

Image Types

Web pages use three types of files as images: GIF, JPEG, and PNG (Table 2–7). **Graphics Interchange Format (GIF)** files have an extension of .gif. A graphic image saved as a GIF (pronounced *jiff* or *giff*) uses compression techniques, called LZW compression, to make it smaller for download on the Web. Standard (or noninterlaced) GIF images are displayed one line at a time when loading. Interlaced GIF images load all at once, starting with a blurry look and becoming sharper as they load. Using interlaced GIFs for large images is a good technique, because a Web page visitor can see a blurred outline of the image as it loads. GIF is a patented format, however, and therefore widespread use is limited.

A second type of image file is **Portable Network Graphics (PNG)**, which has a .png or .ping extension. The PNG (pronounced *ping*) format also is a compressed file format that supports multiple colors and resolutions. The World Wide Web Consortium developed the PNG format as a graphics standard and patent-free alternative to the GIF format. Most newer browsers support PNG images.

Finally, **Joint Photographic Experts Group (JPEG)** files have an extension of .jpg, .jpe, or .jpeg. A JPEG (pronounced *JAY-peg*) is a graphic image saved using a compression technique other than LZW. JPEG files often are used for more complex images, such as photographs, because the JPEG file format supports more colors and resolutions than the other file types.

BTW

Images

Images on Web pages are viewed in a variety of environments, including slow connections to the Internet and slower computers. Optimizing your images is important to increase the speed of download for all of your Web page visitors. Search the Web for more information on image optimization.

Table 2–7 Image Types and Uses

Image Type	Use
Graphics Interchange Format (GIF)	<ul style="list-style-type: none"> Use for images with few colors (<256) Allows for transparent backgrounds
Portable Network Graphics (PNG)	<ul style="list-style-type: none"> Newest format for images Use for all types of images Allows for variation in transparency
Joint Photographic Experts Group (JPEG)	<ul style="list-style-type: none"> Use for images with many colors (>256), such as photographs

If an image is not in one of these formats, you can use a paint or graphics-editing program to convert an image to a .gif, .jpg, or .png format. Some paint programs even allow you to save a GIF image as interlaced. A number of paint and graphics-editing programs, such Adobe Photoshop and Corel Paint Shop Pro, are available in the marketplace today.

BTW**Overusing Images**

Be cautious about overusing images on a Web page. Using too many images may give your Web page a cluttered look or distract the visitor from the purpose of the Web page. An image should have a purpose, such as to convey content, visually organize a page, provide a hyperlink, or serve another function.

Image Attributes

You can enhance HTML tags by using attributes. **Attributes** define additional characteristics for the HTML tag. For instance, you should use the width and height attributes for all tags. Table 2–8 lists the attributes that can be used with the tag. In this chapter, the src and alt attributes are used in the tag. Image attributes will be explained in detail, because they are used in later chapters.

Table 2–8 Image Attributes

Attribute	Function
alt	<ul style="list-style-type: none"> Alternative text to display when an image is being loaded Especially useful for screen readers, which translate information on a computer screen into audio output Should be a brief representation of the purpose of the image Generally should stick to 50 characters or fewer
height	<ul style="list-style-type: none"> Defines the height of the image, measured in pixels Improves loading time
hspace	<ul style="list-style-type: none"> Defines the horizontal space that separates the image from the text
src	<ul style="list-style-type: none"> Defines the URL of the image to be loaded
vspace	<ul style="list-style-type: none"> Defines the vertical space that separates the image from the text
width	<ul style="list-style-type: none"> Defines the width of the image, measured in pixels Improves loading time

To Add an Image

In the early days when the Web was used mostly by researchers needing to share information with each other, having purely functional, text-only Web pages was the norm. Today, Web page visitors are used to a more graphically oriented world, and have come to expect Web pages to use images that provide visual interest. The following step illustrates how to add an image to a Web page by entering an tag in the HTML file using the tags and text shown in Table 2–9.

Table 2–9 Adding an Image

Line	HTML Tag and Text
15	<div>
16	
17	</div>

1

- Click the blank line 14 and then press the ENTER key.
- With the insertion point on line 15, enter the HTML code, as shown in Table 2–9. Press ENTER at the end of each line (Figure 2–26).

Q&A

What is the purpose of the alt attribute?

The alt attribute has three important purposes. First, screen readers used by users with visual impairments read the alternate text out loud. Second, the alternate text is displayed while the image is being loaded.

Finally, the alt tag is required for XHTML compliance.

begin <div> and </div> tags

insert image

alt attribute and text show what text is displayed with mouse over

width and height attributes and sizes

```

<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml/DTD/xhtml-transitional1.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset= utf-8" />
<title>Hunger Fight</title>
</head>
<body>
<h1>Fight Against World Hunger</h1>
<div>

</div>
<p>Our school and schools around the world have recently joined forces to fight against world hunger. The office for the Fight Against World Hunger team is located next to the bookstore. You can drop off food and money donations there. We also have donation drop-off locations at participating grocery stores. See the list of stores as well as additional information on the bulletin board outside our team office.</p>
<h2>What you can do to help:</h2>
<ul>
<li>Join the school's Fight Against World Hunger team with a $10 donation</li>
<li>Buy requested food at participating grocery stores and leave at drop-off locations</li>
<li>Volunteer to collect money and donations around the community</li>
</ul>
</body>
</html>

```

Figure 2–26**Q&A**

What is the purpose of the <div> and </div> tags?

The <div> tag establishes a division within a Web page.

When you use the <div> </div> tags, you are able to design a layout that uses CSS, including inserting images.

Make other visual enhancements.

In addition to images, there are several ways to add visual interest.

Add color to headings. Web developers often use colors to call attention to elements on a Web page. The color selected should coordinate with the images selected for the page. It also should allow the Web page text to be read easily. Some colors, such as dark colors, may not be appropriate because the default black text cannot be displayed effectively on a dark background. When changing the color of an element such as a heading, it is usually best to apply the same style to all headings on the Web page for consistency.

Change the font style of a bulleted list. It is sometimes aesthetically pleasing to change the style of the text in a bulleted list. When you want to call attention to the information, you might want to italicize or bold the bullet text.

Insert a horizontal rule. It is useful to use a horizontal rule to break up text on a Web page. A horizontal rule is used as a divider for a page to separate text sections.

Plan Ahead

BTW

Colors

Figure 2–27 does not list all possible Web colors. Many other colors are available that you can use for Web page backgrounds or text fonts. For more information about colors, see Appendix B or search the Web for browser colors.

Other Visual Enhancements

One way to help capture a Web page visitor's attention is to use color. Many colors are available for use as a Web page background, text, or link. Figure 2–27 shows colors often used on Web pages, with the corresponding six-digit number codes. The six-digit number codes can be used to specify a color for a background, text, or links. The headings on the Fight Against World Hunger Web page are currently black (the default color). You will spruce up the Web page by adding color to the headings using inline styles.

COLORS IN HEX									
Here is a table of common colors with their hexadecimal equivalents. Use the codes to define the desired color for the background, text, or links.									
#ffc6a5	#ff9473	#ff6342	#ff3118	#ff0000	#d60000	#ad0000	#840000	#630000	
#ffe7c6	#ffce9c	#ffb573	#ff9c4a	#ff8429	#d66321	#ad4a18	#844d18	#632910	
#ffffc6	#fffff9c	#fffff6b	#ffff42	#ffff10	#d6c610	#ad9410	#847308	#635208	
#f7ffce	#eefef ad	#e7f784	#def763	#d6ef39	#b5bd31	#8c9429	#6b6b21	#524a18	
#de93bd	#c6ef8c	#adde63	#94d639	#7bc618	#639c18	#527b10	#425a10	#314208	
#ceefbd	#a5de94	#7bc66b	#52b552	#299c39	#218429	#186321	#184a18	#103910	
#c6e7de	#94d6ce	#63bdb5	#31ada5	#089494	#087b7b	#006363	#004a4a	#003139	
#c6eff7	#94d6e7	#63c6de	#31b5d6	#00a5c6	#0084a5	#006b84	#005263	#00394a	
#bdc6de	#949cce	#6373b5	#3152a5	#083194	#082984	#08296b	#08215a	#00184a	
#c6b5de	#9c7bbd	#7b52a5	#522994	#31007b	#29006b	#21005a	#21004a	#180042	
#debddc	#ce84c6	#b552ad	#9c2994	#8c007b	#730063	#5a0052	#4a0042	#390031	
#f7bdde	#e78cc6	#de5aad	#d63194	#ce007b	#a50063	#840052	#6b0042	#520031	
#ffffff	#e0e0e0	#fbfbfb	#a1a1a1	#808080	#616161	#404040	#212121	#000000	
THE SIXTEEN PREDEFINED COLORS									
(Because these colors belong to the RGB spectrum, they will look a bit different on-screen.)									
silver	gray	maroon	green	navy	purple	olive	teal		
white	black	red	lime	blue	magenta	yellow	cyan		

Figure 2–27

BTW

Browser-safe Colors

Web developers used to have to make sure that they used browser-safe colors (Appendix B). The trend for monitors today is to display "true color" which means that any of 16 million colors can be displayed on the monitor. Few people use 8-bit monitors anymore, so you generally do not have to limit yourself to browser-safe colors.

The color codes and names shown in Figure 2–27 can be used for background, text, and link colors. The color property is used in the `<h1>` and `<h2>` tags to specify the color for these headings. The color `#3D60B1` will be used for these headings because it is one of the colors found in the graphical image inserted in the steps above.

Another way to visually enhance the Web page is to change the style of some of the text. This calls attention to that particular text on the Web page. In this section, you change the text in the bulleted list to an italic style using an inline style. This change helps call attention to the three points in that bulleted list.

Finally, you add a horizontal rule to further enhance this Web page. As discussed earlier in the chapter, horizontal rules are lines that act as dividers on a Web page to provide a visual separation of sections on the page. You can use an inline image to add a horizontal rule, or you can use the horizontal rule tag (`<hr />`) to add a simple horizontal rule, as shown in the following steps. Figure 2–28 shows examples of a variety of horizontal rules and the HTML code used to add them. The default horizontal rule is shown in the first rule on the page. Dimension is added to a horizontal rule by increasing the number of pixels that are displayed.

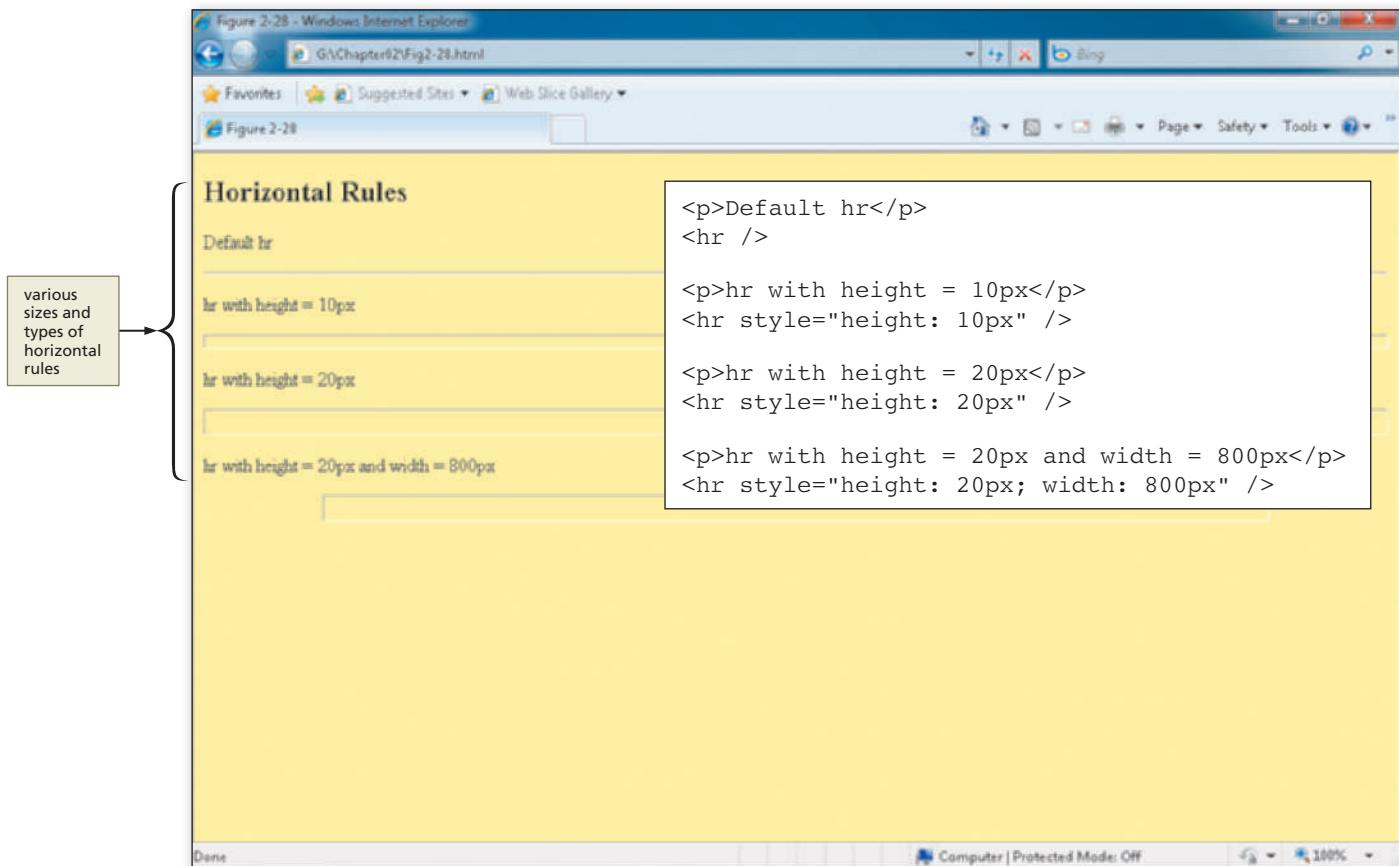


Figure 2–28

To Add Color to Web Page Headings

To change the color of headings on a Web page, the **color** property must be added in the **<h1>** and **<h2>** tags of the HTML file. The **color** property lets you change the color of various elements on the Web page. The following step shows how to add a color using the **color** property in an inline style.

1

- Click after the “1” but before the closing bracket in **<h1>** on line 13 and then press the SPACEBAR.
- Type **style="color: #3D60B1"** as the color code for the Fight Against World Hunger heading (Figure 2-29).
- Click after the “2” but before the closing bracket in **<h2>** on line 21 and then press the SPACEBAR.
- Type **style="color: #3D60B1"** as the color code for the What you can do to help: heading (Figure 2-29).

color code
#3D60B1
added to
headings
in <h1>
and <h2>
tags

 tag on line 13. Another red box highlights the same style in the second

tag on line 21. A callout box points from the text 'color code #3D60B1 added to headings in <h1> and <h2> tags' to these highlighted areas."/> ``` <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd"> <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en"> <head> <meta http-equiv="Content-Type" content="text/html; charset=utf-8" /> <title>Hunger Fight</title> </head> <body> <h1 style="color: #3D60B1">Fight Against World Hunger</h1> <div> </div> <p>Our school and schools around the world have recently joined forces to fight against world hunger. The office for the Fight Against World Hunger team is located next to the bookstore. You can drop off food and money donations there. We also have donation drop-off locations at participating grocery stores. See the list of stores as well as additional information on the bulletin board outside our team office.</p> <h2 style="color: #3D60B1">What you can do to help:</h2> Join the school's Fight Against World Hunger team with a $10 donation Buy requested food at participating grocery stores and leave at drop-off locations Volunteer to collect money and donations around the community </body> </html> ```

Figure 2-29

Q&A

Can I use any hexadecimal code or color name to change colors of headings?

Although you may use any of the hexadecimal codes or color names available, you have to make sure that the color is appropriate for the headings of your Web page. You do not want a heading that is too light in color or otherwise diminishes the headings.

Q&A

Could I have used the copy/paste process to copy the inline style from the **<h1>** tag to the **<h2>** tag?

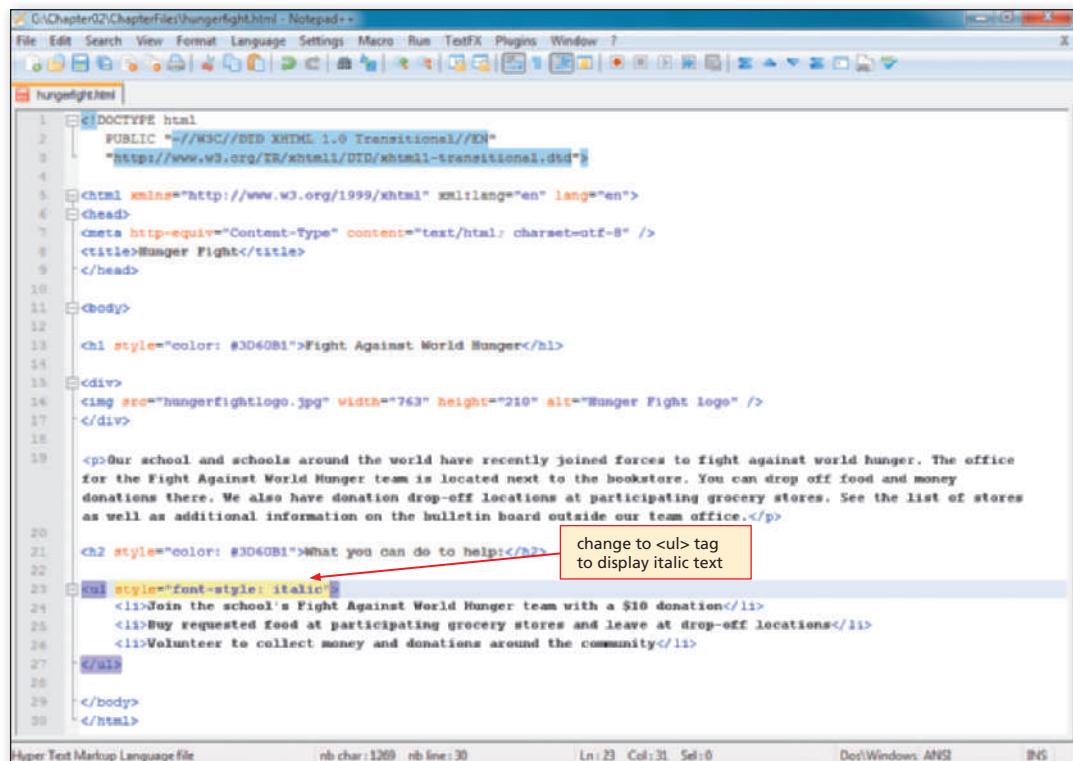
Yes, copying/pasting would have the same result. However, if you want to copy a style to all the headings on a Web page, it is easier to use embedded or external style sheets, which you learn about in the next two chapters.

To Change the Bulleted List Style

To change the style of the bulleted list, you again use an inline style with the font-style property. The **font-style** property lets you change the style of the font, or text, selected. There are three values for the font-style, normal, italic, and oblique, although only normal and italic are widely supported by the browsers of various properties of elements on the Web page. The following step shows how to add a color using the color property in an inline style.

1

- Click after the “|” but before the closing bracket in `` on line 23 and then press the SPACEBAR.
- Type `style="font-style: italic"` as the code (Figure 2–30).



```

1 <!DOCTYPE html
2 PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
6 <head>
7 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
8 <title>Hunger Fight</title>
9 </head>
10
11 <body>
12
13 <h1 style="color: #3D60B1">Fight Against World Hunger</h1>
14
15 <div>
16 
17 </div>
18
19 <p>Our school and schools around the world have recently joined forces to fight against world hunger. The office for the Fight Against World Hunger team is located next to the bookstore. You can drop off food and money donations there. We also have donation drop-off locations at participating grocery stores. See the list of stores as well as additional information on the bulletin board outside our team office.</p>
20
21 <h2 style="color: #3D60B1">What you can do to help:</h2>
22
23 <ul style="font-style: italic">
24 <li>Join the school's Fight Against World Hunger team with a $10 donation</li>
25 <li>Buy requested food at participating grocery stores and leave at drop-off locations</li>
26 <li>Volunteer to collect money and donations around the community</li>
27
28 </ul>
29
30 </body>
31 </html>

```

Hyper Text Markup Language file Line char : 1269 Line : 30 Ln: 23 Col: 31 Sel: 0 Dos/Windows ANSI ANSI

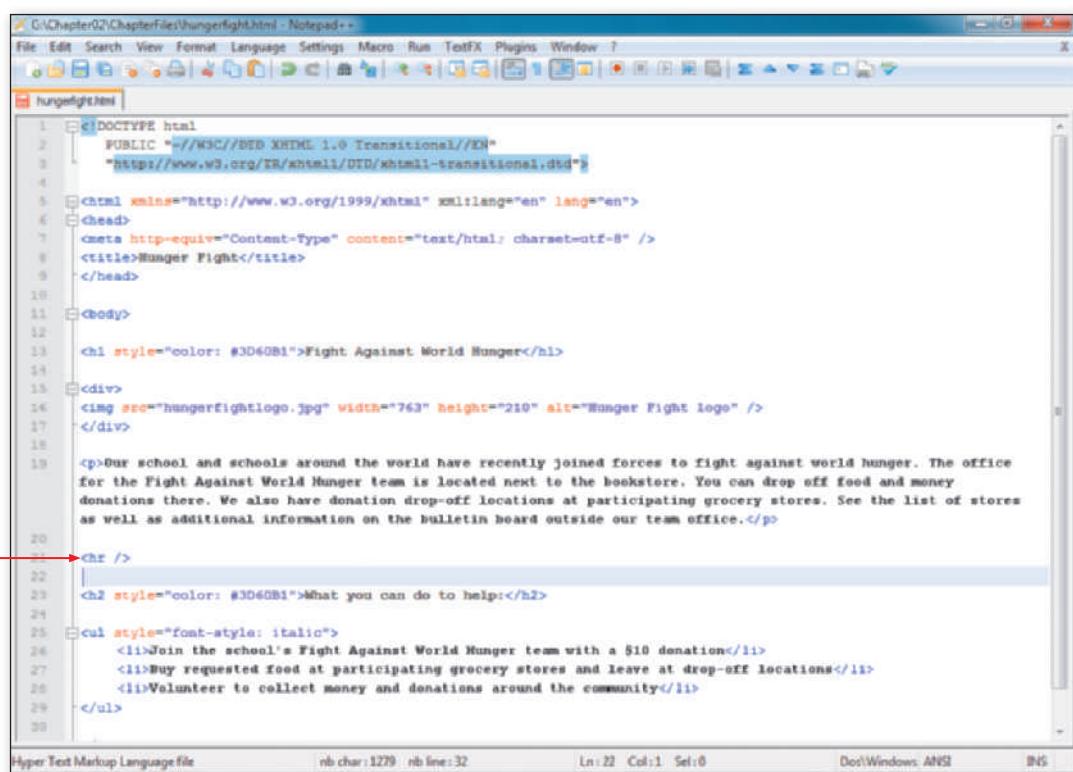
Figure 2–30

To Add a Horizontal Rule

The following step illustrates how to add a horizontal rule to a Web page.

1

- Click the blank line 20 and then press the ENTER key.
- Type `<hr />` as the HTML tag and then press the ENTER key (Figure 2–31).
- Click File on the menu bar and then click Save.



```

1 <!DOCTYPE html
2   PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
6   <head>
7     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
8     <title>Hunger Fight</title>
9   </head>
10
11 <body>
12
13   <h1 style="color: #3D60B1">Fight Against World Hunger</h1>
14
15   <div>
16     
17   </div>
18
19   <p>Our school and schools around the world have recently joined forces to fight against world hunger. The office for the Fight Against World Hunger team is located next to the bookstore. You can drop off food and money donations there. We also have donation drop-off locations at participating grocery stores. See the list of stores as well as additional information on the bulletin board outside our team office.</p>
20
21   <hr />
22
23   <h2 style="color: #3D60B1">What you can do to help:</h2>
24
25   <ul style="font-style: italic">
26     <li>Join the school's Fight Against World Hunger team with a $10 donation</li>
27     <li>Buy requested food at participating grocery stores and leave at drop-off locations</li>
28     <li>Volunteer to collect money and donations around the community</li>
29   </ul>
30

```

Figure 2–31

BTW

HTML and XHTML Tags
The Web has excellent sources that list HTML and XHTML tags. For more information about HTML and XHTML, search for “HTML tags” or “XHTML tags” in a search engine.

To Refresh the View in a Browser

As you continue developing the HTML file in Notepad++, it is a good idea to view the file in your browser as you make modifications. Be sure to click the Refresh button when viewing the modified Web page in the browser, to ensure the latest version of the Web page is displayed. The step on the next page shows how to refresh the view of a Web page in a browser in order to view the modified Web page.

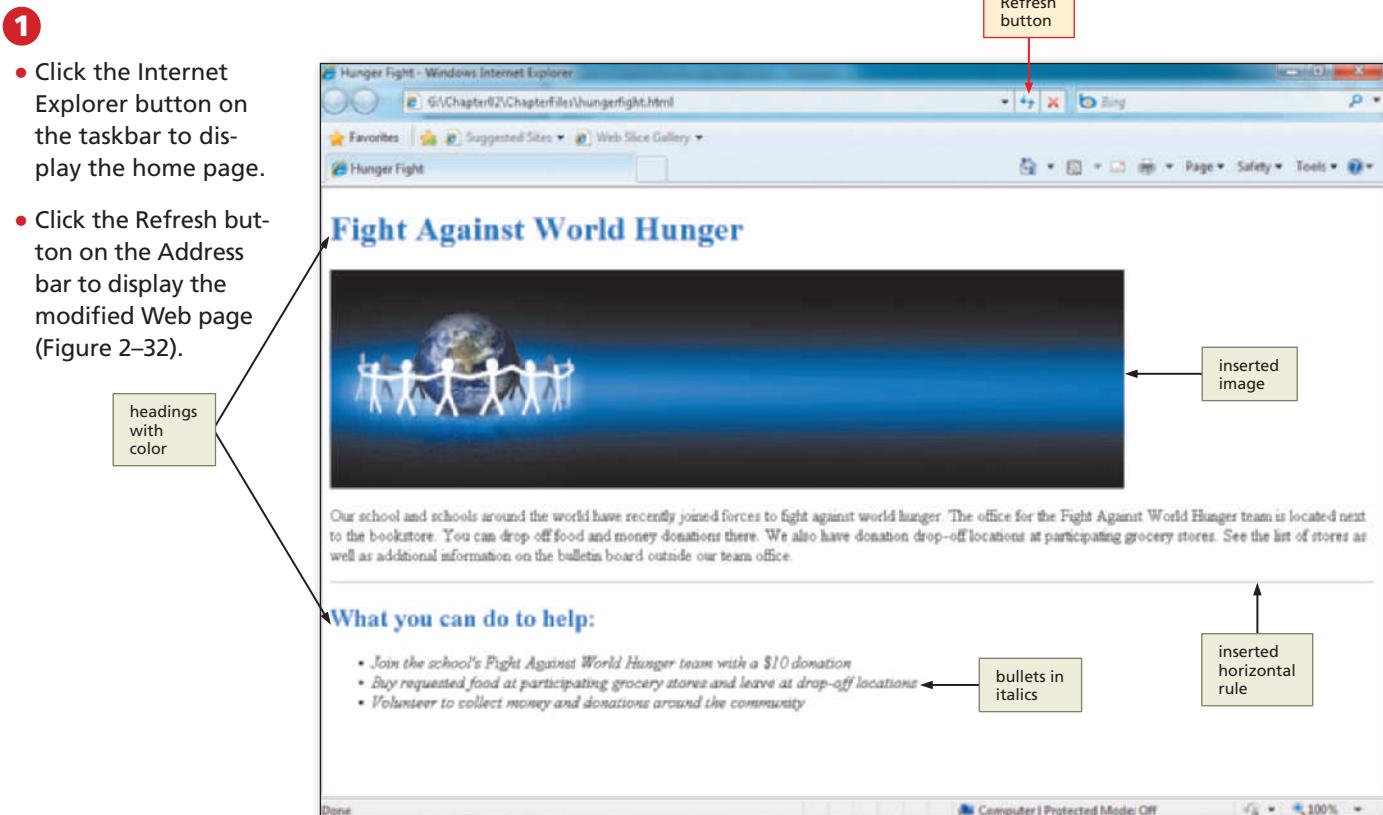


Figure 2–32

Other Ways

1. In Internet Explorer, press F5 to refresh

Validating and Viewing HTML Code

In Chapter 1, you read about validating your HTML code. Many validation services are available on the Web that can be used to assure that your HTML code follows standards. This should always be a part of your Web page testing. The validation service used in this book is the W3C Markup Validation Service (validator.w3.org). The XHTML validator looks at the DOCTYPE statement to see which version of HTML or XHTML you are using, and then checks to see if the code is valid for that version. In this chapter, the project uses Transitional code.

If validation detects an error in your HTML code, you see the warning “Errors found while checking this document as XHTML 1.0 Transitional!” in the header bar, which is red (Figure 2–33a on the next page). The Result line shows the number of errors that you have. You can scroll down the page or click the Jump To: Validation Output link to see detailed comments on each error.

It is important to note that one error can result in more errors. As an example, the `<hr />` tag in the hungerfight.html file was changed to `<hr>` to show code with an error. Figure 2–33b shows that in this case, one initial error resulted in a total of four errors.

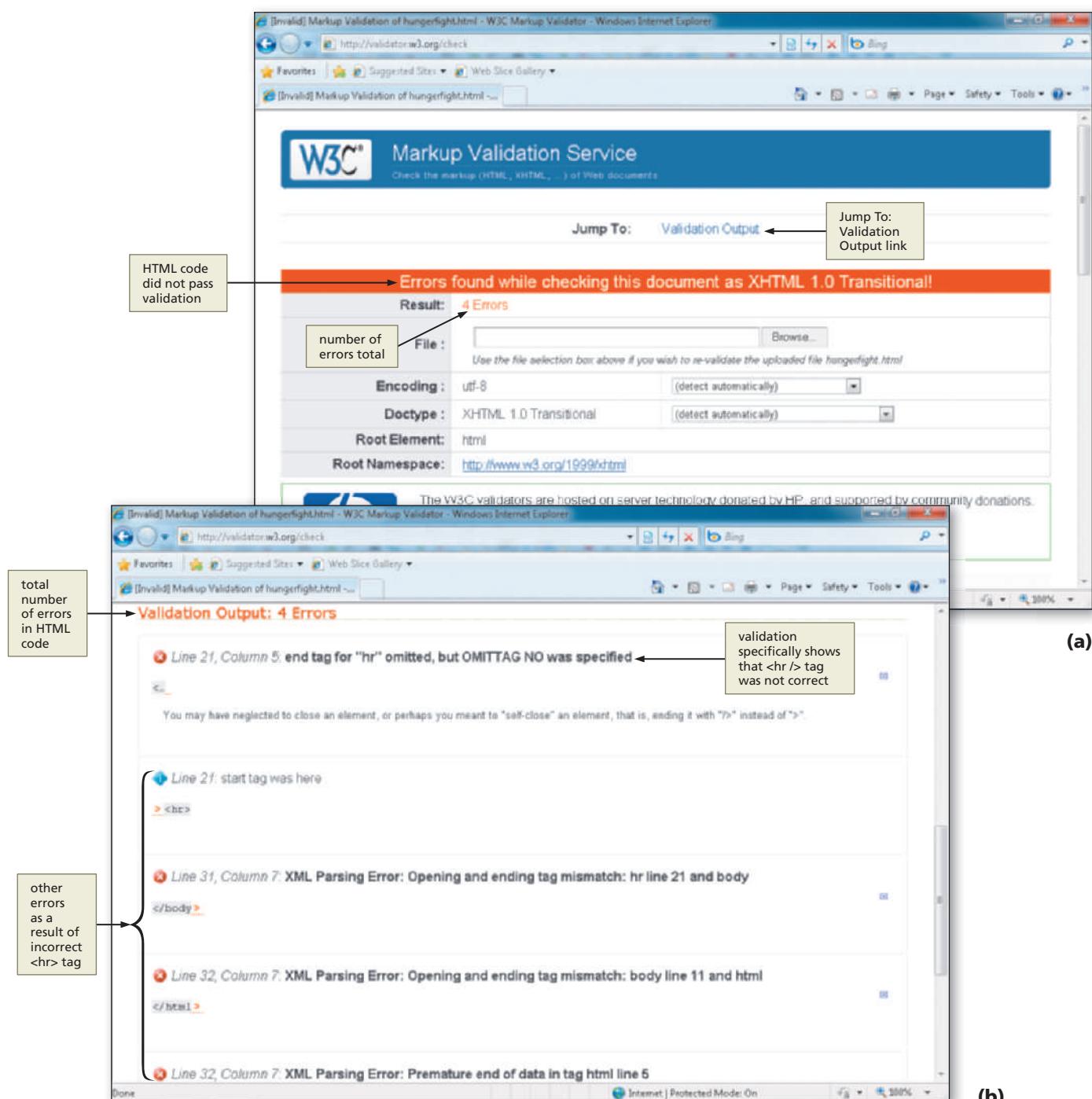


Figure 2-33

Source code is the code or instructions used to create a Web page or program. For a Web page, the source code is the HTML code, which then is translated by a browser into a graphical Web page. You can view the HTML source code for any Web page from within your browser. This feature allows you to check your own HTML source code, as well as to see the HTML code other developers used to create their Web pages. If a feature on a Web page is appropriate or appealing for your Web page, you can view the source to understand the HTML required to add that feature and then copy sections of the HTML code to put on your own Web pages.

To Validate HTML Code

Now that you have added all the basic elements to your Web page and enhanced it with images, color, italics, and rules, you need to validate your code. The following steps illustrate how to validate your HTML code using the W3 validator.

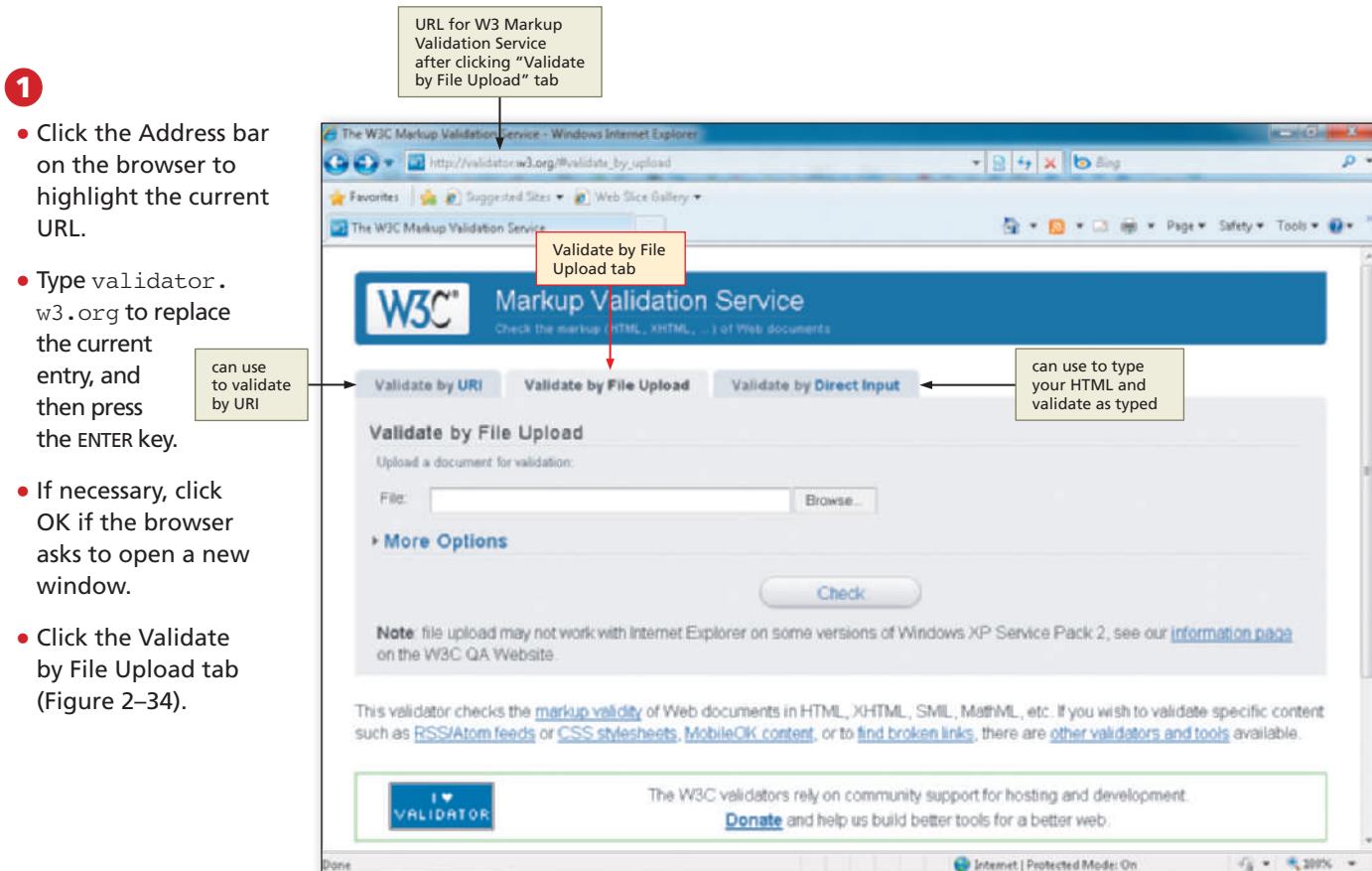


Figure 2–34

2

- Click the Browse button.
- Locate the hungerfight.html file on your storage device and then click the file name.
- Click the Open button on the Choose File to Upload dialog box and the file path and name will be inserted into the File box, as shown in Figure 2–35.

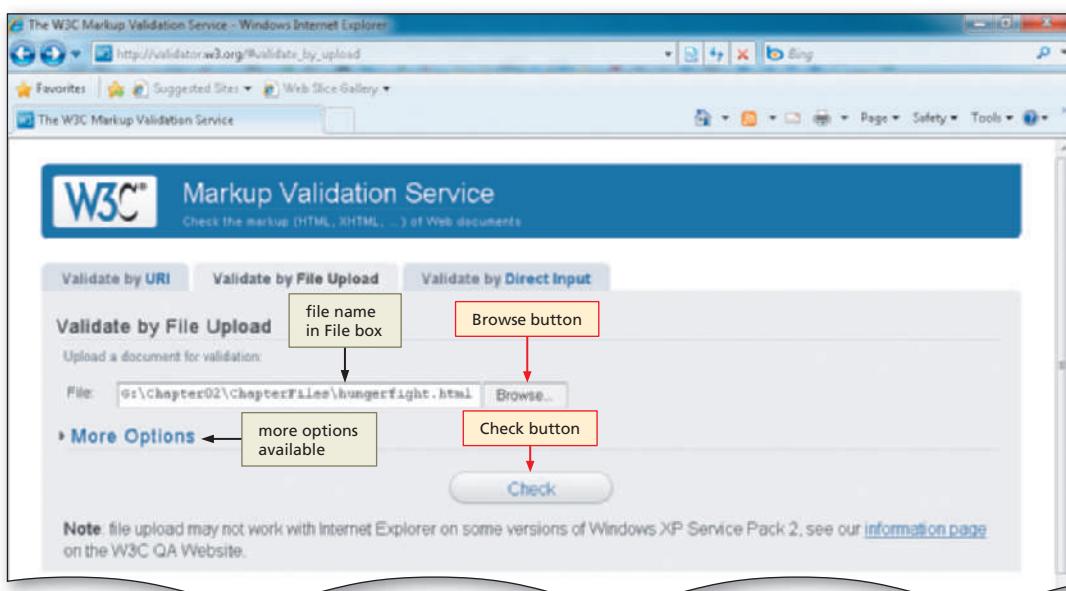


Figure 2–35

3

- Click the Check button (Figure 2–35). The resulting validation should be displayed, as shown in Figure 2–36.
- Return to the Hunger Fight Web page, either by clicking the Back button on your browser or by clicking the Internet Explorer button in the taskbar.

Q&A

How do I know if my HTML code is valid? The notification header will be green, and in the Result area, you should see the word “Passed.”

Q&A

What can I do if my HTML code is not validated?

If your code has errors, edit your HTML file to correct the errors. The Markup Validation Service report lists what is wrong with your code. Once you make the necessary changes and save the file, you can use the Browse button to open the corrected HTML file, then scroll down and click the Revalidate button to validate the changed code.

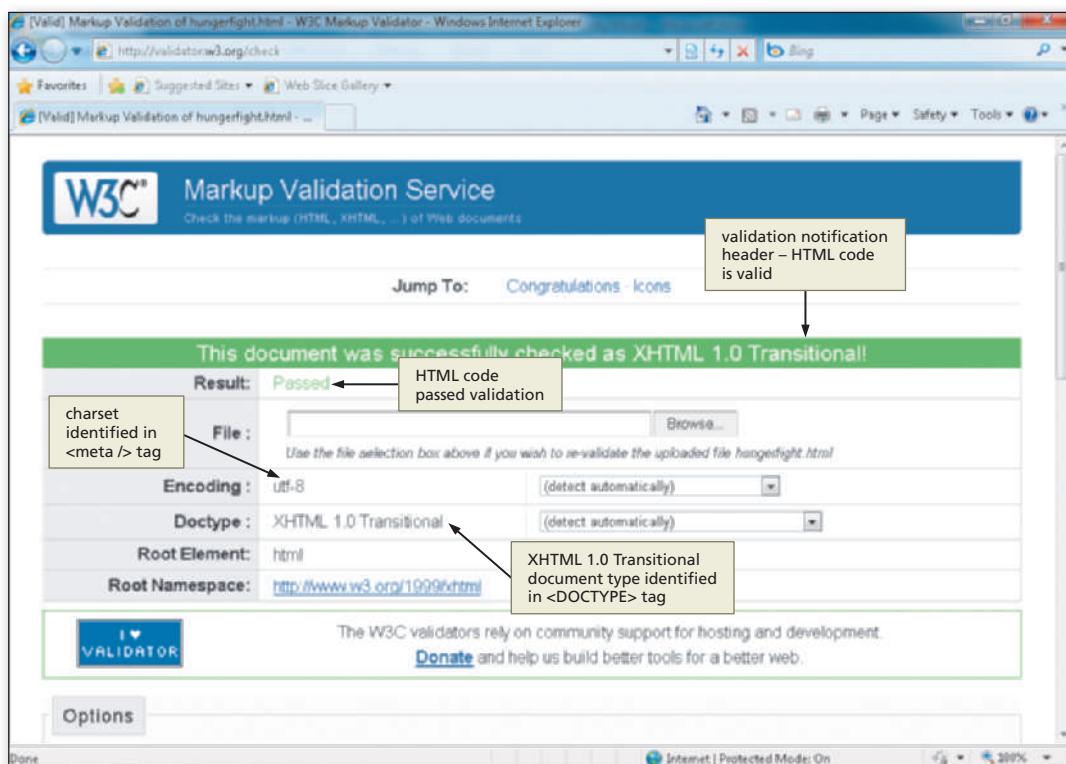


Figure 2–36

To View HTML Source Code for a Web Page

You can use your browser to look at the source code for most Web pages. The following steps show how to view the HTML source code for your Web page using a browser.

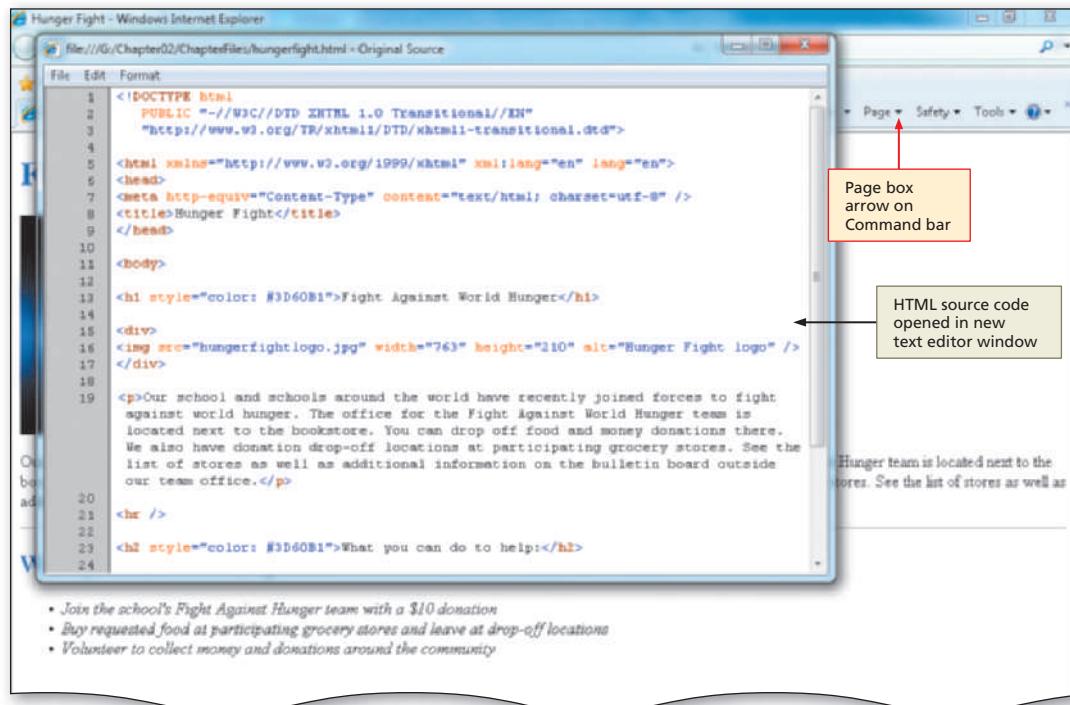
1

- Click Page on the Command bar (Figure 2-37).
- Click View Source to view the HTML code in the default text editor.

Q&A

Do all browsers allow me to view the HTML source code in the same way?

Browsers such as Firefox or Safari differ from Internet Explorer and might use different buttons or menu options to access source code. For instance, in Mozilla Firefox, select View and then Page Source.

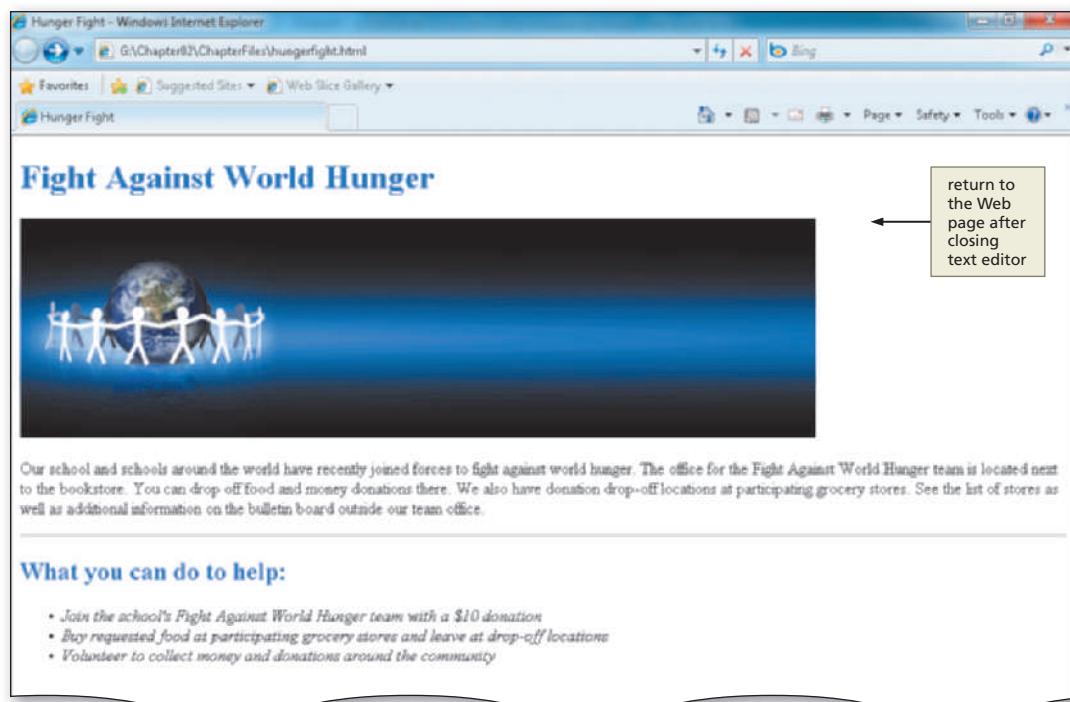
**Figure 2-37****2**

- Click the Close button on the text editor title bar to close the active text editor window (Figure 2-38).

Q&A

What is the default text editor?

It is likely to be Notepad for Internet Explorer, but could be Notepad++ or another editor depending on your browser setup.

**Figure 2-38**

To Print a Web Page and an HTML File

After you have created the HTML file and saved it, you might want to print a copy of the HTML code and the resulting Web page. A printed version of a file, Web page, or other document is called a **hard copy** or **printout**. Printed copies of HTML files and Web pages can be kept for reference or to distribute. In many cases, HTML files and Web pages are printed and kept in binders for use by others. The following steps show how to print a Web page and its corresponding HTML file.

1

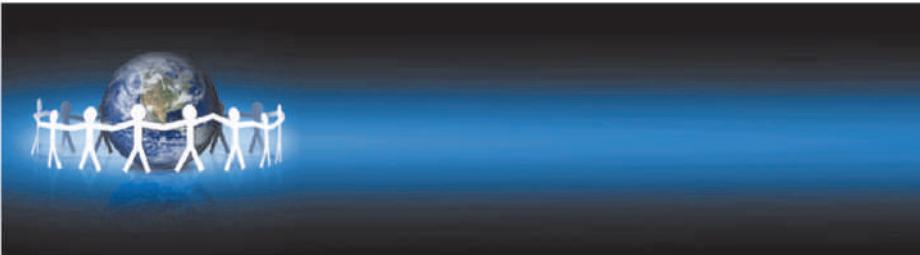
- Ready the printer according to the printer instructions.
- With the Fight Against World Hunger Web page open in the browser window, click the Print icon on the Command bar.
- When the Print dialog box appears, click the Print button.
- When the printer stops printing the Web page, retrieve the printout (Figure 2–39).

Q&A

Are there other ways to print a Web page? Pressing CTRL+P opens the Print dialog box, where you can select print options. You can also use the File menu, Print option.

Hunger Fight Page 1 of 1

Fight Against World Hunger



Our school and schools around the world have recently joined forces to fight against world hunger. The office for the Fight Against World Hunger team is located next to the bookstore. You can drop off food and money donations there. We also have donation drop-off locations at participating grocery stores. See the list of stores as well as additional information on the bulletin board outside our team office.

What you can do to help:

- Join the school's Fight Against World Hunger team with a \$10 donation
- Buy requested food at participating grocery stores and leave at drop-off locations
- Volunteer to collect money and donations around the community

Figure 2–39

2

- Click the Notepad++ button on the taskbar to activate the Notepad++ window.
- Click File on the menu bar, click the Print command, and then click the Print button to print a hard copy of the HTML code (Figure 2-40).

Q&A

Why do I need a printout of the HTML code?

Having a hard-copy printout is an invaluable tool for beginning developers. A printed copy can help you immediately see the relationship between the HTML tags and the Web page that you view in the browser.

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Hunger Fight</title>
</head>

<body>

<h1 style="color: #3D60B1">Fight Against World Hunger</h1>

<div>

</div>

<p>Our school and schools around the world have recently joined forces to fight against world hunger. The office for the Fight Against World Hunger team is located next to the bookstore. You can drop off food and money donations there. We also have donation drop-off locations at participating grocery stores. See the list of stores as well as additional information on the bulletin board outside our team office.</p>

<hr />

<h2 style="color: #3D60B1">What you can do to help:</h2>

<ul style="font-style: italic">
<li>Join the school's Fight Against World Hunger team with a $10 donation</li>
<li>Buy requested food at participating grocery stores and leave at drop-off locations</li>
<li>Volunteer to collect money and donations around the community</li>
</ul>

</body>
</html>
```

-1-

Figure 2-40

Quick Reference

For a list of HTML tags and their associated attributes, see the HTML Quick Reference (Appendix A) at the back of this book, or visit the HTML Quick Reference Web page (scsite.com/HTML6e/qr). For a list of CSS properties and values, see Appendix D.

To Quit Notepad++ and a Browser

The following step shows how to quit Notepad++ and a browser.

- 1** In Notepad++, click the File menu, then Close All.
- 2** Click the Close button on the Notepad++ title bar.
- 3** Click the Close button on the Hunger Fight Internet Explorer title bar.

Chapter Summary

In this chapter, you have learned how to identify the elements of a Web page, define the Web page structure, and enter Web page content using a text editor. You enhanced Web page appearance using inline styles, saved and validated your code, and viewed your Web page and source code in a browser. The items listed below include all the new HTML skills you have learned in this chapter.

- 1. Start Notepad++ (HTML 40)
- 2. Enable Word Wrap in Notepad++ (HTML 42)
- 3. Define the Web Page Structure Using HTML Tags (HTML 42)
- 4. Enter a Heading (HTML 47)
- 5. Enter a Paragraph of Text (HTML 47)
- 6. Create an Unordered List (HTML 50)
- 7. Save an HTML File (HTML 54)
- 8. Start a Browser (HTML 57)
- 9. View a Web Page in a Browser (HTML 59)
- 10. Activate Notepad++ (HTML 61)
- 11. Add an Image (HTML 66)
- 12. Add Color to Web Page Headings (HTML 70)
- 13. Change the Bulleted List Style (HTML 71)
- 14. Add a Horizontal Rule (HTML 72)
- 15. Refresh the View in a Browser (HTML 73)
- 16. Validate HTML Code (HTML 75)
- 17. View HTML Source Code for a Web Page (HTML 77)
- 18. Print a Web Page and an HTML File (HTML 78)
- 19. Quit Notepad++ and a Browser (HTML 80)

Learn It Online

Test your knowledge of chapter content and key terms.

Instructions: To complete the Learn It Online exercises, start your browser, click the Address bar, and then enter the Web address scsite.com/html6e/learn. When the HTML Learn It Online page is displayed, click the link for the exercise you want to complete and read the instructions.

Chapter Reinforcement TF, MC, and SA

A series of true/false, multiple choice, and short answer questions that test your knowledge of the chapter content.

Flash Cards

An interactive learning environment where you identify chapter key terms associated with displayed definitions.

Practice Test

A series of multiple choice questions that test your knowledge of chapter content and key terms.

Who Wants To Be a Computer Genius?

An interactive game that challenges your knowledge of chapter content in the style of a television quiz show.

Wheel of Terms

An interactive game that challenges your knowledge of chapter key terms in the style of the television show, *Wheel of Fortune*.

Crossword Puzzle Challenge

A crossword puzzle that challenges your knowledge of key terms presented in the chapter.

Apply Your Knowledge

Reinforce the skills and apply the concepts you learned in this chapter.

Editing the Apply Your Knowledge Web Page

Instructions: Start Notepad++. Open the file apply2-1.html from the Chapter02\Apply folder of the Data Files for Students. See the inside back cover of this book for instructions for downloading the Data Files for Students, or contact your instructor for information about accessing the required files for this book.

The apply2-1.html file is a partially completed HTML file that you will use for this exercise. Figure 2–41 shows the Apply Your Knowledge Web page as it should be displayed in a browser after the additional HTML tags and attributes are added.

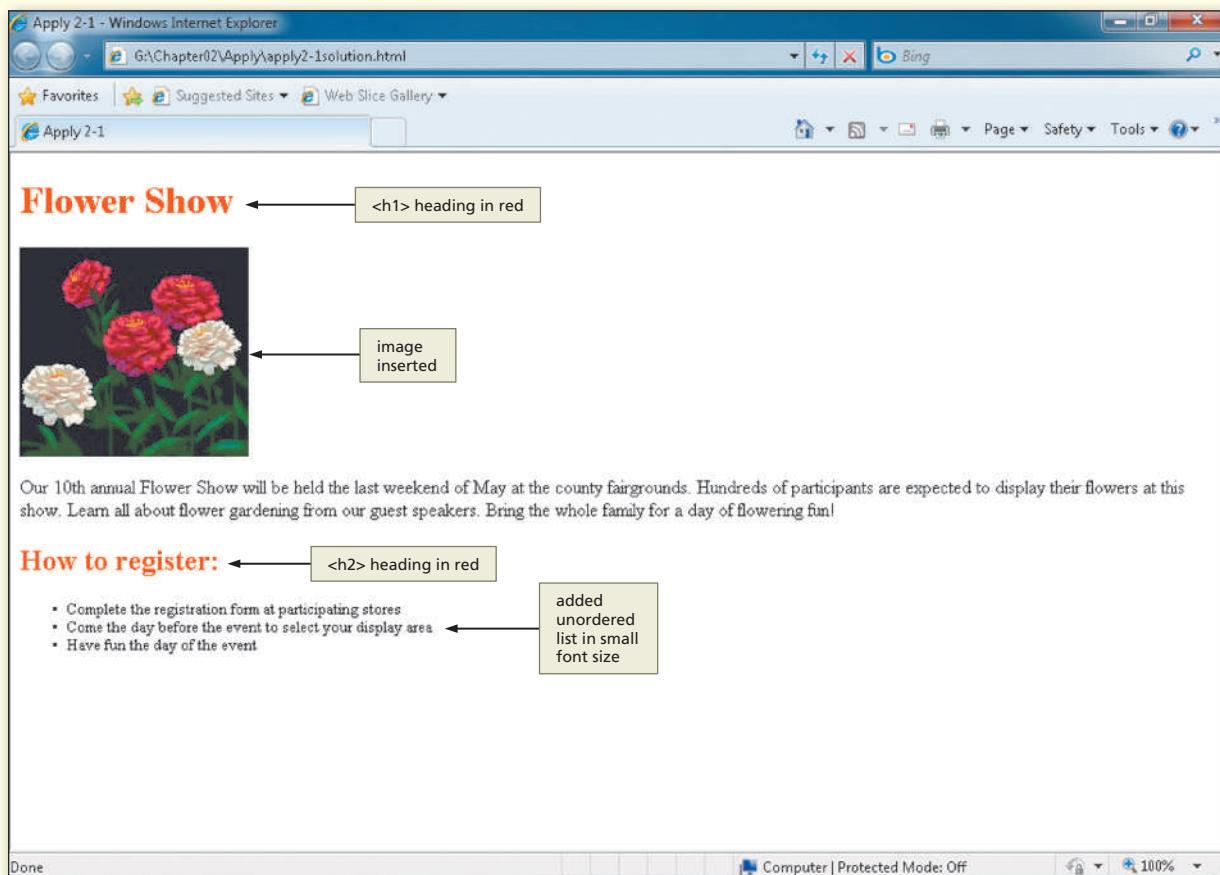


Figure 2–41

Perform the following tasks:

1. Enter g:\Chapter02\Apply\apply2-1.html as the URL to view the Web page in your browser.
2. Examine the HTML file and its appearance in the browser.
3. Using Notepad++, change the HTML code to make the Web page look similar to the one shown in Figure 2–41. Both headings are the color red. (*Hint:* Use the style="color: red" property.)
4. Add the image flowershow.jpg (in the Chapter02\Apply folder) to the Web page. It has a width of 192 pixels and a height of 175 pixels. (*Hint:* Include the image in a <div> </div> container and remember to use the alt attribute.)
5. Make the bulleted list in a small font size. (*Hint:* Use the style="font-size: small" property.)

Continued >

Apply Your Knowledge *continued*

6. Save the revised HTML file in the Chapter02\Apply folder using the file name apply2-1solution.html.
7. Validate your HTML code at <http://validator.w3.org/>.
8. Enter g:\Chapter02\Apply\apply2-1solution.html as the URL to view the revised Web page in your browser.
9. Print the Web page.
10. Submit the revised HTML file and Web page in the format specified by your instructor.

Extend Your Knowledge

Extend the skills you learned in this chapter and experiment with new skills.

Creating a Definition List

Instructions: Start Notepad++. Open the file, extend2-1.html from the Chapter02\Extend folder of the Data Files for Students. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files. This sample Web page contains all of the text for the Web page in bulleted list format. You will add the necessary tags to make this a definition list with terms that are bold, as shown in Figure 2–42.

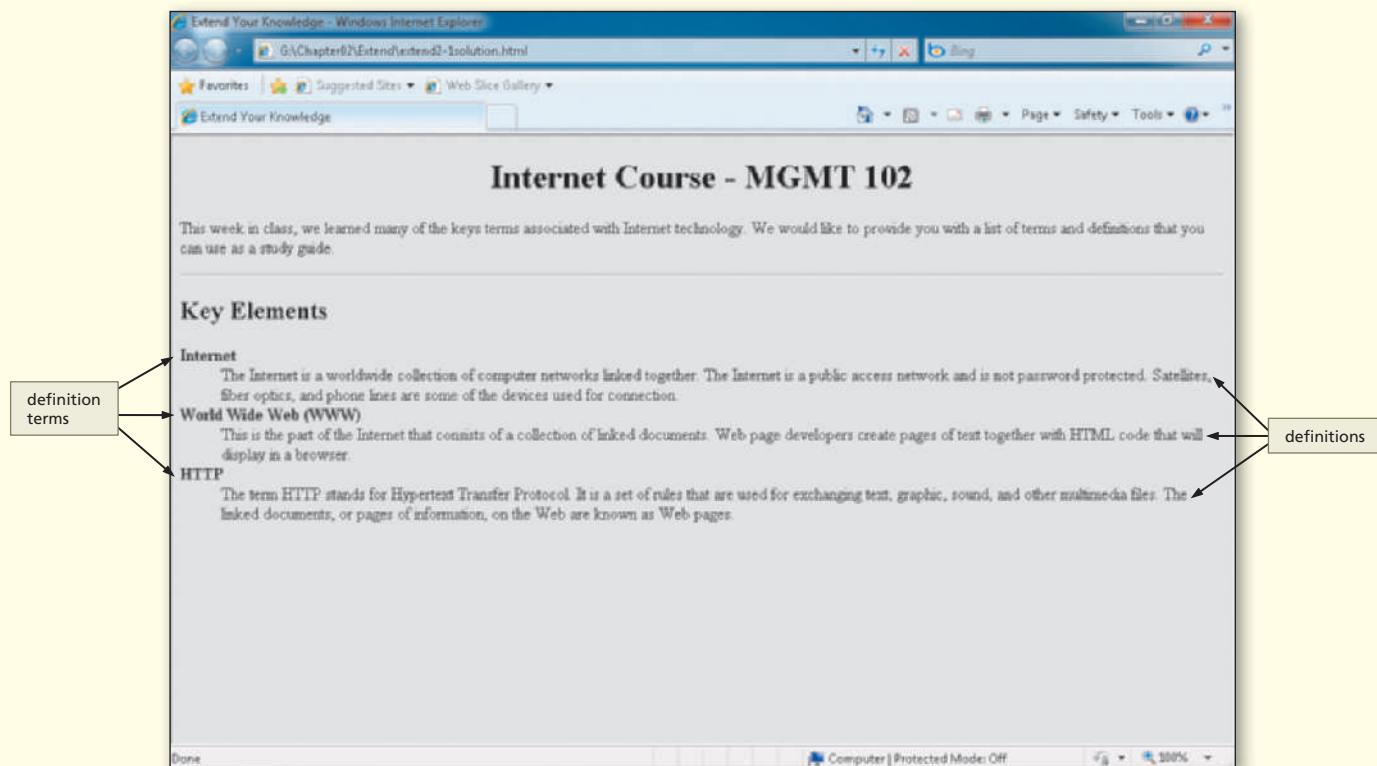


Figure 2–42

Perform the following tasks:

1. Using the text given in the file extend2-1.html, make changes to the HTML code to change the Web page from a bulleted list to a definition list by following the definition list code shown in Figure 2–14 on page HTML 52.

2. Add the additional HTML code necessary to make the terms bold (see font-weight for style in Appendix D) and a background color of #e0e0e0 (see background-color in Appendix D).
3. Save the revised document in the Chapter02\Extend folder with the file name extend2-1solution.html, validate the Web page, and then submit it in the format specified by your instructor.

Make It Right

Analyze a document and correct all errors and/or improve the design.

Correcting the Friendly Reminder Web Page

Instructions: Start Notepad++. Open the file makeitright2-1.html from the Chapter02\MakeItRight folder of the Data Files for Students. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files. The Web page is a modified version of what you see in Figure 2–43. Make the necessary corrections to the Web page to make it look like Figure 2–43. Format the heading to use the Heading 1 style center-aligned on the Web page (see the text-align property in Appendix D to center a heading). Add two size-10px horizontal rules, as shown in Figure 2–43. Save the file in the Chapter02\MakeItRight folder as makeitright2-1solution.html, validate the Web page, and then submit it in the format specified by your instructor.

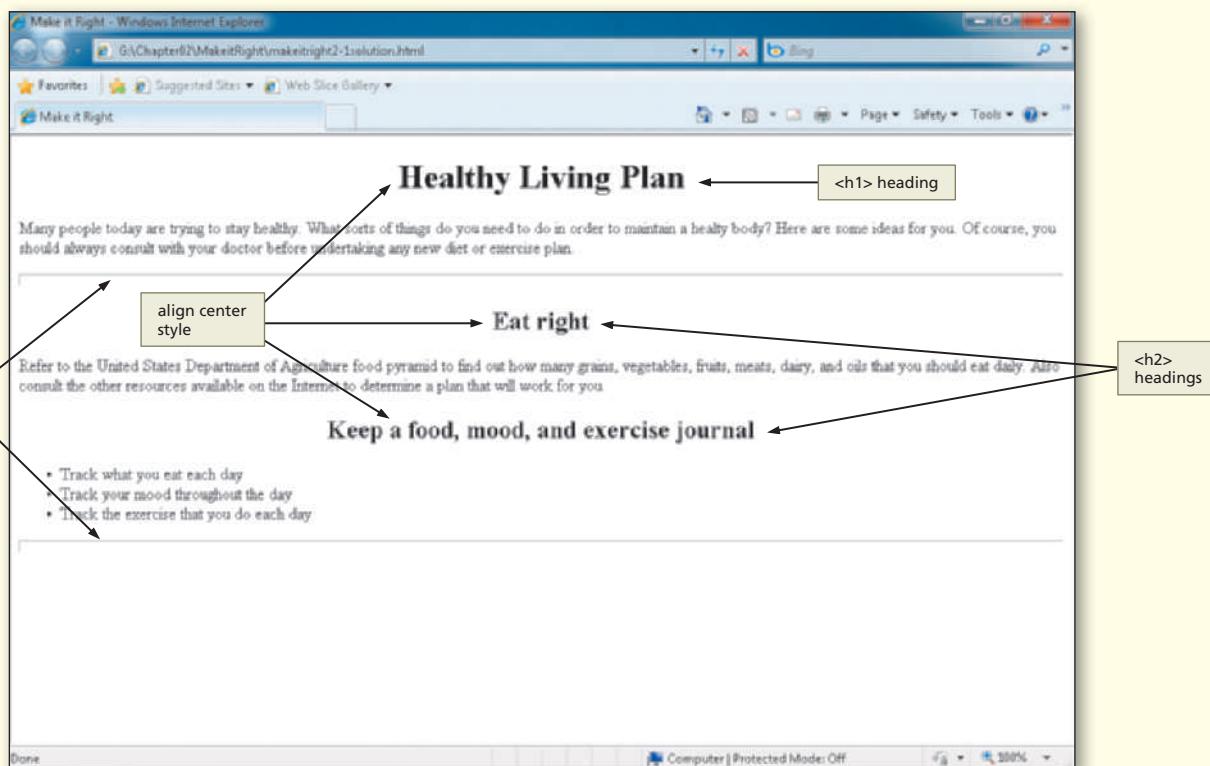


Figure 2–43

In the Lab

Lab 1: Creating a Food Drive Web Page

Problem: You did volunteer work for the Community Food Drive in your city. You would like to recruit other friends to volunteer for community service. You have been asked to create a Web page to display information about why you choose to volunteer and let people know how they also can help, as shown in Figure 2–44.

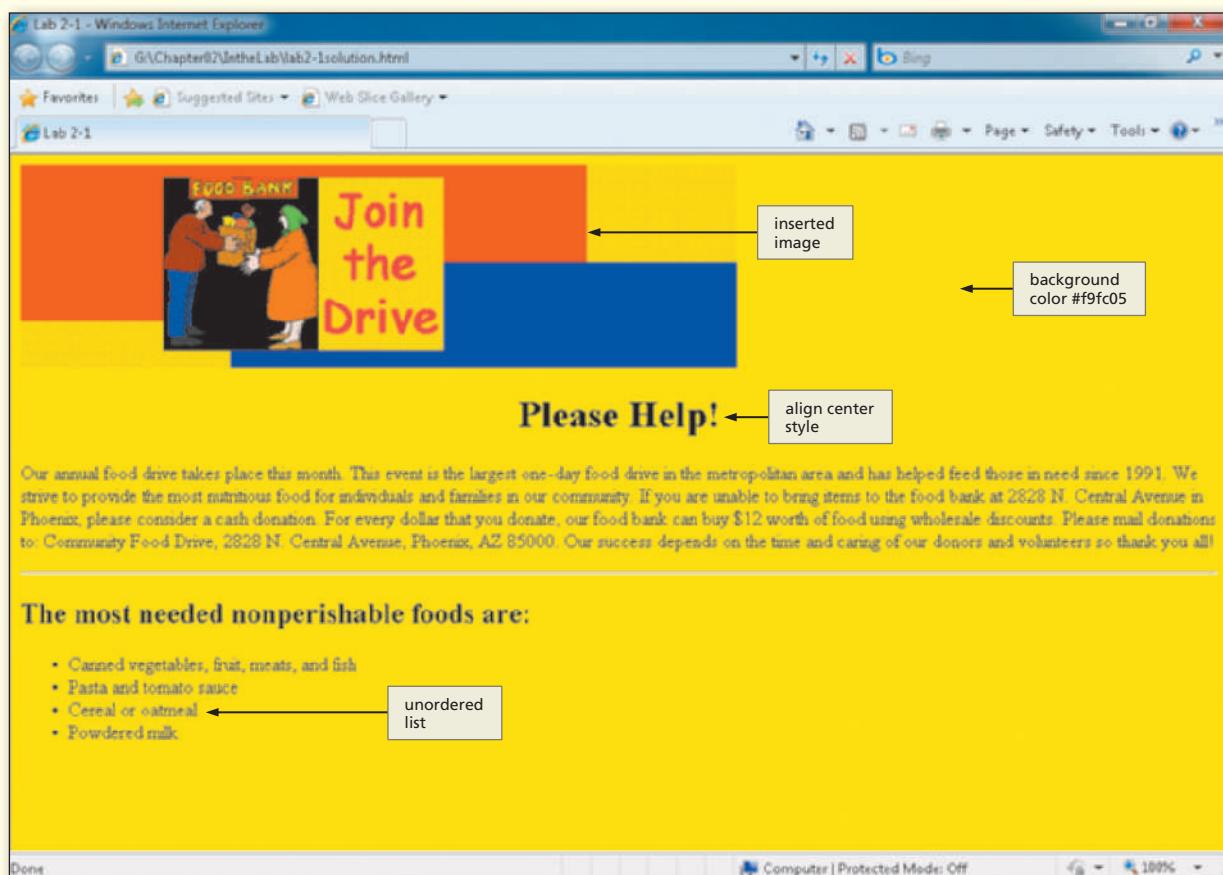


Figure 2–44

Instructions: Perform the following steps:

1. Start Notepad++ and create a new HTML file with the title, Lab 2-1, within the `<title> </title>` tags. For the initial HTML tags, you can use the structure.html file if you created one at the start of this chapter's project, otherwise type the initial tags.
2. Begin the body section by adding the foooddriveLogo.gif image as well as the heading, Please Help! Format the heading to use the Heading 1 style center-aligned on the Web page. (*Hint:* See the `text-align` property in Appendix D to center the heading.)
3. Add a left-aligned heading, as shown, using the Heading 2 style.
4. Add a background color to the Web page using the `#f9fc05` color code. (*Hint:* Use the `style="background-color: color number"` property.)
5. Add an unordered list of nonperishable foods, as shown in Figure 2–44.
6. Save the file in the Chapter02\IntheLab folder as lab2-1solution.html.

7. Print the lab2-1.html file.
8. Enter g:\Chapter02\IntheLab\lab2-1solution.html as the URL to view the Web page in your browser.
9. Print the Web page.
10. Submit the revised HTML file and Web page in the format specified by your instructor.

In the Lab

Lab 2: Creating an Informational Web Page

Problem: You continue to enjoy volunteering and decide to prepare a Web page announcement, such as the one shown in Figure 2–45, to promote the latest food drive.

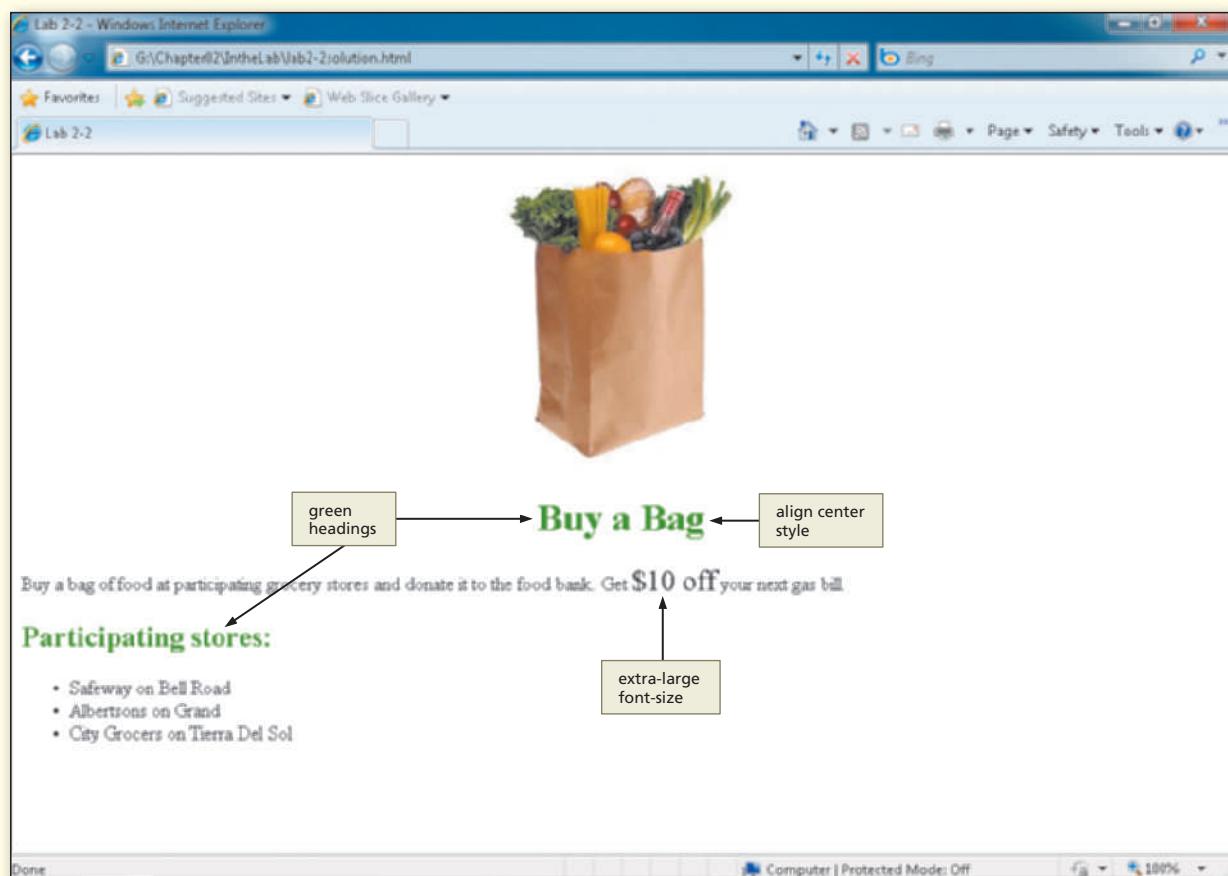


Figure 2–45

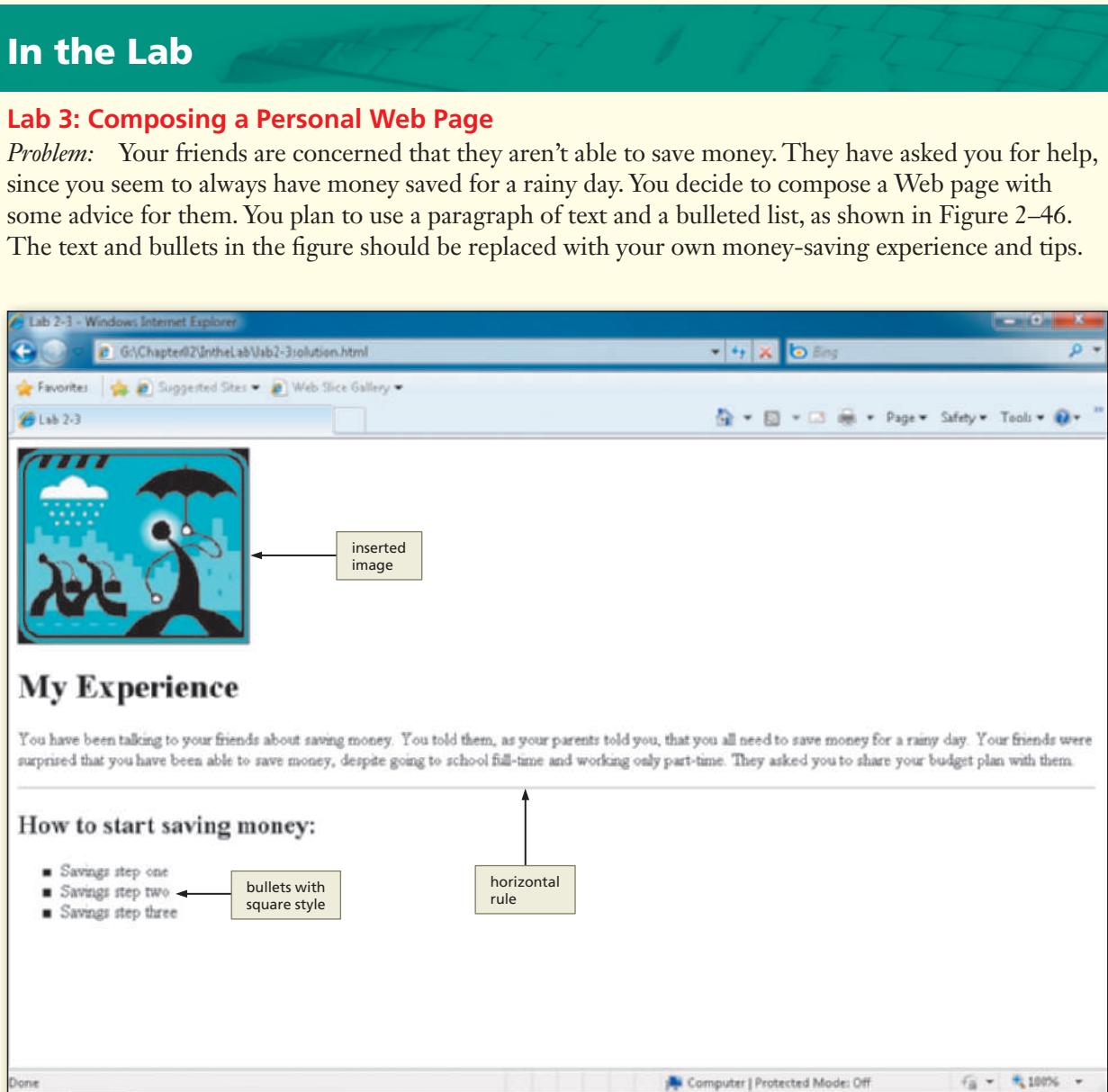
Instructions: Perform the following steps:

1. Start Notepad++ and create a new HTML file with the title Lab 2-2 within the <title> </title> tags.
2. The Web page uses the foodbag.jpg image file, which has a width of 207 and a height of 256. Center the image (see the text-align property in Appendix D). Use the color green for both headings.
3. Add the paragraph of text, as shown in Figure 2–45. Make the words "\$10 off" x-large style of font (see the HTML tag in Appendix A to help with that).
4. Create one bulleted list with the information shown.

Continued >

In the Lab *continued*

5. Save the file in the Chapter02\IntheLab folder using the file name lab2-2solution.html.
6. Print the lab2-2solution.html file.
7. Enter g:\Chapter02\IntheLab\lab2-2solution.html as the URL to view the Web page in your browser.
8. Print the Web page.
9. Submit the revised HTML file and Web page in the format specified by your instructor.

**Figure 2–46**

Instructions: Perform the following steps:

1. Start Notepad++ and create a new HTML file with the title Lab 2-3 within the <title> </title> tags.
2. Include a short paragraph of information and a bulleted list, using a format similar to the one shown in Figure 2–46, to provide information about your money-saving experience.

3. Insert the image file cloudy.jpg, stored in the Chapter02\IntheLab folder. Right-click the image, click Properties, and then click the Details tab to find out the image's dimensions, or open it in a graphics program. Note that the bullets used for the list are square in shape.
4. Save the HTML file in the Chapter02\IntheLab folder using the file name lab2-3solution.html.
5. Enter g:\Chapter02\IntheLab\lab2-3solution.html as the URL to view the Web page in your browser.
6. Print the Web page from your browser.
7. Submit the revised HTML file and Web page in the format specified by your instructor.

Cases and Places

Apply your creative thinking and problem-solving skills to design and implement a solution.

• EASIER •• MORE DIFFICULT

• 1: Add to the Food Drive Web Page

Mr. Wattigney, the director of the Community Food Bank, likes the Web page you created for Lab 2-1. Now that the Food Drive is over, he would like you to update the Web page with new information on upcoming community events. Before updating the page, search the Web to review the Web pages at other food banks or departments of community services for ideas on content to include or formatting to change. What do their Web sites look like? Are there changes you can make to the Lab 2-1 Web page that reflect what other places have done? Using the concepts presented in this chapter, include additional information or change the formatting to make the page more interesting and timely.

• 2: Create an Artist Web Site

You are creating a new Web site for a local photographer. The photographer has asked that you use descriptive alt attributes for images on the Web page, because many of the viewers of his Web page have very slow Internet connections and images often do not load quickly. Search the Web for information on adding useful, descriptive alt attributes for images. Also find information on using thumbnail images. Give suggestions for loading images faster. Create a document with a brief paragraph explaining the information that you found in your research. Make suggestions about how the photographer's Web site can be made more effective.

•• 3: Create a Web Page of CSS Properties

You have learned a lot about using Cascading Style Sheets (CSS) so far in the class. You still have some questions, though, about how to insert certain styles. Research information on the Web (don't forget to look at the W3.org site) to find Web sites that list CSS properties and values that complement Appendix D. In a Word document, add the URLs that you found to be very helpful. Under each URL write a brief paragraph that describes why you thought the site could be useful for new Web developers. Share your list of URLs with fellow students.

•• 4: Create a Personal Web Page

Make It Personal

Your class instructor wants to post all of the students' Web pages on the school server to show what his or her students are interested in. Create a Web page of personal information, listing items such as your school major, jobs that you have had in the past, and your hobbies and interests. To make your personal

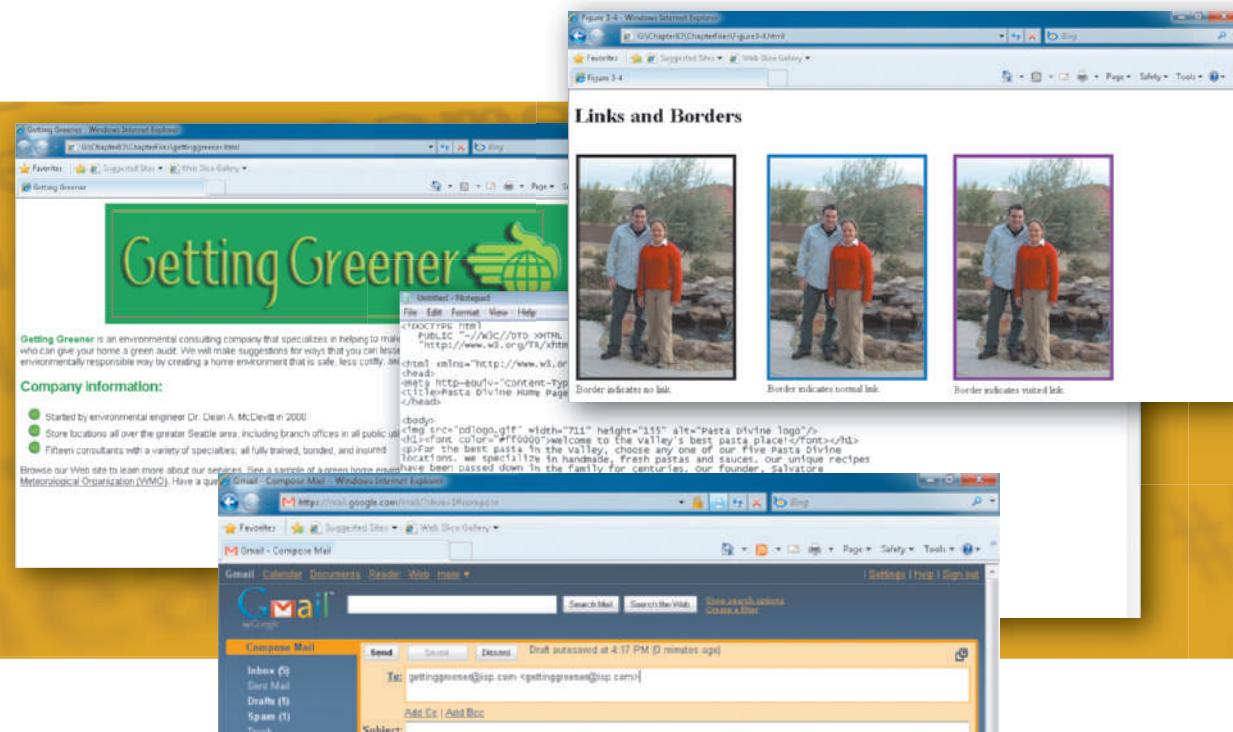
Cases and Places *continued*

Web page more visually interesting, search the Web for images that reflect your interests. (Remember that if the image is copyrighted, you cannot use it on a personal Web page unless you follow the guidelines provided with the image.) Insert an image or two onto the Web page to help explain who you are.

•• 5: Create Web Pages with Different CSS Properties**Working Together**

Work with other students to review the CSS in Appendix D together with the Web sites dedicated to CSS found in Cases and Places 3. Try to find a use for some CSS properties that you have not used so far in this chapter (maybe something with margins). Plan how you can use five CSS properties not used in Chapter 2 to improve one of the Web pages you created in this chapter (either the chapter project or an end-of-chapter project). Utilize the five different CSS properties in the Web page. Explain in a Word document what CSS properties you used, why you used them, and how they improved the Web page.

3 Creating Web Pages with Links, Images, and Embedded Style Sheets



Objectives

You will have mastered the material in this chapter when you can:

- Describe linking terms and definitions
- Create a home page and enhance a Web page using images
- Change body and heading format using embedded style sheets
- Align and add color to text using embedded and inline styles
- Use an inline style to insert an image for bullets in an unordered list
- Add a text link to a Web page in the same Web site
- Add an e-mail link
- Add a text link to a Web page on another Web site
- Use absolute and relative paths
- Save, validate, and view an HTML file and test the links
- Use style classes to add an image with wrapped text
- Add links to targets within a Web page
- Copy and paste HTML code
- Add an image link to a Web page in the same Web site

3 | Creating Web Pages with Links, Images, and Embedded Style Sheets

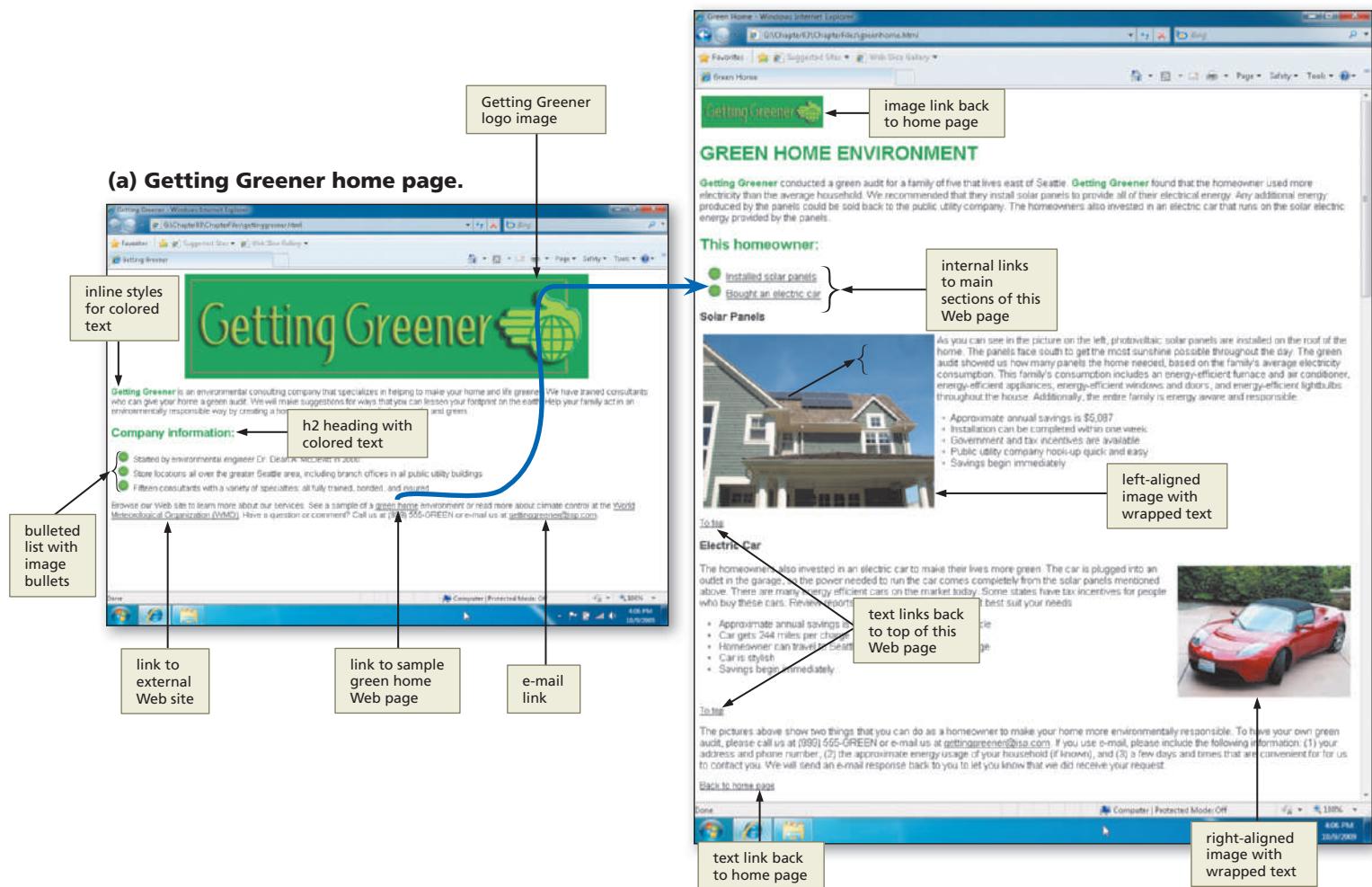
Introduction

One of the most useful and important aspects of the World Wide Web is the ability to connect (link) one Web page to other Web pages — on the same server or on different Web servers — located anywhere in the world. Using hyperlinks, a Web site visitor can move from one page to another, and view information in any order. Many different Web page elements, including text, graphics, and animations, can serve as hyperlinks. In this chapter, you will create Web pages that are linked together using both text links and image links. In the last chapter, you used inline styles to change the appearance of individual elements or HTML tags. In this chapter, you will also use embedded style sheets to set the appearance of elements such as headings and body text for the entire Web page. Before starting on this project, you would have already completed the Web site planning, analysis, and design phases of the Web Development Life Cycle.

Project — Getting Greener Web Site

Chapter 3 illustrates how to use HTML to create a home page for the Getting Greener Web site (Figure 3–1a) and to edit the existing greenhome.html Web page (Figure 3–1b) to improve its appearance and function. Your older brother, Dean, recently opened an environmental consulting company and named it Getting Greener. He would like to advertise his company on the Web and show a sample home on which he has done a green audit. He knows that you have studied Web development in college and asks you to develop two Web pages that are linked together: a home page, and a Web page with the sample home. During your analysis, you determined that there are four basic types of links to use. The first type is a link from one Web page to another in the same Web site. The second type is a link to a Web page on a different Web site. The third type is an e-mail link. The fourth type is a link within one Web page. You plan to utilize all four of these types of links for your brother’s Web site.

The Getting Greener home page (Figure 3–1a) includes a logo image banner, headings, an unordered (bulleted) list with a graphic image as the bullet, an e-mail link, and a text link to a Web page on another Web site. This page also includes a link to the greenhome.html Web page. The Green Home Web page (Figure 3–1b) contains two images with text wrapped around them and internal links that allow visitors to move easily from section to section within the Web page. The Web page also has an image link back to Getting Greener’s home page.

(b) Sample Green Home Web page.**Figure 3–1**

Overview

As you read this chapter, you will learn how to create the Web page shown in Figure 3–1 by performing these general tasks:

- Enter HTML code into the Notepad++ window.
- Save the file as an HTML file.
- Enter basic HTML tags and add text to the file.
- Use embedded style sheets, inline styles, and classes to change the format of text and headings.
- Use an inline style to create a bulleted list with a graphic image as the bullets.
- Add a link to another Web page in the same Web site.
- Add a link to an external Web site.
- Add an e-mail link.
- Add targets and links within the same Web page.
- View the Web pages and HTML code in your browser.
- Validate the Web pages.
- Print the Web pages.

**Plan
Ahead****Project Planning Guidelines**

As you create Web pages, such as the project shown in Figure 3–1 on the previous page, you should follow these general guidelines:

1. **Plan the Web site.** Before developing a multiple-page Web site, you must plan the purpose of the site. Refer to Table 1–4 on page HTML 15 for information on the planning phase of the Web Development Life Cycle. In this phase, you determine the purpose of the Web site, identify the users of the site and their computing environments, and decide who owns the information on the Web page.
2. **Analyze the need.** In the analysis phase of the Web Development Life Cycle, you analyze what content to include in the Web page. The Web development project in Chapter 3 is different than the one completed in Chapter 2 because it contains two Web pages that will be linked together. Part of the analysis phase then includes determining how the multiple Web pages work together to form a Web site.
3. **Design the Web site.** Once the analysis is complete, you design the Web site. In this phase, you determine the content of the site, both text and graphics. Design steps specific to this chapter also include determining links within the site and to external Web sites.
 - a. **Choose the content for the Web pages.** This part of the life cycle also differs from the previous chapter's project because all of the content does not have to appear on one Web page, as it did in Chapter 2. With a multiple-page Web site, you can distribute the content as needed throughout the Web site. Because of the nature of this Web site, pictures are a large part of the content. The Web site owner wants to show a sample of his company's work. Pictures help show what one family has done to become more environmentally responsible.
 - b. **Determine the types of Cascading Style Sheets (CSS) that you will use.** You already learned how to use inline styles to best support the design and purpose of the Web site. In this chapter, you utilize both inline and embedded style sheets to alter the appearance (or style) of various Web page elements. You also incorporate classes with your embedded style sheets to further control the style of elements on the Web page. You need to consider which of these options is best suited for the styles of your Web site.
 - c. **Determine how the pages will link to one another.** This Web site consists of a **home page** (the first page in a Web site) and a secondary Web page to which you will link. You need to determine how to link (e.g., with text or a graphic) from the home page to the secondary page and how to link back to the home page.
 - d. **Establish what other links are necessary.** In addition to links between the home page and secondary Web page, you need an e-mail link. It is standard for Web developers to provide an e-mail link on the home page of a Web site for visitor comments or questions. Additionally, the secondary Web page (*greenhome.html*) is a long page that requires visitors to scroll down for navigation. Because of its length, it is important to provide easy and quick ways to navigate the Web page. You do this using links within the Web page.
4. **Develop the Web page(s) and insert all links.** Once the analysis and design is complete, the Web developer creates the Web page(s) using HTML and CSS. Good Web development standard practices should be followed in this step. Examples of good practices include utilizing the proper initial HTML tags, as shown in the previous chapter, and always identifying alt text with images.
5. **Test all Web pages within the Web site.** An important part of Web development is testing to assure that you are following XHTML standards. For the projects in this book, you will use the World Wide Web Consortium (W3C) validator that allows you to test your Web pages and clearly explains any errors it finds. When testing, you should check all content for accuracy. Also, all links (external, internal, and page to page within the same Web site) should be tested.

When necessary, more specific details concerning the above guidelines are presented at appropriate points in the chapter. The chapter also will identify the actions performed and decisions made regarding these guidelines during the creation of the Web page shown in Figure 3–1.

Using Links on a Web Page

As you have learned, many different Web page elements, including text, images, and animations, can serve as links. Text and images are the elements most widely used as links. Figure 3–2 shows examples of text and image links.

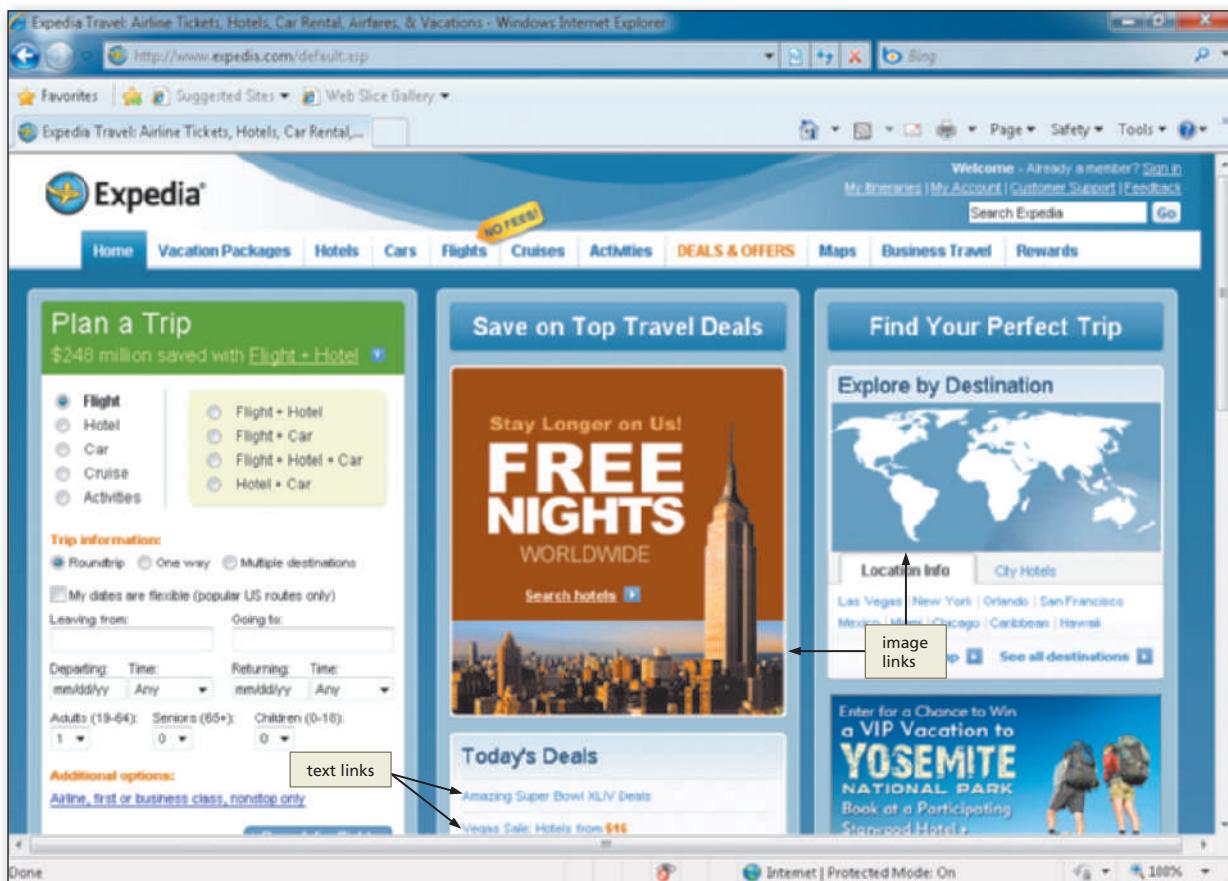


Figure 3–2 Text and image links on a Web page.

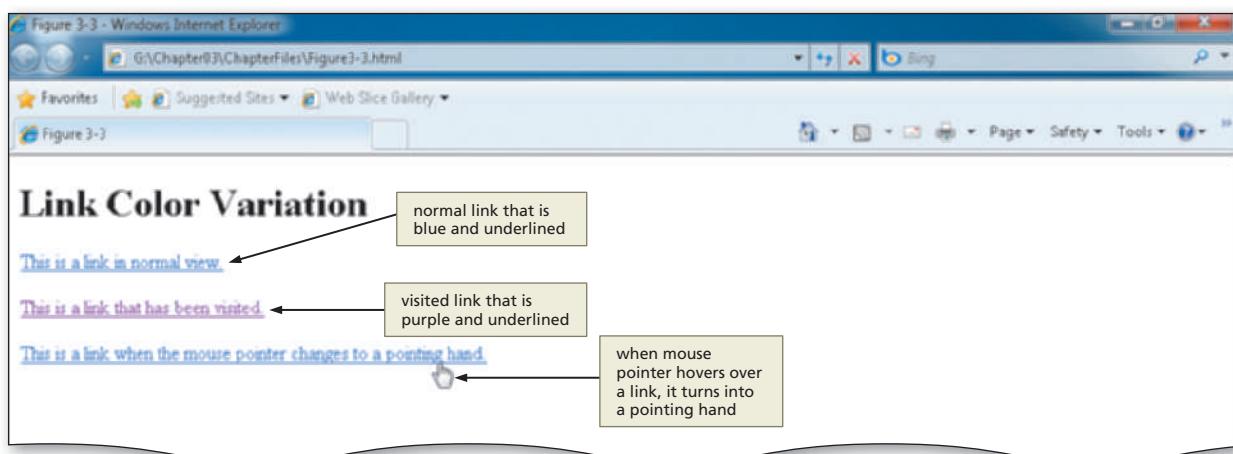
When using text links on a Web page, use descriptive text as the clickable word or phrase. For example, the phrase “Click here” does not explain the purpose of the link to the visitor. By contrast, the phrase “Save up to 40% on flights” indicates that the link connects to a Web page with discounted airline tickets.

When text identifies a link, it often appears as underlined text, in a color different from the main Web page text. Unless otherwise changed in the anchor `<a>` or `<body>` tags, the browser settings define the colors of text links throughout a Web page. For example, with Internet Explorer, the default color for a normal link that has not been clicked (or visited) is blue, a visited link is purple, and an active link (a link just clicked by a user) varies in color. Figure 3–3 on the next page shows examples of text links in all three states (normal, visited, and active). Generally, as shown in Figure 3–3, moving the mouse pointer over a link causes the mouse pointer to change to a pointing hand. This change notifies the user that a link is available from that text or image.

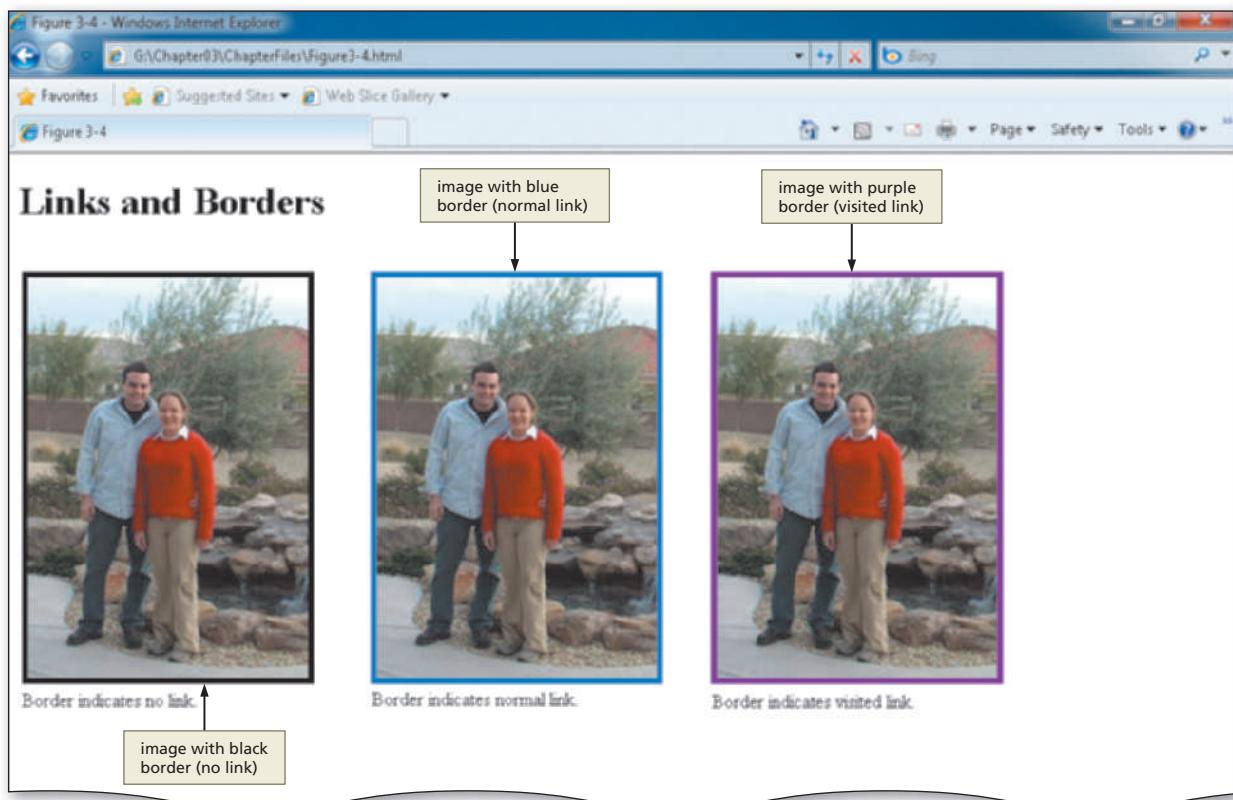


Link Help

Many Web sites provide help for new HTML developers. For more information about links, search for key words such as “HTML Tutorials” or “HTML Help” in any good search engine.

**Figure 3-3 Examples of text link color variations.**

The same color defaults apply to the border color around an image link. A border makes the image appear as if it has a frame around it. If the image has no border, no frame will appear around the image. The color of the border shows whether the border is a link, and whether the link has been visited (Figure 3-4).

**Figure 3-4**

If you want to change the color of text links or image link borders to override the browser defaults, you can designate those changes in the anchor <a> or <body> elements using an embedded or external style sheet, or by using an inline style. Recall that you use an inline style to change the appearance (or style) of a single element. An embedded style sheet is used to change the styles of similar elements in one Web page. Finally, an external style sheet is contained in a separate .css document and is used to change the style in an entire Web site. (You will use an external style sheet in the next chapter.) To use an embedded or external style sheet in the anchor element to change normal, visited, and active link colors from the default, you would use the following format:

```
a {color: black}
```

where color is a designated color name, such as black, or a hexadecimal color code. To make the same change with an inline style, the tag format is:

```
<a style="color: black">
```

You can disable the underlining of a link with the text-decoration property within the anchor tag. The **text-decoration property** allows text to be “decorated” with one of five values: underline, overline, line-through, blink, or none. This property can be used in a variety of tags including the anchor tag. Links, by default, are underlined to indicate that they are links.

In the design phase you should consider carefully the benefits and detriments of any style change, especially to a default style. Be sure that users are still able to immediately see that specific text is used as a link before turning off link underlines or changing the default link color. If you determine that you can effectively turn the underline off on a link (as you do for the image link that you create later in this chapter), you can change the text-decoration attribute to none. To do this with an embedded or external style, you would enter the following code:

```
a {text-decoration: none}
```

To change text-decoration to none with an inline style, enter:

```
<a style="text-decoration: none">
```

Linking to Another Web Page Within the Same Web Site

Web pages often include links to connect one Web page to another page within the same Web site. For example, a visitor can click a link on the home page of a Web site (Figure 3–5a on the next page) to connect and view another Web page on the same Web site (Figure 3–5b). The Web pages created in this project include links to other pages in the same Web site: (1) the Getting Greener home page includes a text link to the Green Home Web page; and (2) the Green Home Web page includes an image link back to the Getting Greener home page. To link the words “green home” on the gettinggreener.html home page to the greenhome.html Web page, you need the following HTML code:

```
<a href="greenhome.html">green home</a>
```

BTW

Link Colors

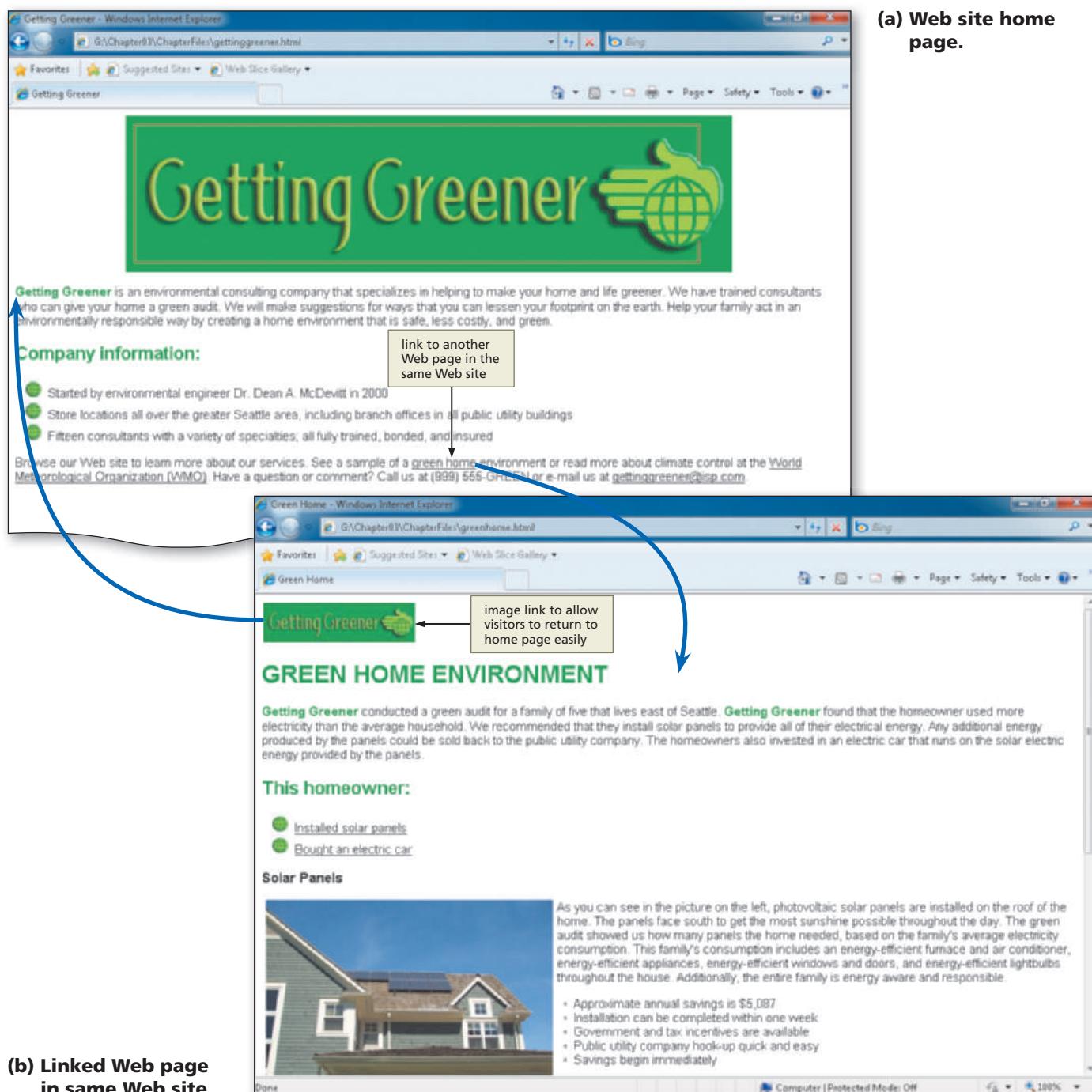
You can change the link colors in popular browsers. In Internet Explorer, you find color selection on the Tools menu, Internet Options. In Mozilla Firefox, click the Tools menu, Options, Content tab. In both browsers, you change colors by clicking a color grid.

BTW

Links on a Web Page

An anchor tag also allows visitors to move within a single Web page. Use the name attribute to allow movement from one area to another on the same page. This linking technique is useful, particularly on long pages. An index of links also can provide easy access to various areas within the Web page.

The href in the anchor <a> tag indicates that when the words “green home” are clicked, the visitor links to the greenhome.html Web page. You end the link with the tag. If you did not add the tag, then all text after the words “green home” would be linked.

**Figure 3-5**

Linking to a Web Page in Another Web Site

One of the most powerful features of Web development is the ability to link to Web pages outside of your Web site. Web developers use these links to connect their Web pages to other Web pages with information on the same topic. The links are what give the Web its value as an interconnected resource and provide its “webbiness.” In this project,

the home page (Figure 3–6a) includes a link to a page on another Web site where the visitor can find additional information about climate control (Figure 3–6b). To link the words “World Meteorological Organization (WMO)” on the gettinggreener.html home page to an external Web site, you need the following HTML code:

```
<a href="http://www.wmo.int/">World Meteorological Organization  
(WMO)</a>
```

Notice that the code is basically the same as that used to link to a Web page within the same Web site. However, you have to add the complete URL (<http://www.wmo.int/>) when you link to an external Web site.



Figure 3-6

Linking Within a Web Page

Links within a Web page allow visitors to move quickly from one section of the Web page to another. This is especially important in Web pages that are long and require a visitor to scroll down to see all of the content. Many Web pages contain a list of links like a menu or table of contents at the top of the page, with links to sections within the Web page (Figure 3–7). In this project, the Green Home Web page includes links from the top section of the Web page to other sections within the page, as well as links back to the top of the Web page. There are two steps to link within a Web page. First, you have to set a target using a name that makes sense to the purpose of the link. Then, you create a link to that target using the name given. The following HTML code shows an example of a target named solar and then the use of that target as a link. The first statement is inserted at the top of the section of the Web page to which you want to link. The second statement is inserted into the bulleted list at the top of the Web page.

```
<a name="solar"></a>
<a href="#solar">Installed solar panels</a>
```

Again, notice that you also use the anchor `<a>` tag for this type of link. However, with this inner-page link, you insert the `#` before the target name to indicate that you want to link to a specific section of the Web page, not necessarily the top of the page. Sometimes when you are browsing the Web, you might see a `#` used in a link address. That generally links you to a specific section of the Web page.

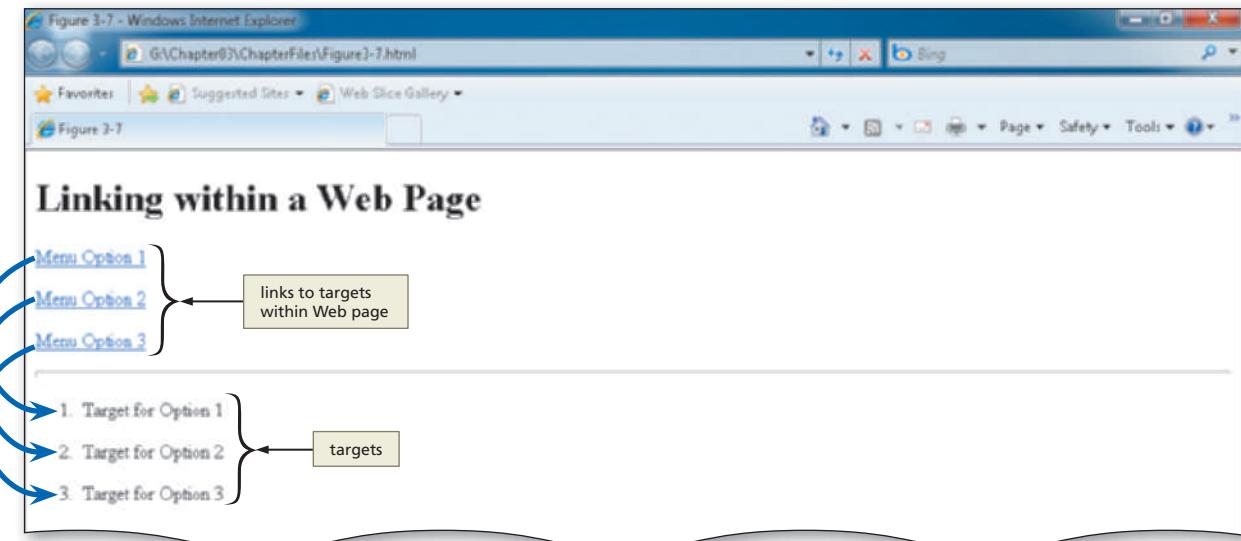


Figure 3–7 Web page with internal links.

BTW E-Mail Links

You can assign more than one e-mail address to a `mailto:` tag. Use the form `"mailto:first@isp.com, second@isp.com"` in the `` tag. Some older browsers may not support this tag.

Linking to an E-Mail Address

A well-designed Web page always provides a way for visitors to contact the person at the company responsible for maintaining the Web site or addressing customer questions and comments. An easy way to provide contact information is to include an e-mail link on the Web site's home page, as well as on other pages in the Web site. As shown in Figure 3–8, when a visitor clicks the **e-mail link**, it automatically opens a new message in the default e-mail program and inserts the appropriate contact e-mail address in the To field. Visitors then can type and send an e-mail to request additional information, comment on the

Web site, or notify the company of a problem with its Web site. (*Note:* If your browser is not configured to send e-mail, the e-mail link will not work.) The following HTML code shows an example of how to link the words gettinggreener@isp.com to an e-mail link.

```
<a href="mailto:gettinggreener@isp.com">gettinggreener@isp.com</a>
```

You again use the anchor `<a>` tag for this type of link. In the `href` attribute, though, you use the `mailto:`e-mail address as the value. It may seem strange to have the e-mail address `gettinggreener@isp.com` twice in this code. The first occurrence of the e-mail address is for the link itself. The second occurrence of `gettinggreener@isp.com` is used for the words on the Web page that you use as the link.

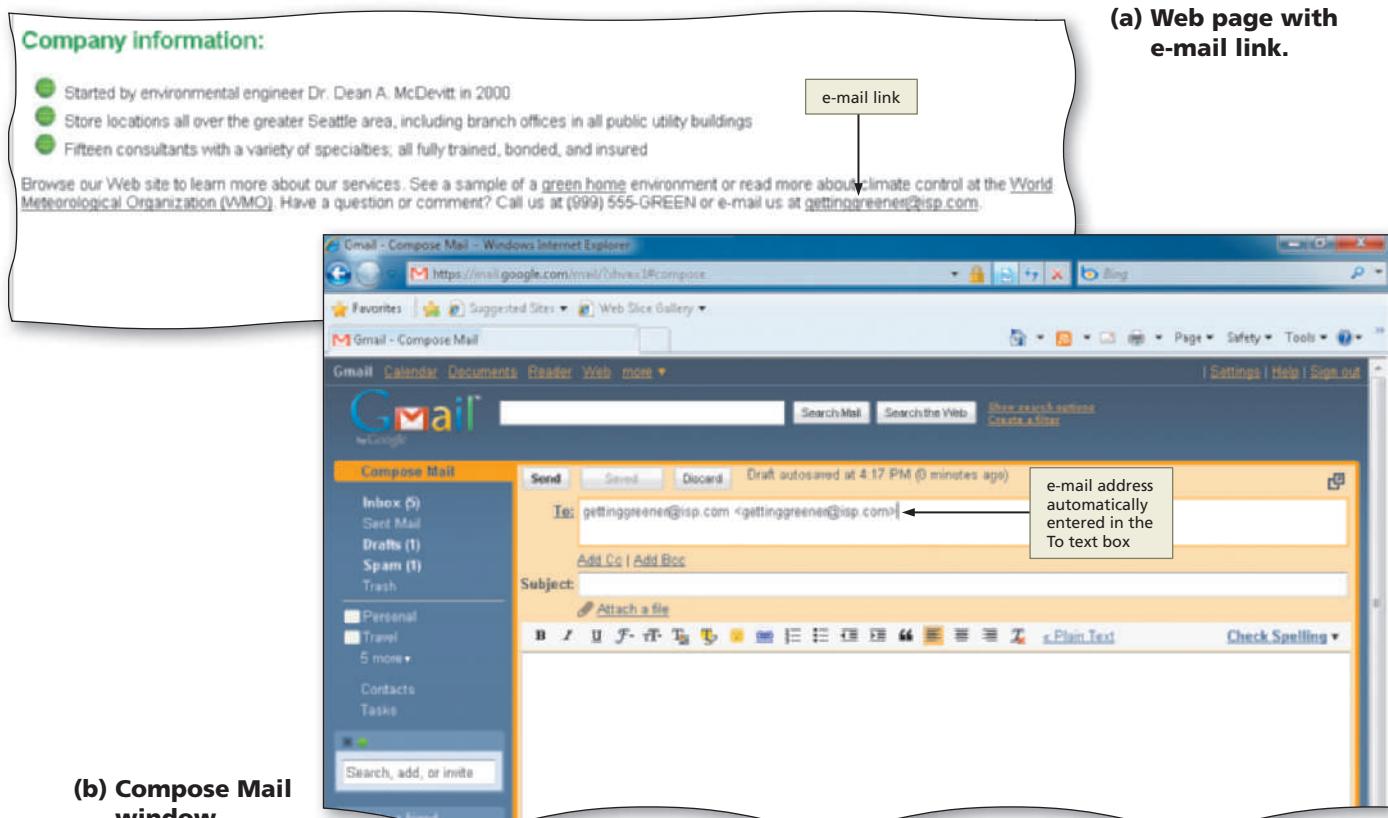


Figure 3–8

Creating a Home Page

The first Web page developed in this chapter is the home page of the Getting Greener Web site. A home page is the main page of a Web site, which visitors to a Web site generally will view first. A Web site home page should identify the purpose of the Web site by briefly stating what content, services, or features it provides. The home page also should indicate clearly what links the visitor should click to move from one page on the site to another. A Web developer should design the Web site in such a way that the links from one Web page to another are apparent and the navigation is clear. The Web site home page also should include an e-mail link, so visitors easily can find contact information for the individual or organization. Many Web sites now include an additional e-mail link to the Web development team. Users can utilize this e-mail link to notify the Web developers of any problems with the Web site or to comment on the site.

You begin creating the home page by starting Notepad++ and entering the initial HTML tags. Then you add an image, heading, text, and an unordered list to your home page. Finally, you add text and e-mail links, and then test the links.

To Start Notepad++

The following steps, which assume Windows 7 is running, start Notepad++ based on a typical installation. You may need to ask your instructor how to start Notepad++ for your computer.

- 1** Click the Start button on the Windows taskbar to display the Start menu.
- 2** Click All Programs at the bottom of the left pane on the Start menu to display the All Programs list.
- 3** Click Notepad++ in the All Programs list.
- 4** Click Notepad++ in the list to display the Notepad++ window. If there are files already open in Notepad from previous projects, close them all now by clicking the Close button on each open file.
- 5** If the Notepad++ window is not maximized, click the Maximize button on the Notepad++ title bar to maximize it (Figure 3–9).
- 6** Click View on the menu bar.
- 7** If the Word wrap command does not have a check mark next to it, click Word wrap.

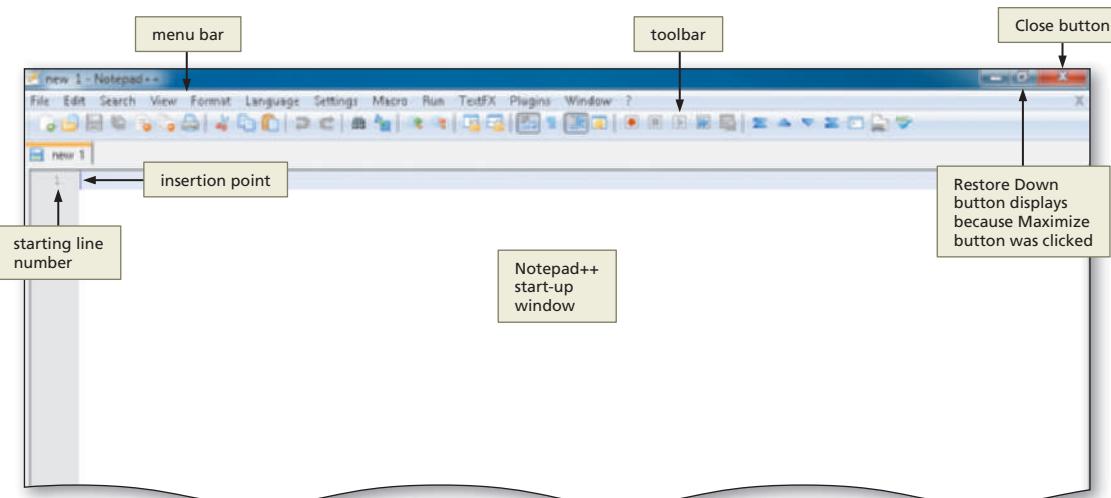


Figure 3–9

To Enter Initial HTML Tags to Define the Web Page Structure

BTW

Copy Initial Structure
Remember that you can type in the initial HTML tags and save that code in a file called structure.html, which you can then open and use as the basis for all HTML files. This eliminates the need for you to type this same code at the beginning of every HTML file. Just remember to save structure.html with a new name as soon as you open it.

Just as you did in Chapter 2, you start your file with the initial HTML tags that define the structure of the Web page. Table 3–1 contains the tags and text for this task.

Table 3–1 Initial HTML Tags

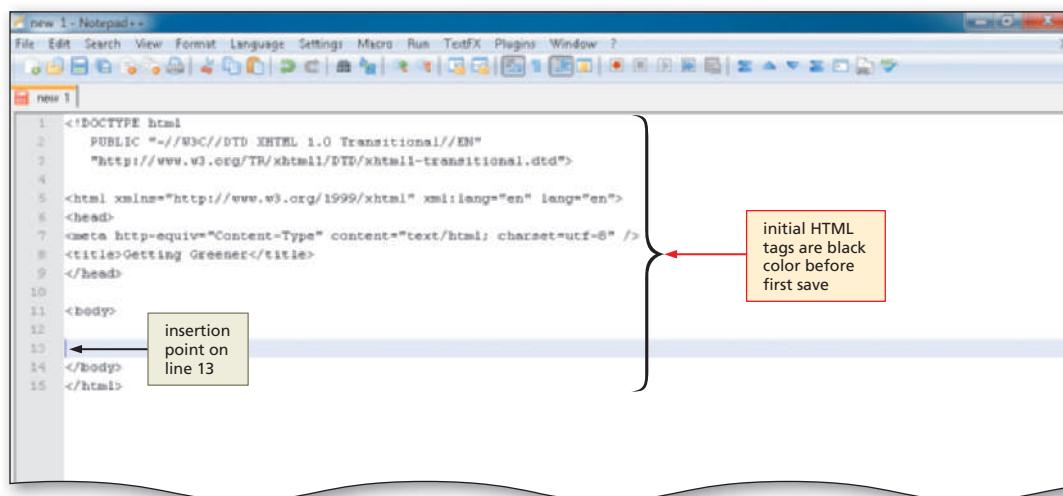
Line	HTML Tag and Text
1	<!DOCTYPE html
2	PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3	"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4	
5	<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

Table 3-1 Initial HTML Tags (continued)

Line	HTML Tag and Text
6	<head>
7	<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
8	<title>Getting Greener</title>
9	</head>
10	
11	<body>
12	
13	</body>
14	</html>

The following steps illustrate how to enter the initial tags that define the structure of the Web page.

- 1 Enter the HTML code shown in Table 3-1. Press enter at the end of each line. If you make an error as you are typing, use the backspace key to delete all the characters back to and including the incorrect characters, then continue typing.
- 2 Position the insertion point on the blank line between the <body> and </body> tags (line 12) and press the enter key (Figure 3-10).
- 3 Compare what you typed to Figure 3-10. If you notice errors, use your mouse pointer or arrow keys to move the insertion point to the right of each error and use the backspace key to correct the error.

**Figure 3-10**

To Save an HTML File

With the initial HTML code for the Getting Greener home page entered, you should save the file. Saving the file frequently ensures you won't lose your work. Saving a file in Notepad++ also adds color to code that can help you identify different elements more easily. The following step illustrates how to save an HTML file in Notepad++.

- 1** With a USB flash drive connected to one of the computer's USB ports, click File on the Notepad++ menu bar and then click Save.
- 2** Type `gettinggreener.html` in the File name text box (do not press enter).
- 3** Click Computer in the left pane of the Windows Explorer window to display a list of available drives.
- 4** If necessary, scroll until UDISK 2.0 (G:) or the name of your storage device is displayed in the list of available drives.
- 5** Open the Chapter03\ChapterFiles folder.
- 6** Click the Save button in the Save As dialog box to save the file on the USB flash drive in the Chapter03\ChapterFiles folder with the name `gettinggreener.html`.

Plan Ahead

Identify how to format various elements of the text.

Before inserting the graphical and color elements on a Web page, you should plan how you want to format them. By effectively utilizing graphics and color, you can call attention to important topics on the Web page without overpowering it. Consider the following formatting suggestions.

- **Effectively utilize graphics.** An important part of Web development is the use of graphics to call attention to a Web page. Generally, companies utilize the same logo on their Web site as they use on print material associated with the company, such as business cards and letterheads. Using the same graphical image on all marketing materials, including the Web site, is a good way to provide a consistent visual and brand message to customers.
- **Utilize headings that connect to the graphics.** In many cases, companies use the logo banner as the main heading on their home page, as opposed to using an `<h1>` heading. It is sometimes good to coordinate the color of the headings and graphics contained on the Web page to the logo banner. This can bring attention to the company logo banner, and it makes the Web page look cohesive with coordinating colors. Heading size standards should generally be followed from `h1` (the largest) to `h6` (the smallest). In this project, though, you use the company logo banner as your main heading, so you have no `h1` heading. You therefore start with the next smaller size heading, `h2`, as shown in Figure 3–1a on page HTML 91. Figure 3–1b on page HTML 91 shows the use of `h1` and `h2` headings in appropriate precedence.

To Add a Center-Aligned Banner Image Using an Inline Style

The Getting Greener home page includes an image logo banner to provide visual appeal, catch the visitor's interest, and promote the company's brand. The following steps illustrate how to add an image to a Web page using an inline style to center the image and an `` tag to insert the image. To use the inline style, you need the `<div>` `</div>` tags. Remember that the `<div></div>` tags create a container that defines logical divisions in your Web page. The `<div>` tag is similar to a paragraph tag, but it allows you to divide the page into larger sections and to define the style of whole sections within your Web page. You could define a section of your page and give that section a

different style from the surrounding text. When you use the `<div> </div>` tags, you are able to design a layout that uses CSS, including inserting images. You use the `<div></div>` tags in this case to center the image on the Web page. Table 3–2 contains the code for adding the centered logo banner.

Table 3–2 HTML Code for Adding a Center-Aligned Banner Image

Line	HTML Tag and Text
13	<code><div style="text-align: center"></code>
14	<code> </code>
15	<code></div></code>

1

- With the insertion point on line 13, enter the HTML code shown in Table 3–2, pressing ENTER at the end of each line. Make sure to indent the second line of code by using the TAB key.

This separates the start and end `<div>` tags from the `` tag, highlighting the image insertion. Press the ENTER key twice at the end of line 15 to position the insertion point on line 17 (Figure 3–11).

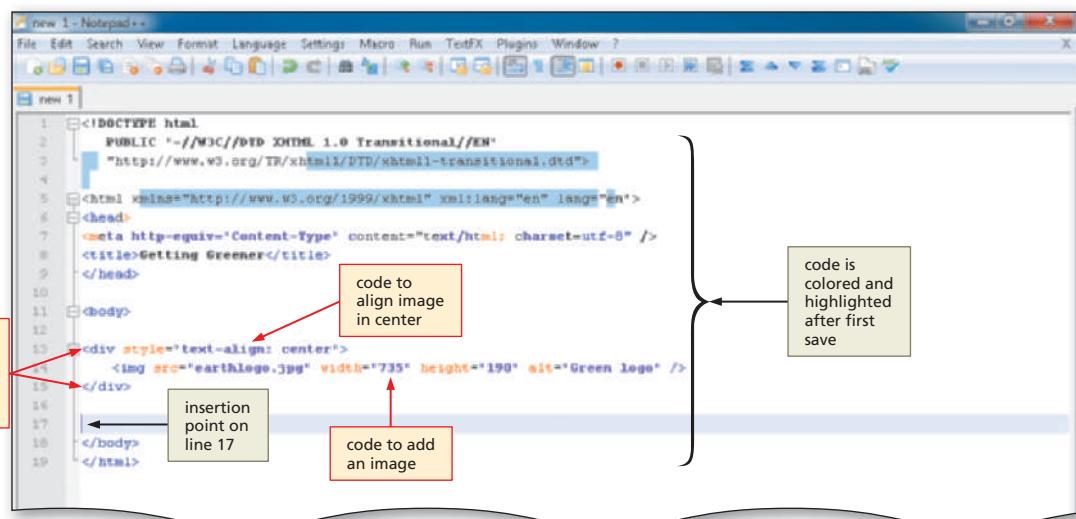


Figure 3–11

Q&A

Why should I include the width, height, and alt attributes?

Adding width and height attributes can improve page loading time because the browser does not have to figure the width and height before loading the image. Never use the height and width attributes to resize an image, however. Use graphic editing software to resize it and save it with a different file name. The height and width attributes as used in the `img` tag should reflect the actual image size. The alt attribute provides information about the purpose of the image when the user's mouse hovers over the image and while the image is loading.

Q&A

What is the purpose of the `<div> </div>` tags?

The `<div>` tag defines a division or a section in an HTML document. The `<div>` tag is often used to group elements to format them with styles. In this case, to center the image, you have to use the `<div> </div>` tags. The statement `style="text-align: center"` is the inline style that centers this image. An image is left-aligned by default.

**Plan
Ahead****Identify how to format text elements of the home page.**

You should always make a plan before inserting the text elements of a Web page. By formatting the characters and paragraphs on a Web page, you can improve its overall appearance. Effectively formatting the text also makes the message or purpose of the Web page clearer to the users. On a Web page, consider the following formatting suggestions.

- **Use default text size when appropriate.** The body text consists of all text between the heading and the bottom of the Web page. This text is the main content of the Web page and should be used to highlight the key points of your message. You can vary your content by utilizing both paragraphs of text and lists.
- **Determine what text formatting to use.** In a long Web page, it may help to vary your text as a way to break information up between headings. Using bold, color, or italicized text sparingly gives the Web page a more interesting look. Make sure not to overdo the formatting of text because you can make the page look cluttered. It is more difficult to find the content for which you are searching in a cluttered Web page.
- **Determine what style sheets to use.** Consider using style sheets to vary the format of text elements. If the text varies across paragraphs, an inline style is good to use. If you want all of the text in the Web page to be the same, an embedded style sheet is appropriate. If the text is common across more than one Web page, an external style sheet (discussed in the next chapter) should be used.
- **Highlight key text with ordered or unordered lists.** An ordered or unordered list contains specific information that is more clearly identified by a list versus a paragraph of text. In this project, you use a bulleted (unordered) list but vary it by changing the type of bullet used. Using an image as a bullet gives the Web page a nice look and is different than the standard (default) disc bullet for unordered lists.
- **Determine other information suitable for the home page.** Other information that is suitable for a home page includes: the company address (often found in the logo), a phone number, and an e-mail link.

To Add Paragraphs of Text

After the earth logo image for the Getting Greener home page is inserted, you need to add two paragraphs of text introducing Getting Greener. Table 3–3 shows the tags and text to enter.

Table 3–3 HTML Code for Adding Two Paragraphs of Text

Line	HTML Tag and Text
17	<p>Getting Greener is an environmental consulting company that specializes in helping to make your home and life greener. We have trained consultants who can give your home a green audit. We will make suggestions for ways that you can lessen your footprint on the earth. Help your family act in an environmentally responsible way by creating a home environment that is safe, less costly, and green.</p>
18	
19	<p>Browse our Web site to learn more about our services. See a sample of a green home environment or read more about climate control at the World Meteorological Organization (WMO). Have a question or comment? Call us at (999) 555-GREEN or e-mail us at gettinggreener@isp.com .</p>

- 1** With the insertion point on line 17, enter the HTML code shown in Table 3–3. Press enter twice after the `</p>` tag on line 17 and once after the `</p>` tag on line 19. After entering the two paragraphs, the insertion point is on line 20 (Figure 3–12).

Q&A Do I have to end all paragraphs of text with the `</p>` tag?

A Web page without `</p>` tags would display in the browser correctly. This Web page would not pass validation using the w3c Markup Validation Service, however. One missed `</p>` tag will result in many errors during validation.

Q&A What if I wanted the second paragraph to start without a blank line above it?

If you wanted the second paragraph to move to the next line without a blank line in between, you would use the `
` tag instead of `<p></p>`.

```

<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Getting Greener</title>
</head>
<body>
<div style="text-align: center">

</div>
<p>Getting Greener is an environmental consulting company that specializes in helping to make your home and life greener. We have trained consultants who can give your home a green audit. We will make suggestions for ways that you can lessen your footprint on the earth. Help your family act in an environmentally responsible way by creating a home environment that is safe, less costly, and green.</p>
<p>Browse our Web site to learn more about our services. See a sample of a green home environment or read more about climate control at the World Meteorological Organization (WMO). Have a question or comment? Call us at (999) 555-GREEN or e-mail us at gettinggreener@isp.com.</p>
</body>
</html>

```

Figure 3–12

Using an Inline Style to Add Interest to a Bulleted List

You've decided to highlight company information on the home page with an unordered (bulleted) list. To integrate the list with the rest of the page, and to add interest, you want to make the bullets the same as the earth symbol in the company logo. To do this, you'll add an h2 heading above the list and use an inline style to create the bullets.

Inserting an h2 heading above the unordered list visually separates the list from other elements on the Web page and indicates what the items in the list describe. An h1 heading is normally the main heading of the Web page, but in this case, the logo (earthlogo.jpg) is used in lieu of the h1 heading. It serves the same purpose to identify the company and call attention to the top of the Web page. Using an h1 heading under the logo banner would be redundant. So you will start with an h2 heading because it is smaller and an appropriate size to set off the bulleted list.

There are a variety of list-style types (bullets and numbers) that you can use, as described in Chapter 2. You used an inline style with the list-style-type property to make square bullets for the Chapter 2 project. For this Web page, you are going to use the list-style-image property to use a small image (earthbullet.jpg) instead of a disc, circle, or square bullet. You do this by identifying the image earthbullet.jpg as the URL to display in place of a bullet.

To Create an Unordered (Bulleted) List Using Images as Bullets

Table 3–4 shows the HTML code used to add an h2 heading and create a bulleted list with “earth” image bullets for the Getting Greener home page.

Table 3–4 HTML Code for Creating an Unordered (Bulleted) List with Image Bullets

Line	HTML Tag and Text
19	<h2>Company information:</h2>
20	
21	<ul style="list-style-image: url(earthbullet.jpg)">
22	Started by environmental engineer Dr. Dean A. McDevitt in 2000
23	Store locations all over the greater Seattle area, including branch offices in all public utility buildings
24	Fifteen consultants with a variety of specialties; all fully trained, bonded, and insured
25	

The following step shows how to create the unordered (bulleted) list that appears on the Getting Greener home page.

1

- Click the blank line 18 and press the ENTER key.
 - Enter the HTML code shown in Table 3–4.
 - After the in line 25, press the ENTER key to insert a blank line on line 26 (Figure 3–13).

Q&A

What if I wanted to use a different bullet type?

For an open circle bullet, use list-style-type: "circle". To use the default disc (filled circle) bullet, the list-style-type does not need to be included.

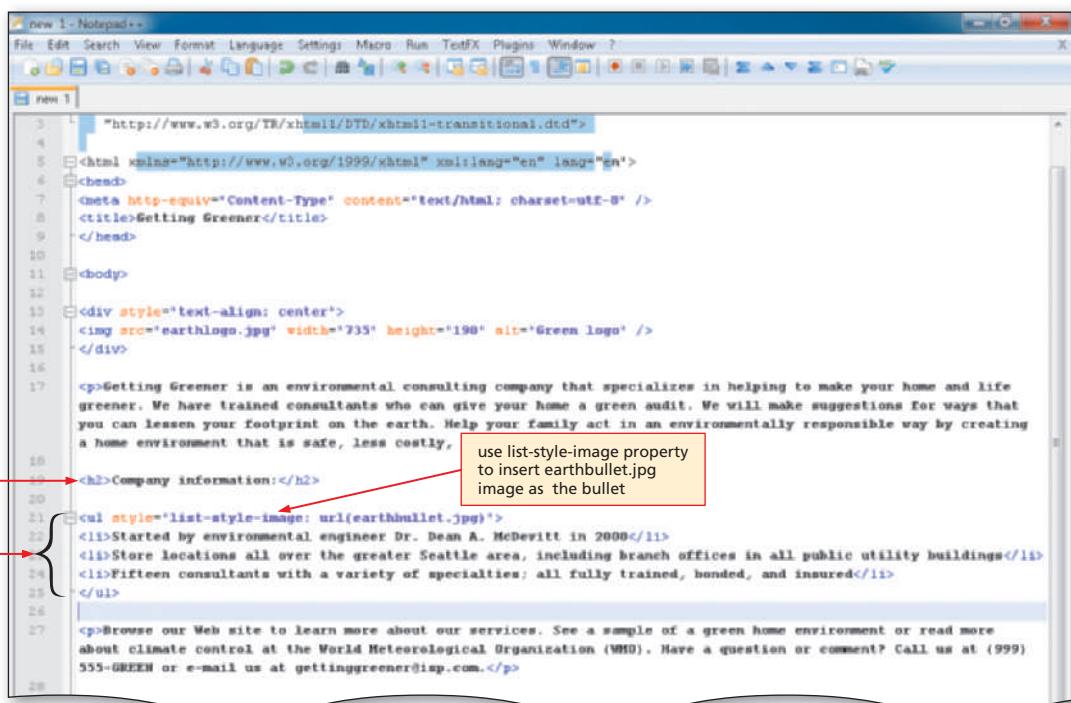


Figure 3-13

Plan how and where to use the four types of links.

- **Identify how to link from the home page to another page in the same Web site.** Linking to another Web page in the same Web site is often done with text links. When determining what words to use, make sure that the text links are clear and easy to understand. Using a phrase such as “click here” is not one that clearly identifies where the link will go. Choosing words such as “green home” tells the Web site visitor to click that link if they want to see a sample of a green home.
- **Use an e-mail link on the home page.** A good standard practice is to include an e-mail link on the home page. Again, using words such as “click here” are not as effective as using a company’s actual e-mail address (gettinggreener@isp.com in this case) as the e-mail link text.
- **Determine external links for the home page.** Visitors to a Web site might want additional information on a topic, so a link also can be included on the home page. Linking to an external Web site (i.e., one that is outside of the boundaries of the current Web site) is appropriate to provide additional information. Again, it is important to select words or phrases that make sense for that link.
- **Use internal links on long Web pages.** Another good standard practice is to include links within a Web page when the page is long (i.e., when you have to press the PAGE DOWN key several times to get to the end of the Web page). Internal links help visitors navigate more easily within long Web pages. Also consider using links to help the visitor easily return back to the top of a long Web page.

Plan Ahead

Adding a Text Link to Another Web Page Within the Same Web Site

For the purpose of this Web site, the `<a>` and `` tags are used to create links on a Web page. The `<a>` tag also is called the **anchor tag** because it is used to create anchors for links to another page in the same Web site, to a Web page in an external Web site, within the same Web page, and for e-mail links. The anchor tag can also be used to specify the base language of the target URL or to specify the media type of the link. The `href` attribute stands for a hyperlink reference. This is a reference (an address) to a resource on the Web. Hyperlinks can point to any resource on the Web, including an HTML page, an image, a sound file, or a video. The basic form of the tag used to create a link is:

```
<a href="URL">linktext</a>
```

where `linktext` is the clickable word or phrase that is displayed on the Web page and the value for `href` (hypertext reference) is the name or URL of the linked page or file. Table 3–5 shows some of the `<a>` tag attributes and their functions.

BTW

Other Links

You also can create a link to other Web pages (that is, non-`http`), an FTP site, and newsgroups. To link to an FTP site, type `ftp://URL` rather than `http://URL` as used in this project. For a newsgroup, type `news:newsgroup name`, and for any particular article within the newsgroup, type `news:article name` as the entry.

Table 3–5 `<a>` Tag Attributes and Functions

Attribute	Function
<code>href</code>	Specifies the URL of the linked page or file.
<code>name</code>	Defines a name for the current anchor so it may be the target or destination of another link. Each anchor on a Web page must use a unique name.
<code>rel</code>	Indicates a forward relationship from the current document to the linked document. The value of the <code>rel</code> attribute is a link type, such as <code>prev</code> , <code>next</code> , <code>index</code> , or <code>copyright</code> . For example, the Web page <code>chapter3.html</code> might include the tag <code></code> to indicate a link to the Web page for the next chapter, <code>chapter4.html</code> .

Table 3–5 <a> Tag Attributes and Functions (continued)

Attribute	Function
rev	Indicates a reverse (backward) relationship from the current document to the linked document. The value of the rev attribute is a link type, such as prev, next, index, or copyright. For example, the chapter3.html Web page might include the tag <code></code> to indicate a link to the Web page for the previous chapter, chapter2.html.
type	Specifies the content type (also known as media types or MIME types) of the linked page or file to help a browser determine if it can handle the resource type. Examples of content types include text/html, image/jpeg, video/quicktime, application/java, text/css, and text/javascript.

Before creating a link, be sure you know the URL or name of the file to be linked and the text that will serve as the clickable word or phrase. The words should be descriptive and tell the Web page visitor the purpose of the link. For the Getting Greener home page, the text link is a phrase in a paragraph at the bottom of the Web page.

To Add a Text Link to Another Web Page Within the Same Web Site

The Getting Greener home page includes a text link to the Green Home Web page, which is part of the same Web site. The following step illustrates how to add a text link to another Web page within the same Web site.

1

- Click immediately to the left of the g in the word green on line 27.
- Type `` to start the link, setting the Web page greenhome.html as the linked Web page.
- Click immediately to the right of the e in home on line 27. Type `` to close the link (Figure 3–14).

```

16 <p>Getting Greener is an environmental consulting company that specializes in helping to make your home and life
17 greener. We have trained consultants who can give your home a green audit. We will make suggestions for ways that
18 you can lessen your footprint on the earth. Help your family act in an environmentally responsible way by creating
19 a home environment that is safe, less costly, and green.</p>
20
21 <h2>Company information:</h2>
22 <ul style="list-style-image: url(earthbullet.jpg)">
23 <li>Started by environmental engineer Dr. Dean A. McDevitt in 2000</li>
24 <li>Store locations all over the greater Seattle area, including branch offices in all public utility buildings</li>
25 <li>Fifteen consultants with a variety of specialties; all fully trained, bonded, and insured</li>
26 </ul>
27 <p>Browse our Web site to learn more about our services. See a sample of a <a href="greenhome.html">green home</a>
28 environment or read more about climate control at the World Meteorological Organization (WMO). Have a question or
29 comment? Call us at (999) 555-GREEN or e-mail us at gettinggreener@isp.com. </p>
30
</body>
</html>

```

Figure 3–14

Q&A

What is the href attribute for?

The href stands for “hypertext reference” and precedes the URL of the destination Web page.

Q&A

How will I know if my text is a link when it is displayed in the browser?

In the browser, the mouse pointer turns into a pointing finger where there is a link. Also, as the default, text used as a link will be blue and underlined. You can change the color and style of a link, and you do that later in the chapter.

Q&A

What happens if I forget to insert the `` tag on a link?

A text link without the `` tag will not display correctly in the browser. If you forget to use the `` tag to end this text link, all of the text beyond the `` tag will serve as that link. In this example, all of the text that follows the g in green will link to the greenhome.html Web page, which is certainly not what you want.

Adding an E-Mail Link

Adding an e-mail link is similar to adding a text link, but instead of using a URL as the href attribute value, the href attribute value for an e-mail link uses the form:

```
<a href="mailto:address@email.com">linktext</a>
```

where the href attribute value uses the word mailto to indicate it is an e-mail link, followed by a colon and the e-mail address to which to send the e-mail message. When the browser recognizes a **mailto** URL in a clicked link, it automatically opens a new message in the default e-mail program and inserts the appropriate contact e-mail address in the To field. The clickable text used for an e-mail link typically is the e-mail address used in the e-mail link. The Web page also should provide some information before the link, so visitors know the purpose of the e-mail link.

BTW

E-Mail Links

E-Mail Links may not work in a school setting because there may not be an e-mail client installed on the school servers. You therefore may not be able to test this link.

To Add an E-Mail Link

The Getting Greener home page includes an e-mail link so customers can contact Getting Greener for additional information or to comment on the Web page. The <a> and tags used to create a text link to a Web page also are used to create an e-mail link. The following step shows how to add an e-mail link to a Web page.

1

- Click immediately to the left of the g in the beginning of gettinggreener@isp.com on line 27. Type as the start of the e-mail link. This will link to the e-mail address gettinggreener@isp.com when the link is clicked.

```

18
19
20
21 <h2>Company Information:</h2>
22
23 <ul style="list-style-type: none;">
24 <li>Started by environmental engineer Dr. Dean A. McDevitt in 2000</li>
25 <li>Stores locations all over the greater Seattle area, including branch offices in all pu
26 <li>Fifteen consultants with a variety of specialties; all fully trained, bonded, and insu
27 <p>Browse our Web site to learn more about our services. See a sample of a <a href="greenhome.html">green home</a>
environment or read more about climate control at the World Meteorological Organization (WMO). Have a question or
comment? Call us at (999) 555-GREEN or e-mail us at <a href="mailto:gettinggreener@isp.com">gettinggreener@isp.com
</a></p>

```

line 28 → end e-mail link

e-mail link with mailto: attribute

text that displays on Web page surrounded by e-mail link

Figure 3-15

- Click immediately after the m in isp.com and before the period in the e-mail address text on line 27.
- Type to end the e-mail link, as shown in Figure 3-15.

Q&A

I see two occurrences of gettinggreener@isp.com on line 27. Why do I need two?

The first occurrence of gettinggreener@isp.com (the one within the link <a> tag following the mailto:) is the destination of the link. The second occurrence of gettinggreener@isp.com is the text link itself that will be displayed in the browser.

Adding Other Information to an E-Mail Link

Sometimes, you need to add a message in the body of the e-mail in addition to the subject. This technique can be very helpful when more than one e-mail link is positioned on a Web page, and each link has a different purpose. For instance, one e-mail might be used for general questions, whereas another link might be used for specific information. You also can include a carbon-copy (cc) address. For instance, to include just a subject or to include a subject and body message text in the above mailto:, you would complete the following steps.

To Add a Subject to an E-Mail Link

- 1 Type `` as the tag.

Sometimes, you need to add a message in the body of the e-mail in addition to the subject. This technique can be very helpful when more than one e-mail link is positioned on a Web page, and each link has a different purpose. For instance, one e-mail might be used for general questions, whereas another link might be used for specific information. Using the subject and body attributes can be helpful for this scenario. Notice that the two attributes (subject and body) are separated by an ampersand in the following example. The following step shows how to add the subject “green home” to the e-mail together with the message text “How can I get a green audit?” as shown in Figure 3–16.

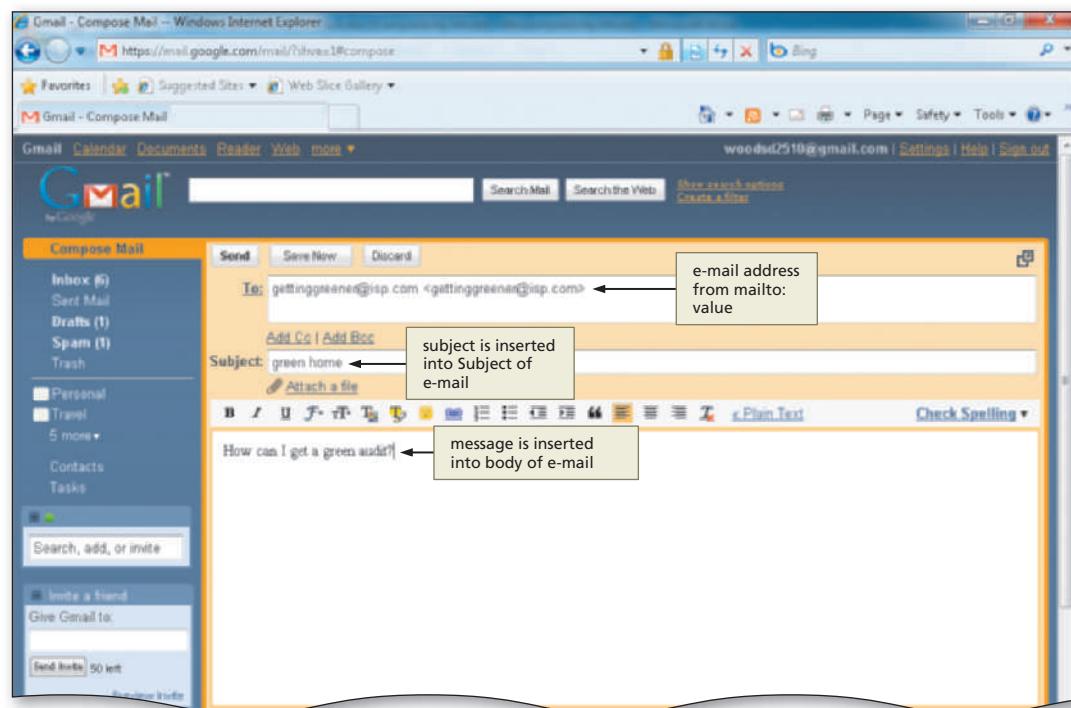


Figure 3–16

To Add a Subject Together with Body Message Text

- 1 Type `` as the tag.

To Add a Text Link to a Web Page in Another Web Site

The <a> and tags used to create a text link to a Web page within the same Web site also are used to create a link to a Web page in another Web site. The following step illustrates how to add a text link on the Green Home Web page to an external Web page that describes climate control information.

1

- Click immediately to the left of the W in World on line 27 and type to add the text link that will connect to the external Web site when clicked.

```

18
19
20
21 <h2>Company information</h2>
22
23 <ul style="list-style-image: url(earthbullet.jpg)">
24 <li>Started by environmental engineer Dr. Dean A. McDevitt in 2000</li>
25 <li>Stores locations all over the greater Seattle area, including branch</li>
26 <li>Fifteen consultants with a variety of specialties; all fully trained</li>
27 <p>Browse our Web site to learn more about our services. See a sample of our< a href="greenhome.html">green home</a> environment or read more about climate control at the <a href="http://www.wmo.int/">World Meteorological Organization (WMO)</a>. Have a question or comment? Call us at (999) 555-GREEN or e-mail us at< a href="mailto:gettinggreener@isp.com">gettinggreener@isp.com</a>.</p>
</body>

```

HyperText Markup Language File nb char : 1647 nb line : 30 Ln : 27 Col : 248 Sel : 0

Figure 3–17

- Click immediately to the right of the) in (WMO on line 27 and type to end the tag, as shown in Figure 3–17.

Q&A

How do these links appear when displayed in the browser?

If the default is used, the text link is displayed in a blue, underlined font to indicate it is a link. The mouse pointer also changes to the pointing hand when moved over the link text. You change that color later in the chapter.

Q&A

When I type in the URL in the Address box of my browser, I never type in the http:// part of the URL. Why do I have to add the http:// in the link?

Although you do not need to type the http:// into the URL on the browser, you always must include this as part of the href when creating external links. See the discussion on absolute and relative paths below for more information.

Q&A

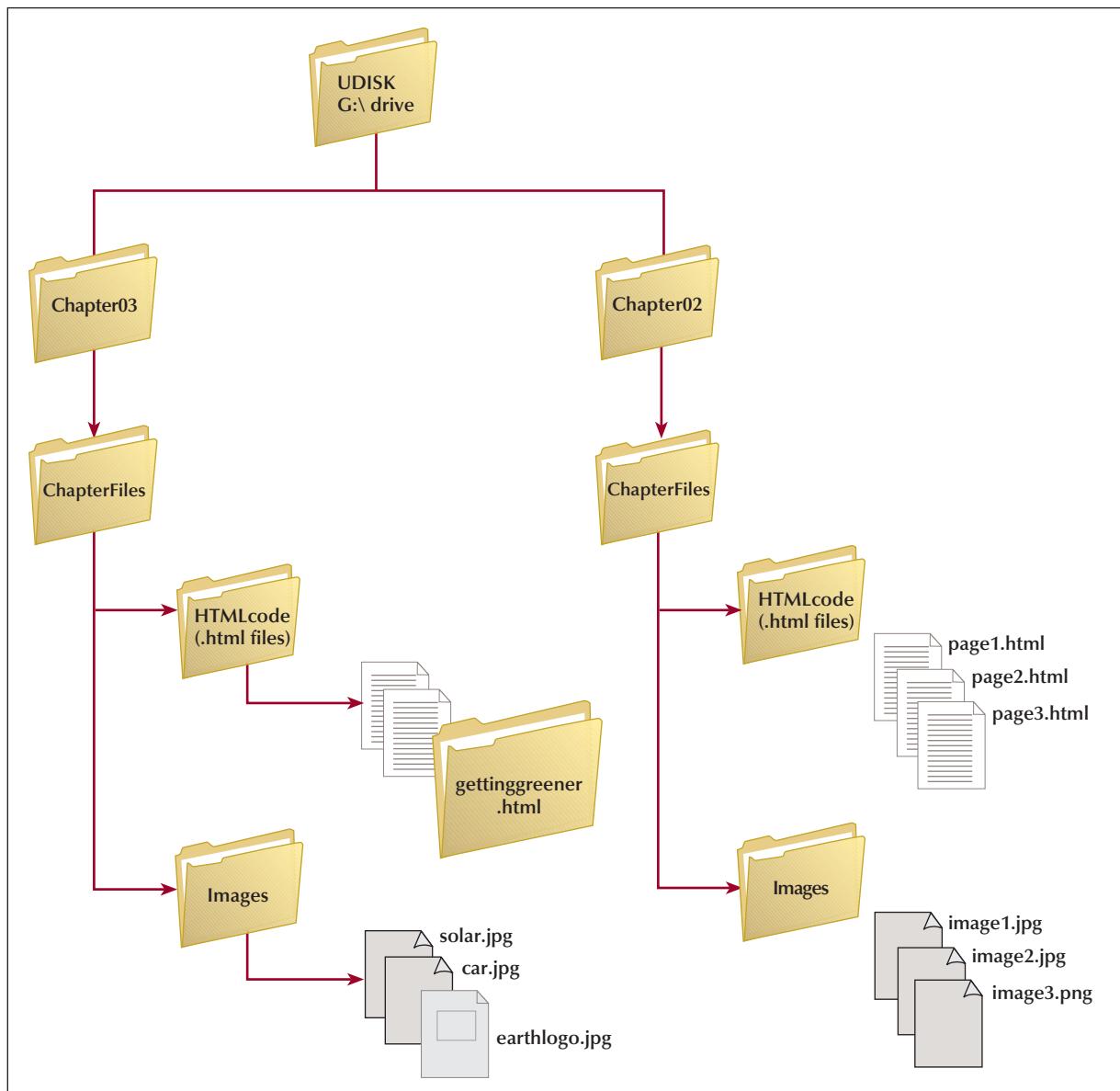
Why did I need the http:// part of the URL for this external link, but I did not need that for the green home link?

The Green Home Web page is stored in the same folder as the home page from which you are linking. You, therefore, do not need to include any information other than the name of the Web page file. Review the following section on absolute and relative paths for more information.

Using Absolute and Relative Paths

At this point, it is appropriate to revisit the overall concept of how the files are organized and saved. As noted in the last chapter, the projects in this book use a very simple folder structure. In this book, the graphical images are stored in the same folder as the HTML files, for example, in the Chapter03\ChapterFiles folder. For most real-world applications, however, it would be more appropriate to separate the HTML code and the graphical images into different folders. Figure 3–18 on the next page shows an example of a more complex file structure that could be used for this book.

To understand how to use this sort of folder structure, you need to identify the folder location, or path, to the files. A **path** describes the location (folder or external Web site) where the files can be found, beginning with the UDISK G:\ drive (or another drive

**Figure 3–18**

on your computer). This beginning location also is known as root. You can use either an absolute or relative path when identifying the location of the files. An **absolute path** specifies the exact address for the file to which you are linking or displaying a graphic. You can think of an absolute path as the complete address of a house, including the house number, street name, city, state, and zip. In order to use that absolute address, you would have to give the entire path (or address) to a person who wants to get to that particular house. When you are referencing a Web page from a server outside of the server on which your Web pages reside, you have to use an absolute path. In this chapter, you use the absolute path to the World Meteorological Organization (WMO). This is because that Web page is located outside of the server (or storage media) on which the Web pages created in the chapter reside. Your link statement for this external Web site is:

```
<a href="http://www.wmo.int/">World Meteorological Organization (WMO)</a>
```

Although absolute paths indicate the specific addresses of files, they can be cumbersome. If you have to move any of the files to a different folder or a different Web server, then all absolute paths would have to change.

Relative paths specify the location of a file, relative to the location of the file that is currently in use. This is analogous to telling someone your house is located four doors down from the only gas station on that street. Your address in this case is relative to the beginning point, the gas station. Because your user has a beginning point (the gas station), you can describe the ending address (the house) relative to the beginning. A relative path allows you to move up the folder structure. So in the example in which you want to display the image `earthlogo.jpg` (stored in the `Images` subfolder) from the Web page `gettinggreener.html` (stored in the `HTMLcode` subfolder) within the `Chapter03\ChapterFiles` folder, you would use the following relative path structure:

```
\Images\earthlogo.jpg
```

Looking at Figure 3–18, you would store the image `earthlogo.jpg` in the `Images` folder and store the Web page itself, the `gettinggreener.html` file, in the `HTMLcode` subfolder. If you moved to the `HTMLcode` subfolder and viewed the `gettinggreener.html` file, the image `earthlogo.jpg` would not appear because it is not in the same subfolder. To display the `gettinggreener.html` file with the `earthlogo.jpg` image, you would use the following relative path structure:

```

```

Another example is the relative addressing that you use in this chapter to link to the second Web page from the home page, and vice versa. The HTML code to link from the home page, `gettinggreener.html` to the second Web page is:

```
<a href="greenhome.html">green home</a>
```

and to go from the `greenhome.html` Web page back to the home page, your HTML code is:

```
<a href="gettinggreener.html">home page</a>
```

For another example, let's see what the code would look like if you had two folders for this chapter project. You currently have one folder `ChapterFiles` in the `Chapter03` folder. To have the same basic folder structure as you see in Figure 3–18, you would create a second folder named `Images` that is on the same folder level as `ChapterFiles`. So you would then have two folders in the `Chapter03` folder. If you moved your images (`solar.jpg` and `car.jpg`) to that `Images` folder, your HTML code to access those images would look like this:

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

It is better to use relative paths for flexibility whenever feasible. If the root folder (i.e., the “highest” folder in the hierarchy) must change for some reason, you do not have to change all addressing if you used relative paths. As an example, if you had to change

from g:\Chapter03\ to another folder, all related subfolders would automatically change. With absolute addressing, all paths would have to be individually changed.

Adding Interest and Focus with Styles

In Chapter 2, you learned how to vary the size of headings with the `<h1>` through `<h6>` tags. Any text on a Web page, including headings, can be formatted with a different color or style to make it stand out by using style properties. Table 3–6 lists some properties that can be used to enhance standard text on a Web page using styles. Remember that CSS and styles are the preferred technique to satisfy XHTML standards.

BTW**Font Properties**

Refer to Appendix D for a more complete list of CSS font properties and values. You can also set font characteristics with the HTML `` tag (see Appendix A), but this deprecated tag is not used in this project.

Table 3–6 Font Properties and Values

Property	Function
color	<ul style="list-style-type: none"> Changes the font color Can use a six-digit color code or color name
font-family	<ul style="list-style-type: none"> Changes the font face or type Values include fonts, such as Verdana or Arial; text appears using the default font if the font face is not specified
font-size	<ul style="list-style-type: none"> Changes the font size. Value can be an actual numeric size, a percentage, or values such as large, medium, small, etc.
font-style	<ul style="list-style-type: none"> Changes the style of a font Values include normal, italic, and oblique
font-weight	<ul style="list-style-type: none"> Changes the weight of a font Values include normal, bold, bolder, and lighter

Figure 3–19 shows how several of these attributes affect the appearance of text.

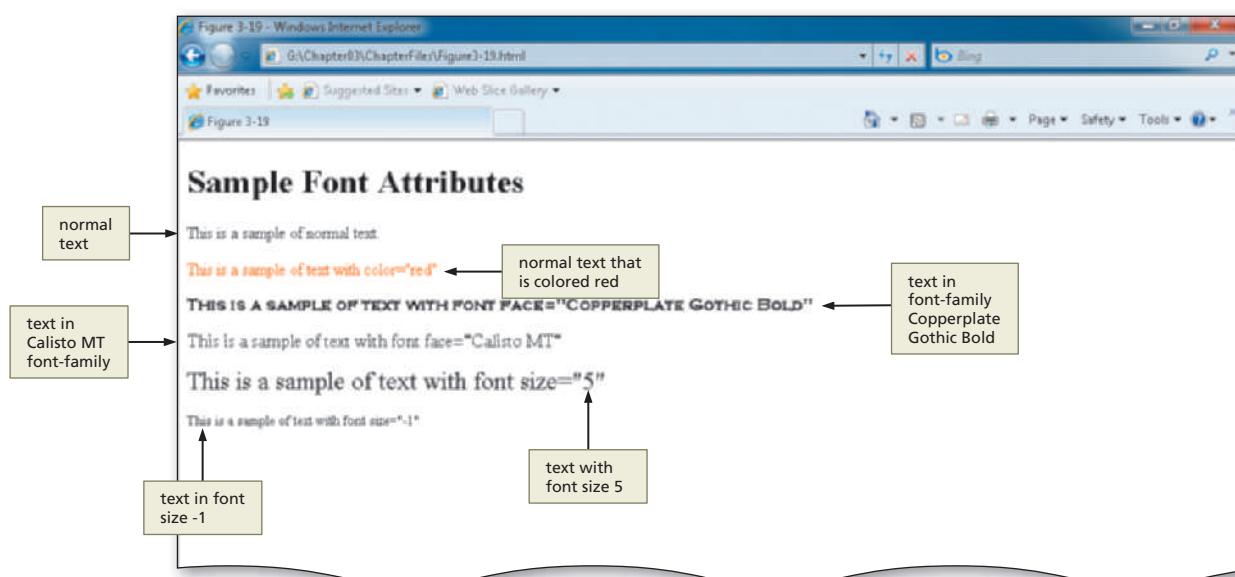


Figure 3–19 Examples of various fonts.

Adding Interest and Focus with HTML Tags

There is another way to format text that is compliant with the XHTML 1.0 Transitional standard used in this book. Web pages that use the Transitional DOCTYPE statement allow the use of deprecated tags, as explained in Chapter 2. These Web pages validate the HTML elements and attributes, including deprecated elements, successfully. Text can also be formatted using the formatting tags in HTML. Earlier in the project, you changed the font-weight of some text on the Web page using an inline style. Instead of that style sheet, you could have also used the HTML bold tags (** **) that make text bold. HTML provides a number of tags that can be used to format text, several of which are listed in Table 3–7. These cannot be used for the XHTML standard above 1.0, but they can be used for the XHTML 1.0 Transitional standard.

Table 3–7 Text Formatting Tags

HTML Tag	Function
<code> </code>	Physical style tag that displays text as bold
<code><big> </big></code>	Increases the font size in comparison to the surrounding text
<code><blockquote> </blockquote></code>	Designates a long quotation; indents margins on sections of text
<code> </code>	Logical style tag that displays text with emphasis (usually appears as italicized)
<code><i> </i></code>	Physical style tag that displays text as italicized
<code><pre> </pre></code>	Sets enclosed text as preformatted material, meaning it preserves spaces and line breaks; often used for text in column format in another document pasted into HTML code
<code><small> </small></code>	Decreases the font size in comparison to the surrounding text
<code> </code>	Logical style tag that displays text with strong emphasis (usually appears as bold)
<code><sub> </sub></code>	Displays text as subscript (below normal text)
<code><sup> </sup></code>	Displays text as superscript (above normal text)
<code><tt> </tt></code>	Displays text as teletype or monospace text

BTW

Deprecated and Obsolete Tags

A deprecated element or attribute is one that has been outdated. Deprecated elements may become obsolete in the future, but most browsers continue to support deprecated elements for backward compatibility. You can still use deprecated tags with a Transitional document type as used in this book. Obsolete elements and attributes have no guarantee of browser support, and they are no longer defined in the W3C specification.

BTW

Logical versus Physical Styles

For more information about the differences between logical and physical styles, search the Web for the key words “HTML logical style” or “HTML physical style”.

Figure 3–20 shows a sample Web page with some HTML text format tags. These tags fall into two categories: logical style tags and physical style tags. **Logical style tags** allow a browser to interpret the tag based on browser settings, relative to other text on a Web page. The `<h2>` heading tag, for example, is a logical style that indicates the heading text should be larger than regular text but smaller than text formatted using an `<h1>` heading tag. The `` tag is another logical style, which indicates that text should have a strong emphasis, and which most browsers interpret as displaying the text in bold font. **Physical style tags** specify a particular font change that is interpreted strictly by all browsers. For example, to ensure that text appears as bold font, you would enclose it between a start `` and end `` tag. The `` tag is a better fit for XHTML standards, and it does not dictate how the browser displays the text. In practice, the `` and `` tags usually have the same result when the Web page is displayed.

(a) Formatting samples.

The screenshot shows a Windows Internet Explorer window titled "Figure 3-20 - Windows Internet Explorer". The address bar shows the file path "G:\Chapter03\ChapterFiles\Figure3-20.html". The page content is titled "Sample Formatting Tags" and contains the following text samples with annotations:

- normal text**: "This is a sample of normal text with no formatting."
- emphasized text**: "This is a sample of **strong** text." (Annotation: "text in strong font style")
- text with superscript and subscript**: "This is a sample of text that is surrounded by the start and end **blockquote** tags. Using these tags blocks the text in from the right and the left margins. This technique can be used to emphasize quotations of text that are large." (Annotation: "text surrounded by **blockquote** tags to indent it")
- underlined text**: "This is a sample of underlined text."
- text that looks like teletype**: "This is teletype or monospace text."

(b) HTML code for formatting tags.

```

<h1>Sample Formatting Tags</h1>
This is a sample of normal text with no formatting.

<strong>This is a sample of strong text.</strong>

<em>This is a sample of emphasized text.</em>

<blockquote>This is a sample of text that is surrounded by the start and end blockquote tags. Using these tags blocks the text in from the right and the left margins. This technique can be used to emphasize quotations of text that are large.</blockquote>

This is a sample of using the <sub>subscript</sub> and <sup>superscript</sup> tags.

<tt>This is teletype or monospace text.</tt>

<u>This is a sample of underlined text.</u>

```

Figure 3-20

BTW**CSS Types**

Remember that an inline style changes the style of an individual element; an embedded style sheet changes the style of an entire Web page; and an external style sheet changes the style in multiple Web pages in the same Web site. If you want to change the style of a single element, use an inline style.

Style Sheet Precedence Review

In Chapter 2, you learned how to insert an inline style. The project in this chapter also uses an inline style and introduces you to embedded style sheets. You learn about the third and final form of style, external style sheets, in Chapter 4. It will be helpful at this point to review the information from Chapter 2 on the precedence of styles (see Table 2-6 on page HTML 63). An inline style is used to control the style within an individual HTML tag and takes precedence over both embedded and external style sheets. An embedded sheet is used to change the style of an element over one Web page, but overrides or takes precedence over any styles defined in an external style sheet. An external style sheet is a separate document with a .css extension that is used to define styles for multiple pages in a Web site.

Because styles have different levels of precedence, all three types of styles can be used on a single Web page. For example, in this part of the chapter, you define body, anchor, and heading styles with embedded style sheets that are used for both the getting-green.html file and the greenhome.html file. You also insert a few inline styles on each Web page. Because of the precedence rules, the inline styles take precedence over the embedded style sheets. For instance, if you use an embedded style sheet to make all paragraphs Garamond font-family in normal text type and size 12, you can override that font-family, style, and size for a specific paragraph with an inline style within that paragraph's

<p> tag. Maybe there is a paragraph that you want to highlight, so you make it bold with an inline style. Or maybe there is a paragraph that you want to downplay, and you make it smaller and italic with an inline style. It is important to determine how and when to use the various styles in the design phase of Web development.

Identify which level of style or style sheet to use.

Because of precedence rules, it is generally better to look at the broadest level style first. In this chapter project, you use inline and embedded style sheets, with embedded being the broader level. In other words, an embedded style sheet is used for the entire Web page, and an inline style is used in a particular HTML tag. In Chapter 4, you will add an external style sheet (the broadest level) to your chapter project.

- **Identify what styles need to be different than the standards used across the Web site.** Sometimes you need to vary a style in order to call attention to the content or pull attention away from it. Many Web sites have a legal statement on the bottom of the home page. That is not necessarily something that needs to be the same font-size as the rest of the content on that Web page. So you may choose a smaller font-size and maybe make the text italic for that content. Two types of style sheets are used for styles that are different across a Web site, embedded and inline.
- **Use embedded style sheets to affect a single Web page.** This type of style is good to use if you want the style to affect just one (or a few) Web pages, and not all pages across the Web site.
- **Use inline styles for individual styles.** If you want to change the style of one or a few elements of one Web page, then using inline styles is the most appropriate. If a style is intended for most (or all) of the Web page, consider using an embedded or external style sheet.

Plan Ahead

Using Embedded Style Sheets

An embedded style sheet is used to control the style of a single Web page. To add an embedded style sheet to a Web page, you insert a start <style> tag at the top of the Web page within the <head> </head> tags that define the header section. After adding the desired style statements, you end the embedded style sheet by adding an end </style> tag.

The following code shows an example of an embedded style sheet to set the h1 heading to the Garamond font family, point size 32. This code would be added between the <head> and </head> tags within <style> </style> tags:

```
<style type="text/css">
h1 {font-family: Garamond;
    font-size: 32pt}
</style>
```

In this embedded style, the h1 (header size 1) element is the **selector**, and the remainder of the code is the **declaration**. The declaration sets the values for two different properties. The first property-value statement sets the h1 font family to Garamond. The second property-value statement sets the font size to 32 point. This means that the browser will display all h1 headers in 32-point Garamond font. You could use this embedded style sheet to easily change all h1 headings, in lieu of making the same change with an inline style in each individual heading tag.

The various types of Cascading Style Sheets allow you to control many different property values for various elements on a Web page. Table 3–8 lists six main properties and related options that are used in CSS. A complete list of properties and property values that can be used in CSS is included in Appendix D.

BTW**HTML/CSS Terminology**

In HTML, a tag is a special instruction to the browser to specify how the Web page is displayed. Many tags have attributes that help to further modify what is displayed. In CSS, a style statement is made up of a selector and a declaration. The part of the style statement that identifies the page element(s) is called the *selector*. The part of the style statement that identifies how the element(s) should appear is called the *declaration*. A declaration includes at least one type of style, or property, to apply to the selected element.

Table 3–8 CSS Properties and Options

Property Name	Options That Can Be Controlled
background	<ul style="list-style-type: none"> • color • image • position
border	<ul style="list-style-type: none"> • color • style • width
font	<ul style="list-style-type: none"> • family • size • style • variant • weight
list	<ul style="list-style-type: none"> • image • position • type
margin	<ul style="list-style-type: none"> • length • percentage
text	<ul style="list-style-type: none"> • alignment • decoration • indentation • spacing • white space

BTW**Specifying Alternative Fonts**

If a Web page font is not available on users' computers, you can create a list of fonts and the browser will determine the font to use. For example, if the Web page uses a Geneva font, but Arial or Helvetica would also work well, you create a comma-separated list of acceptable fonts, using your text as the code. If a Web page uses a font that Web page visitors do not have on their computers, the Web page appears using a default font (usually Times New Roman).

The following code shows an example of an embedded style sheet that you will use in the chapter project:

```
<style type="text/css">
body {font-family: Arial, Verdana, Garamond;
      font-size: 11pt}
h1, h2 {color: #00934a}
a {color: black}
a:hover {background: #00934a;
          color: white}
</style>
```

This embedded style sheet defines four elements on the page: body, headings, links, and the link-hover property. The first style statement uses the **body** selector to specify that all text on the Web page should be one of the font families: Arial, Verdana, or Garamond, in 11 point size. Computers do not always have every font-family installed, so Web developers usually specify multiple font-families. If the first font-family is not available, then the next takes effect. If none of the named font-families are installed, the computer's default font is used. Separate the font-families by commas.

The second style statement defines values for the h1 and h2 properties. The value #00934a will give all h1 and h2 headings on this Web page the color green. On the home page, there is no h1 heading, but there is one on the second Web page, and later in the

chapter you will use this same embedded style sheet for that Web page, which has both h1 and h2 headings.

The third style statement defines one property of the link element. The selector **a** is used to indicate the link element. The property-value statement *color: black* changes from the default blue color for links. Because the style statement uses **a** as the selector, it changes all link states (normal, visited, active) to these property values. You also can define a unique style for normal, visited, and active links by creating three separate style statements with **a:link**, **a:visited**, and **a:active** as the selectors.

The last style statement uses the **a:hover** selector to define the style of a link when the mouse pointer points to, or **hovers** over, a link. In this statement, you use a pseudo-class (**hover**) to have more control over the anchor (link) element. A **pseudo-class** is attached to a selector with a colon to specify a state or relation to the selector to give the Web developer more control over that selector. The format to use with a pseudo-class is entered in the form:

```
selector:pseudo-class { property: value; }
```

with a colon between the selector and the pseudo-class. There are four pseudo-classes that can be used when applied to the anchor or link selector:

- link, for an unvisited link
- visited, for a link to a page that has already been visited
- active, for a link when it gains focus (for example, when it is clicked)
- hover, for a link when the cursor is held over it.

The hover statement tells the browser to display white link text on a green (#00934a) background when the mouse hovers over the link. Adding a link hover style significantly changes the look of the links and adds a dimension of interactivity to the Web page.

The `<!--` and `-->` code used in the embedded style sheet (just after the start style `<style>` and just before the end style `</style>` tags) are comment lines. These comment lines tell the browser to ignore whatever is between the comment lines if the browser cannot interpret the code between. So if your Web page user has a browser that is not current, it may not be able to interpret embedded style sheets. An older browser would see the start of the comment `<!--` and treat anything between that line and the `-->` as a comment. This is a good Web development technique.

Recall that embedded style sheets have the second-highest level of precedence of the three types of styles. Although an inline style overrides the properties of an embedded style sheet, the embedded style sheet takes precedence over an external style sheet.

To Add Embedded Style Sheet Statements

Table 3–9 shows the CSS code for an embedded style sheet to specify the font and size for body text on this Web page.

Table 3–9 CSS Code for an Embedded Style Sheet

Line	CSS Selectors and Declarations
9	<code><style type="text/css"></code>
10	<code><!--</code>
11	
12	<code>body {font-family: Arial, Verdana, Garamond;</code>

Table 3–9 CSS Code for an Embedded Style Sheet (continued)

Line	CSS Selectors and Declarations
13	font-size: 11pt}
14	
15	h1, h2 {color: #00934a}
16	
17	a {color: black}
18	
19	a:hover {background: #00934a;
20	color: white}
21	
22	-->
23	</style>

The following step shows how to enter the embedded style sheet code to change h1 and h2 headings to green, change all links to blank, and change the color of the link hover to green background with white text to provide visual impact.

1

- Click immediately to the right of the > in </title> on line 8 and press the ENTER key.
- Type the code in Table 3–9 but do not press the ENTER key at the end of line 23 (Figure 3–21).

Q&A

What other styles can I use on my Web pages?

Appendix D lists available CSS properties and values. You can also search the Web for examples of how CSS are used for Web development. Finally, be an active Web page visitor and review the source code on Web pages with styles that you think are particularly effective or ineffective.

```

<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Getting Greener</title>
<style type="text/css">
<!-->
<body> (font-family: Arial, Verdana, Garamond; font-size: 11pt)
<h1, h2 {color: #00934a}
<a {color: black}
<a:hover {background: #00934a; color: white}
-->
</style>
</head>
<body>
<div style="text-align: center">

</div>

```

Figure 3–21

Is there a way to eliminate the underline on text links altogether?

You can use a:hover text-decoration: none to eliminate the underline completely. Understand, however, that users look for underlined text when they try to find the links on a Web page, so use this cautiously.

To Add an Inline Style for Color

The following step shows how to enter an inline style to add a green color (#00934a) in a bold font-weight to provide visual impact and call attention to the company name, Getting Greener. With this inline style, you use the ` ` tags. The ` ` tags create a container into which a user can add an inline style. The `` provides a finer level of control for styles, as opposed to the `<div> </div>` tags, which define block-level structure or division in the HTML document. The `` tag tells the browser to apply the chosen styles to whatever is within the `` container.

1

- With the insertion point right after the `>` in `<p>` on line 32, type `` (Figure 3-22).
- With the insertion point right after the `r` in Greener on line 32, type `` (Figure 3-22).

Q&A

What other font-weights could I have used?

The four font-weights are normal, bold, bolder, and lighter. If you don't specify a font-weight, normal is the default.

```

new 1 - Notepad++
File Edit Search View Format Language Settings Macro Run TestFX Plugins Window ?
new 1
12 body {font-family: Arial, Verdana, Garamond;
13     font-size: 11pt}
14
15 h1, h2 {color: #00934a}
16
17 a {color: black}
18
19 a:hover {background: #00934a;
20     color: white}
21
22
23
24
25
26
27
28
29
30
31
32 <p><span style="color: #00934a; font-weight: bold">Getting Greener</span> is an environmental consulting company
33 that specializes in helping to make your home and life greener. We have trained consultants who can give your home
34 a green audit. We will make suggestions for ways that you can lessen your footprint on the earth. Help your family
35 act in an environmentally responsible way by creating a home environment that is safe, less costly, and green.</p>
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
59
60
61
62
63
64
65
66
67
68
69
69
70
71
72
73
74
75
76
77
78
79
79
80
81
82
83
84
85
86
87
88
89
89
90
91
92
93
94
95
96
97
98
99
99
100
101
102
103
104
105
106
107
108
109
109
110
111
112
113
114
115
116
117
118
119
119
120
121
122
123
124
125
126
127
128
129
129
130
131
132
133
134
135
136
137
138
139
139
140
141
142
143
144
145
146
147
148
149
149
150
151
152
153
154
155
156
157
158
159
159
160
161
162
163
164
165
166
167
168
169
169
170
171
172
173
174
175
176
177
178
179
179
180
181
182
183
184
185
186
187
187
188
189
189
190
191
192
193
194
195
196
197
197
198
199
199
200
201
202
203
204
205
206
207
207
208
209
209
210
211
212
213
214
215
216
216
217
218
218
219
219
220
221
222
223
224
225
225
226
227
227
228
228
229
229
230
231
232
233
234
235
235
236
236
237
237
238
238
239
239
240
241
242
243
244
244
245
245
246
246
247
247
248
248
249
249
250
251
252
253
253
254
254
255
255
256
256
257
257
258
258
259
259
260
260
261
261
262
262
263
263
264
264
265
265
266
266
267
267
268
268
269
269
270
270
271
271
272
272
273
273
274
274
275
275
276
276
277
277
278
278
279
279
280
280
281
281
282
282
283
283
284
284
285
285
286
286
287
287
288
288
289
289
290
290
291
291
292
292
293
293
294
294
295
295
296
296
297
297
298
298
299
299
300
300
301
301
302
302
303
303
304
304
305
305
306
306
307
307
308
308
309
309
310
310
311
311
312
312
313
313
314
314
315
315
316
316
317
317
318
318
319
319
320
320
321
321
322
322
323
323
324
324
325
325
326
326
327
327
328
328
329
329
330
330
331
331
332
332
333
333
334
334
335
335
336
336
337
337
338
338
339
339
340
340
341
341
342
342
343
343
344
344
345
345
346
346
347
347
348
348
349
349
350
350
351
351
352
352
353
353
354
354
355
355
356
356
357
357
358
358
359
359
360
360
361
361
362
362
363
363
364
364
365
365
366
366
367
367
368
368
369
369
370
370
371
371
372
372
373
373
374
374
375
375
376
376
377
377
378
378
379
379
380
380
381
381
382
382
383
383
384
384
385
385
386
386
387
387
388
388
389
389
390
390
391
391
392
392
393
393
394
394
395
395
396
396
397
397
398
398
399
399
400
400
401
401
402
402
403
403
404
404
405
405
406
406
407
407
408
408
409
409
410
410
411
411
412
412
413
413
414
414
415
415
416
416
417
417
418
418
419
419
420
420
421
421
422
422
423
423
424
424
425
425
426
426
427
427
428
428
429
429
430
430
431
431
432
432
433
433
434
434
435
435
436
436
437
437
438
438
439
439
440
440
441
441
442
442
443
443
444
444
445
445
446
446
447
447
448
448
449
449
450
450
451
451
452
452
453
453
454
454
455
455
456
456
457
457
458
458
459
459
460
460
461
461
462
462
463
463
464
464
465
465
466
466
467
467
468
468
469
469
470
470
471
471
472
472
473
473
474
474
475
475
476
476
477
477
478
478
479
479
480
480
481
481
482
482
483
483
484
484
485
485
486
486
487
487
488
488
489
489
490
490
491
491
492
492
493
493
494
494
495
495
496
496
497
497
498
498
499
499
500
500
501
501
502
502
503
503
504
504
505
505
506
506
507
507
508
508
509
509
510
510
511
511
512
512
513
513
514
514
515
515
516
516
517
517
518
518
519
519
520
520
521
521
522
522
523
523
524
524
525
525
526
526
527
527
528
528
529
529
530
530
531
531
532
532
533
533
534
534
535
535
536
536
537
537
538
538
539
539
540
540
541
541
542
542
543
543
544
544
545
545
546
546
547
547
548
548
549
549
550
550
551
551
552
552
553
553
554
554
555
555
556
556
557
557
558
558
559
559
560
560
561
561
562
562
563
563
564
564
565
565
566
566
567
567
568
568
569
569
570
570
571
571
572
572
573
573
574
574
575
575
576
576
577
577
578
578
579
579
580
580
581
581
582
582
583
583
584
584
585
585
586
586
587
587
588
588
589
589
590
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
600
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
133
```

To Validate HTML Code

- 1** Open Internet Explorer.
- 2** Navigate to the Web site validator.w3.org
- 3** Click the Validate by File Upload tab.
- 4** Click the Browse button.
- 5** Locate the `gettinggreener.html` file on your storage device and click the file name.
- 6** Click the Open button.
- 7** Click the Check button. A successful validation should be displayed, as shown in Figure 3–23a. If you have errors in your code, you may see a screen similar to Figure 3–23b. In this example, the errors relate to a missing `</p>` tag.

Q&A

What if my HTML code does not validate?

If your code has errors, you should edit your HTML file to correct the errors. The Markup Validation Service report lists clearly what is wrong with your code. Once you make the necessary changes and save the file, you can again use the Browse button to open the corrected HTML file. You then use the Revalidate button to validate the changed code.

Q&A

Why is the first error shown in Figure 3–23b on line 34, not line 32, where the `</p>` is missing?

Usually, the validator shows the exact error and line number on which the error occurs. Sometimes, however, the validator shows subsequent errors that occur as a result of the initial error. You may have to look a line prior to the first error line shown to find the initial error, as in this case.

BTW

Common Validation Errors

Common validation errors include not spelling tags, selectors, or attributes correctly; using uppercase letters (except for DOCTYPE); and not nesting tags correctly. A single coding error can cause many lines of errors during validation. For instance, Figure 3–23b shows a Web page that has seven errors caused by a single missing `</p>` tag on line 32.

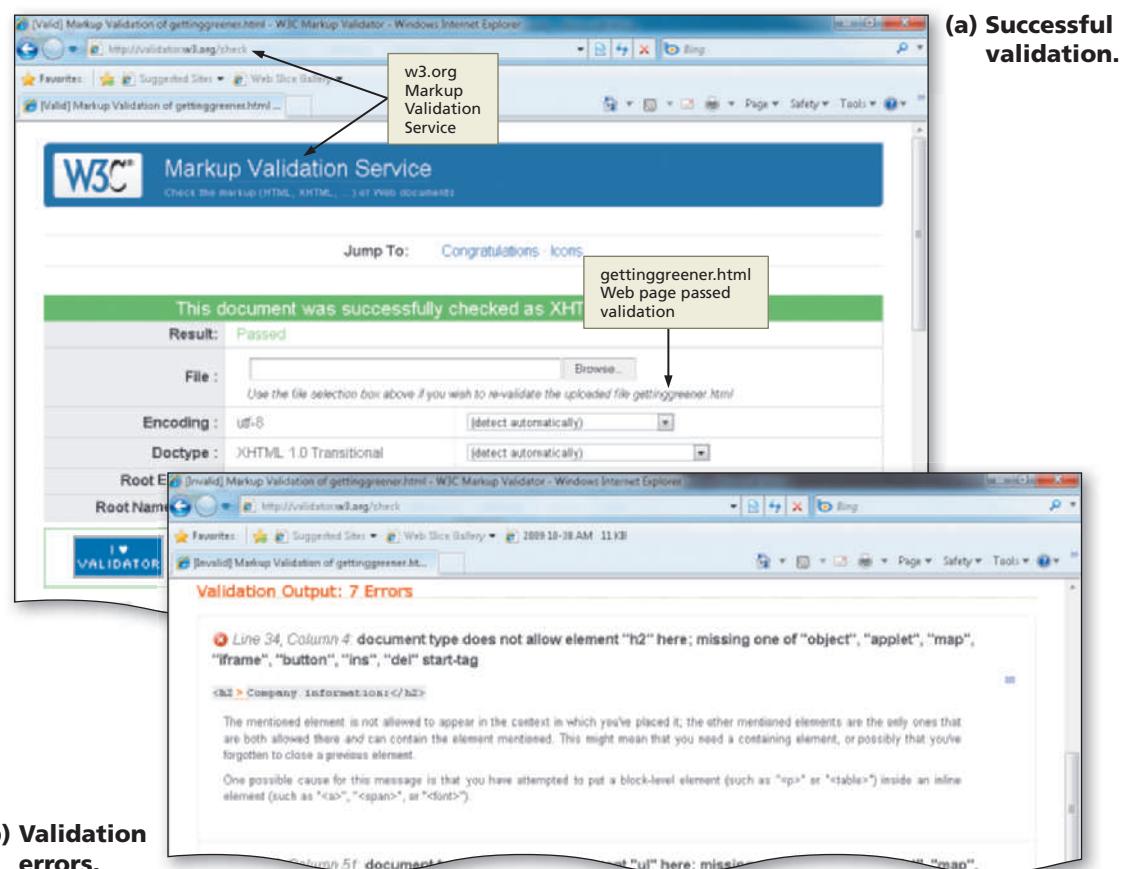


Figure 3-23

To Print an HTML File

After your HTML code has passed validation, it's a good idea to make a hard copy printout of it.

- 1** Click the Notepad++ button on the taskbar to activate the Notepad++ window.
- 2** Click File on the menu bar and then click the Print command, and then click the Print button to print a hard copy of the HTML code (Figure 3-24).

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Getting Greener</title>
<style type="text/css">
<!--
body {font-family: Arial, Verdana, Garamond;
      font-size: 11pt}

h1, h2 {color: #00934a}

a {color: black}

a:hover {background: #00934a;
          color: white}
-->
</style>
</head>

<body>

<div style="text-align: center">
    
</div>

<p><span style="color: #00934a; font-weight: bold">Getting Greener</span> is an
environmental consulting company that specializes in helping to make your home and life
greener. We have trained consultants who can give your home a green audit. We will make
suggestions for ways that you can lessen your footprint on the earth. Help your family
act in an environmentally responsible way by creating a home environment that is safe,
less costly, and green.</p>

<h2>Company information:</h2>

<ul style="list-style-image: url(earthbullet.jpg)">
<li>Started by environmental engineer Dr. Dean A. McDevitt in 2000</li>
<li>Store locations all over the greater Seattle area, including branch offices in all
public utility buildings</li>
<li>Fifteen consultants with a variety of specialties; all fully trained, bonded, and
insured</li>
</ul>

<p>Browse our Web site to learn more about our services. See a sample of a <a href=
"greenhome.html">green home</a> environment or read more about climate control at the <a
href="http://www.wmo.int/">World Meteorological Organization (WMO)</a>. Have a
question or comment? Call us at (999) 555-GREEN or e-mail us at
<a href="mailto:gettinggreener@isp.com">gettinggreener@isp.com</a>.</p>

</body>
</html>
```

Figure 3-24

To View a Web Page

The following steps illustrate how to view the HTML file in a browser.

- 1** Open Internet Explorer.
- 2** In Internet Explorer, click the Address bar to select the URL in the Address bar.
- 3** Type `g:\Chapter03\ChapterFiles\gettinggreener.html` (or the specific path to your file) to display the new URL in the Address bar and then press the **ENTER** key (Figure 3–25).

Q&A

What if my page does not display correctly?

Check your `gettinggreener.html` code carefully in Notepad++ to make sure you have not made any typing errors or left anything out. Correct the errors, re-save the file, and try again.

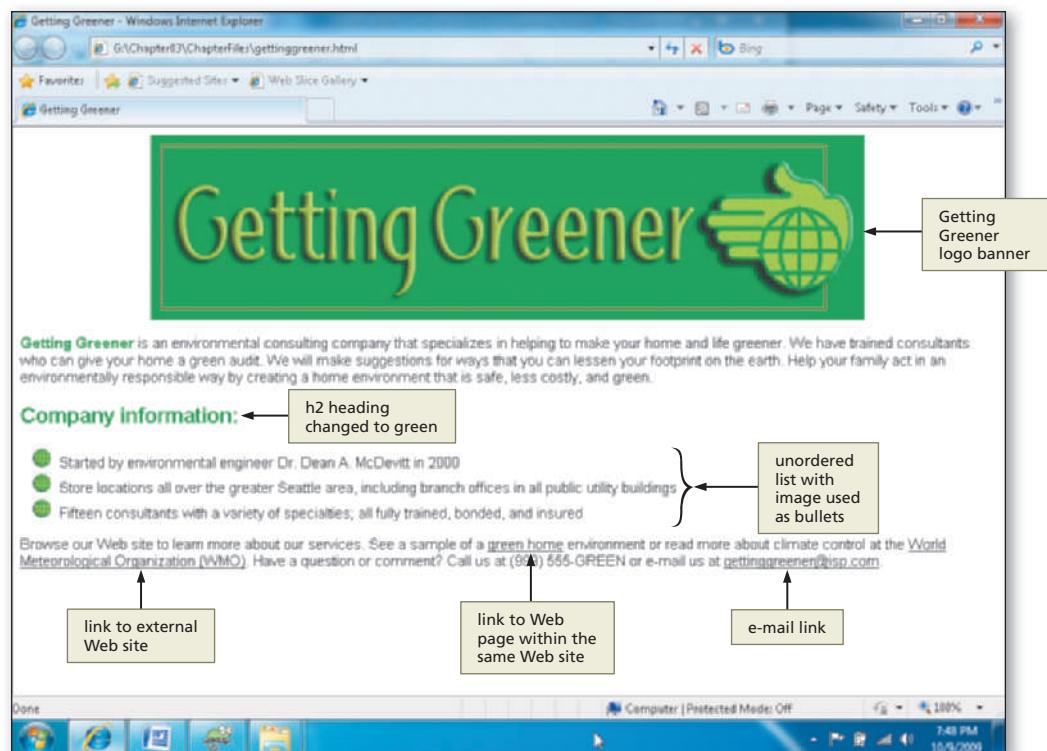


Figure 3–25

Test your Web page.

- **Determine what you need to test.** It is important to have a test plan when you test your Web pages. Planning what to test assures that all functionality of the Web page is tested. You should specifically test the display of the Web page itself and test that all of the links on the Web page work correctly.
- **Test the Web page as displayed in the browser.** Certainly the first part of testing is to verify that your Web page is displayed in the browser as intended. Ask yourself the following questions: (1) Are the images all displayed where they should be? (2) Is the text presented as intended? (3) Are the links displayed as intended?
- **Test the links.** In your testing plan, you need to address all of the links that you have inserted into the Web page. It is especially important to test external links, that is, those over which you have no control. If you need to link outside of the Web pages that you developed, then periodically test the links to make sure they are still valid. It helps to create a matrix that includes three columns for information. The first column contains information about all of the links on the Web page. The second column contains information about the intended results of those links. The third column is the one that you complete during testing. If the link tests as it should, you can note that by putting a check mark in the third column. If the link test result is not as it should be, you can note in the third column what the result is. Using a technique such as this makes it easier to do thorough testing. When you know what the results of the test should be, it helps you verify valid links. This is an excellent technique to use when there are different people developing and testing the Web pages. The matrix will notify the developers of the test results clearly.

Plan Ahead**BTW Web Page Testing**

An important part of Web page development is testing Web page links. For more information about link testing, search the Web for key words such as "HTML testing" or look at the World Wide Web Consortium (w3.org) Web site.

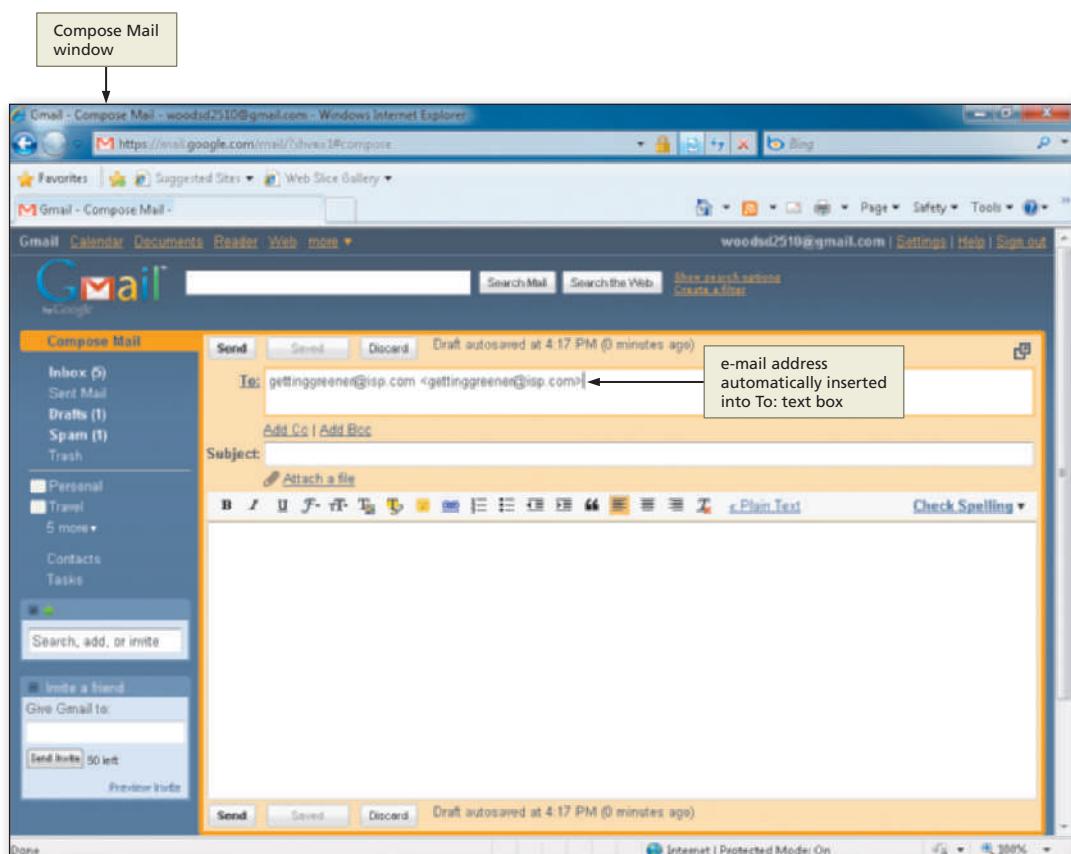
To Test Links on a Web Page

The following steps show how to test the links in the Getting Greener home page to verify that they work correctly.

1

- With the Getting Greener home page displayed in the browser, point to the e-mail link, gettinggreener@isp.com and then click the link to open the default e-mail program with the address gettinggreener@isp.com in the To: text box, as shown in Figure 3–26.

- Click the Close button in the Compose Mail window. If a dialog box asks if you want to save changes, click No.

**Figure 3–26**

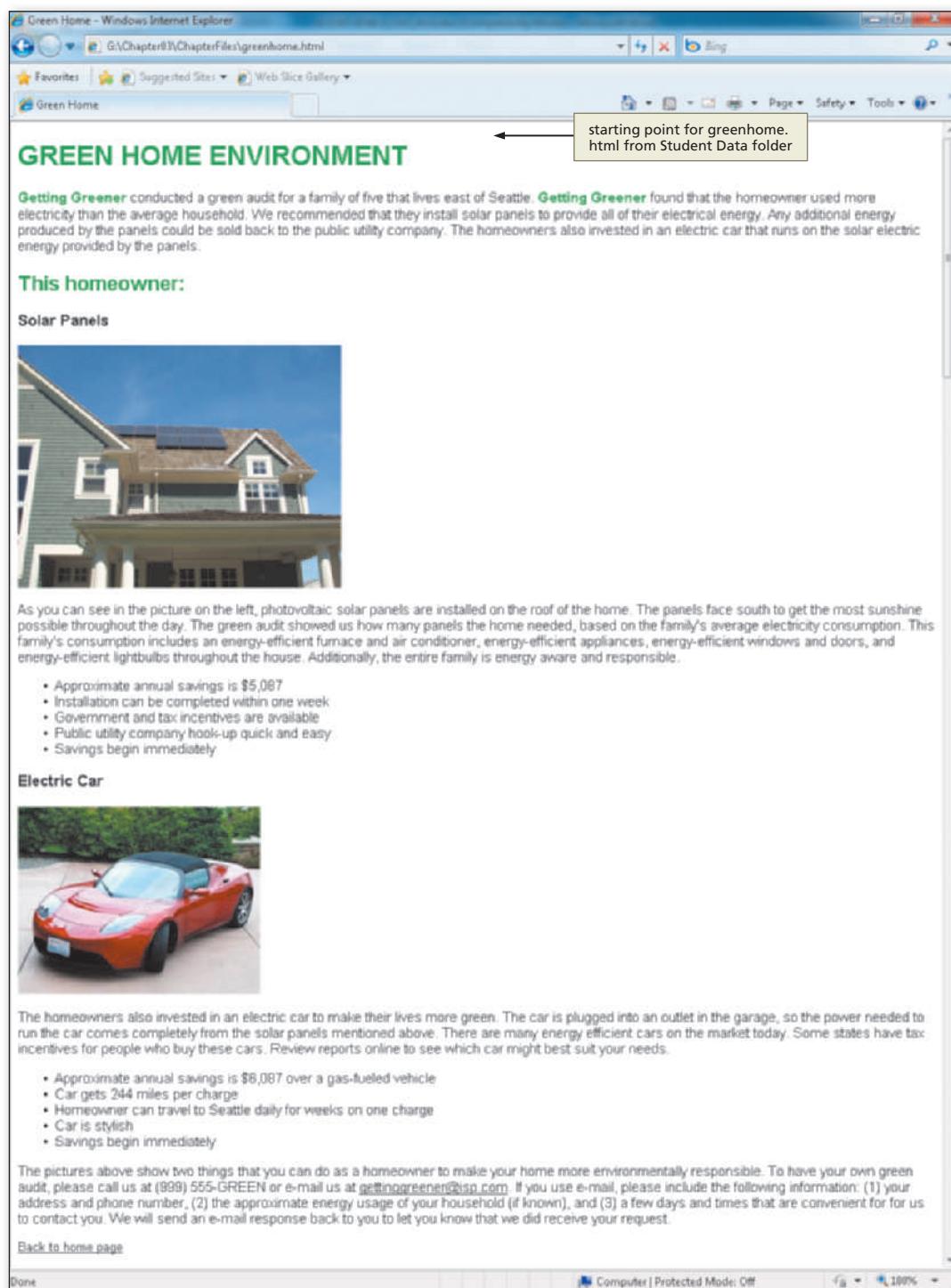
2

- Click the WMO link to test the external link on the Web page. Close the browser window or use the Back button to return to the Getting Greener home page.
- With the USB flash drive in drive G, point to the green home link and click the link. The secondary Web page, greenhome.html, is displayed (Figure 3–27), although it is not completed.

Q&A

My e-mail does not work when I click the link. Why does that happen?

You may not have an e-mail client installed on your computer or your school's servers. You therefore may not be able to test this e-mail link.

**Figure 3–27**

To Print a Web Page

Print the Web page for future reference.

- 1** Close the browser window or click the Back button on the Standard toolbar to return to the Getting Greener home page.
- 2** Click the Print icon on the Command bar.
- 3** Once the Getting Greener home page is printed (Figure 3–28), click the green home link to return to that Web page.

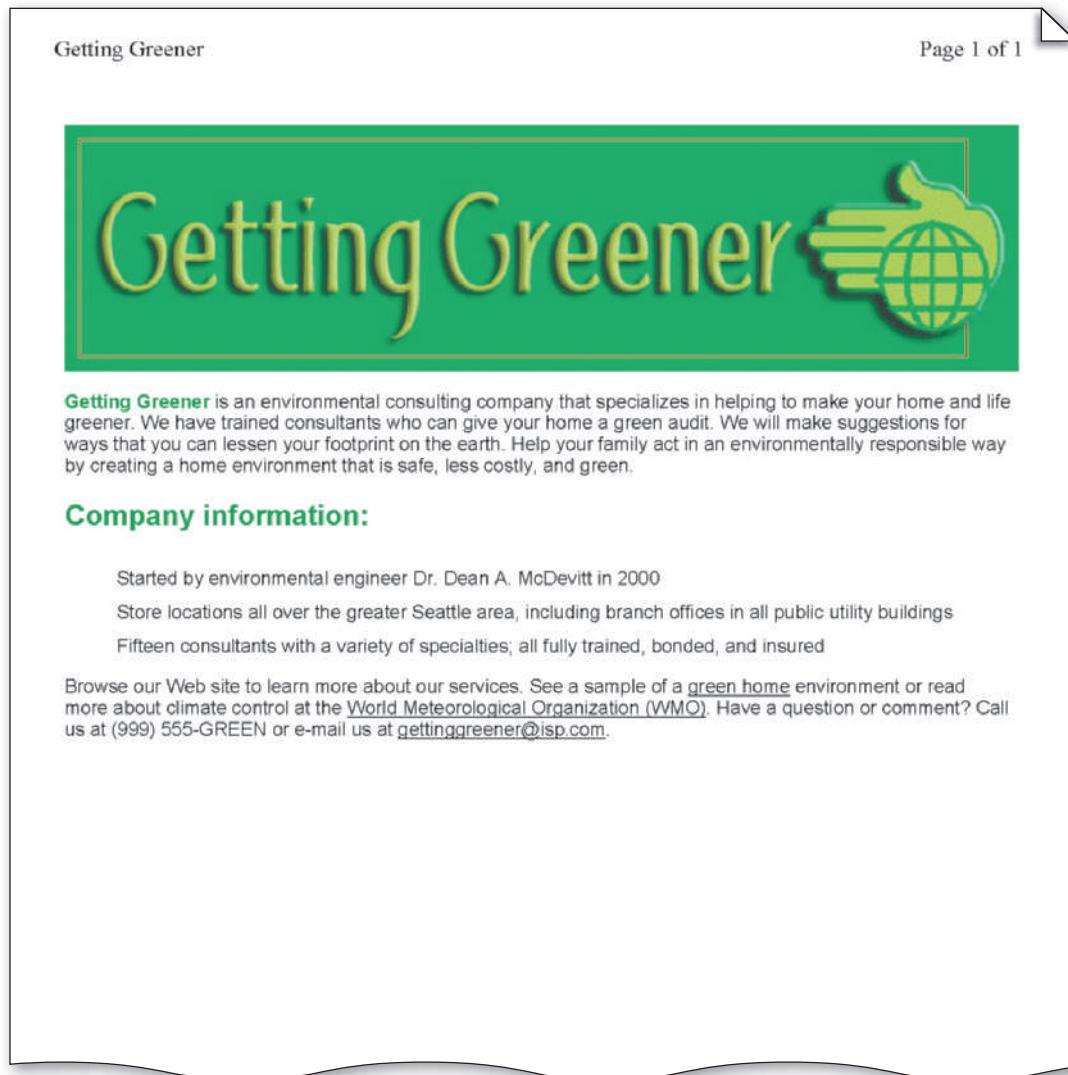
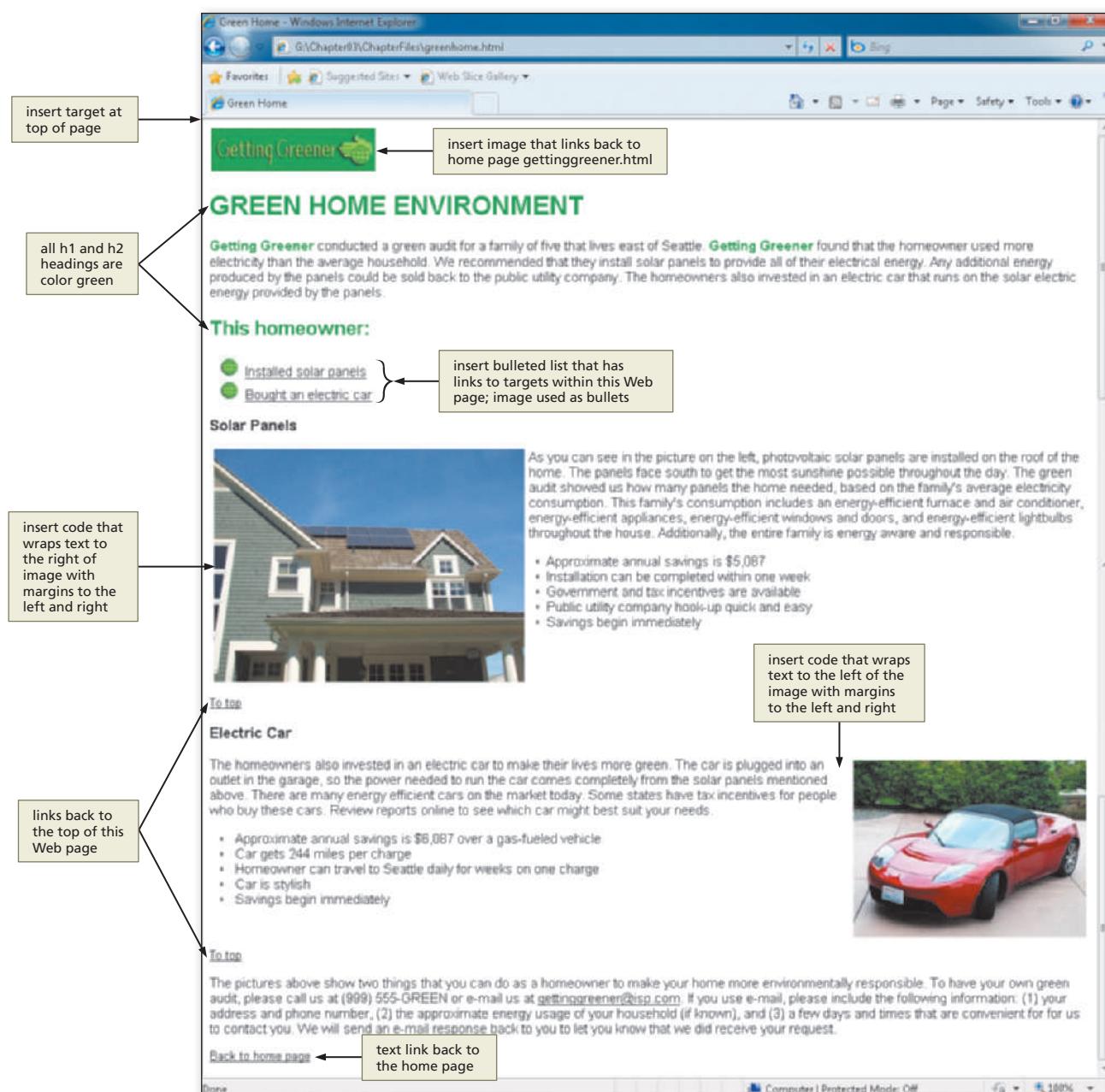


Figure 3–28

Editing the Second Web Page

With the home page complete, the next step is to enhance the Green Home Web page. For this part of the project, you will download an existing Web page file and edit the HTML code to create the Web page, as shown in Figure 3–29 on the next page. You will add two images and set text to wrap around the images. You also will add two additional types of links: links within the same Web page and an image link to a Web page in the same Web site.

As you have learned, the `<a>` tag used to create a link must specify the page, file, or location to which it links. In the case of a link within a Web page, the `<a>` tag specifies a **target**, or named location, in the same file. Before adding the links and targets in the Green Home Web page, you need to add an unordered (bulleted) list that uses the

**Figure 3–29****BTW**

Web Page Improvement
Web page development is an ongoing process. In Web page development, you create a Web page, view it in a browser, and then look for ways to improve the appearance of the page.

(earthbullet.jpg) image as the bullets. This list contains two items — Installed solar panels and Bought an electric car — and must be added to the page. The list items will serve as the links that are directed to the heading at the top of each major section of the Green Home Web page. When clicked, these links will move the Web page visitor to the targets, which are named solar and car, respectively.

Because the Web page is so long, it is a good design practice to provide users with a quick way to move back to the top of the Web page without scrolling back. For this purpose, the Web page includes two text links named To top. These links are located just above the Solar Panels and Electric Car headings. When clicked, any To top link takes the Web page visitor back to the top of the page.

To complete the Green Home Web page, you will create an image link, so users can click the back to home page link to return to the Getting Greener home page. There

is already a text link inserted at the bottom of the Web page that can be used to return to the home page. It is always important to provide a link back to the home page from subsequent Web pages. Your visitors should not have to use the Back button on the browser to return to the home page.

To Open an HTML File

The following steps illustrate how to open the greenhome.html file in Notepad++.

- 1 Click the Notepad++ button on the taskbar.
- 2 With a USB flash drive connected to one of the computer's USB ports, click File on the menu bar and then click Open.
- 3 Click Computer in the navigation pane to display a list of available drives.
- 4 If necessary, scroll until UDISK 2.0 (G:) is displayed in the list of available drives.
- 5 If necessary, navigate to the USB drive (G:). Click the Chapter03 folder, and then click the ChapterFiles folder in the list of available folders.
- 6 Click greenhome.html in the list of files.
- 7 Click the Open button in the Open dialog box to display the HTML code for the greenhome.html Web page, as shown in Figure 3–30.

Q&A If I open another file in Notepad++, will I lose the gettinggreener.html file?

The last saved version of gettinggreener.html will still be on the USB drive, even though another HTML file is open in Notepad++. Additionally, even after you open the new file in Notepad++ the other file (gettinggreener.html) remains open in another tab in Notepad++; that is one of the benefits of Notepad++; you can have more than one file open at the same time.

```

<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Green Home</title>
<style type="text/css">
<!--
body {font-family: Arial, Verdana, Garamond; font-size: 10pt}
h1, h2 {color: #009944}
a {color: black}
a:hover {background-color: #009944; color: white}
-->
</style>
</head>
<body>
<!--Insert image style classes here-->
-->
<!--Insert Top target here-->
<!--Insert image and link back to home here-->

```

Figure 3–30

**Plan
Ahead**

- **Determine what graphic images will be used and how to format them.** They say that a picture is worth a thousand words. In Web development, it sometimes makes your message clearer and more attractive if you use pictures. In the planning stage, you have to consider which pictures will help (and not hinder) your content. You also have to decide how to align the text relative to the pictures. Sometimes it makes sense to put the text above or below the picture. It might also be appropriate to wrap the text around the picture. You need to determine all of these specifics before you create the Web page.
- **Identify what links are needed on a long Web page.** When you have an especially long Web page (one in which the visitor has to use the PAGE DOWN key), you should provide links within the Web page for easier navigation. You need to decide where it makes sense to put page breaks. Often it is best to put a link to major topics within the Web page. Make sure that the Web page visitor can easily move to those areas by providing links toward the top of the Web page.
- **Use links back to the top of the page.** Another good technique for long Web pages is to allow visitors to link back to the top of the Web page easily from several places on the page. Providing links back to the top of a long Web page makes browsing more enjoyable.
- **Create a link back to the home page.** If possible, you should always provide a link from secondary Web pages back to the home page. Your visitors should not have to use the Back button on the browser to get back to the home page of the Web site. A common Web development practice is to use a company logo (often a smaller version) to navigate back to the home page. Again, the purpose of this image link as well as other links mentioned here is to make your Web site easy to navigate.

Working with Classes in Style Statements

Notice that the greenhome.html file contains the same embedded style sheet that you created in the gettinggreener.html file earlier in this chapter. For the second Web page, you will add one additional element (img) to the embedded style sheet. In order to utilize the image element as needed on the second Web page (greenhome.html), you need to understand the concept of classes as used with CSS. CSS classes give you more control over the style on a Web page.

Recall that a style statement is made up of a selector and a declaration. The part of the style statement that identifies the page elements is called the selector.

```
a {color: black}
```

The example above shows a section of the embedded style sheet used in the greenhome.html Web page. The selector in the example is the a (the anchor or link). The part of the style statement that identifies how the element(s) should appear is called the declaration. In this example, the declaration is everything between the curly brackets. This includes the property named color and the value named black.

There is another level of control that you can have over the styles that display on a Web page. For example, rather than having all paragraphs of text appear in the same style, you might want the style of the first paragraph on a page to be different from the other paragraphs of text. To gain more control for these purposes, you can define specific elements of an HTML file as a category, or **class**. You then can create a specific style for each class. Using classes in CSS thus allows you to apply styles to HTML tags selectively.

Using a class, for example, you could apply one style to a beginning paragraph and a different style to a closing paragraph on the same Web page.

Defining and using classes in CSS is a two-step process. First, any elements that belong to the class are marked by adding the tag:

```
class="classname"
```

where classname is the identifier or name of the class.

Any word can be used as a class name, as long as it does not contain spaces. In general, however, you should use descriptive names that illustrate the purpose of a class (for example, beginning, legallanguage, or copyrighttext), rather than names that describe the appearance of the class (for example, bluetext, largeritalic, or boldsmallarial). Using names that describe the purpose makes the code easier to read and more flexible. For this chapter, you will use the class names align-left and align-right in the img element. This immediately tells someone reviewing this code that the styles defined by those classes are used to align images either left or right.

After you have named the classes, you can use the class names in a selector and define a specific style for the class. For example, within the <style> tags in an embedded or external style sheet, you enter a style statement in the format:

```
p.beginning {color: red;
    font: 20pt}
```

where the p indicates that the class applies to a specific category of the paragraph tag and beginning is the class name. The tag and the class name are separated by a period. Together, the tag and the class name make up the selector for the style statement. The declaration then lists the property-value statements that should be applied to elements in the class.

For instance, if you want to display the beginning paragraph text in a 20-point red font, you would add a style statement like the one shown in the sample code in Figure 3–31a on the next page and then use the tag, <p class="beginning">, to apply the style defined by the declaration associated with the p.beginning selector. If the paragraph <p> tag is used without the class name, the paragraph appears in the default format or other format as defined by a style. To use this class name in an HTML tag, you would type:

```
<p class="beginning">
```

as the code.

In addition to the style for the beginning paragraphs, Figure 3–31a shows an example of HTML code with classes defined for and applied to the middle and end paragraphs. Figure 3–31b shows how the resulting Web page appears in the browser.

You can add as many classes to your Web pages as you need. This is a very useful Web development technique that allows flexibility and variety in a Web page. One drawback is that classes can be defined for use only in embedded or external style sheets. Because the purpose of using classes is to format a group of elements, not individual elements, classes do not work in inline styles.

The example below shows another section of the embedded style sheet in the greenhome.html Web page. In this example, you use a pseudo-class to have more control over the hover state in the anchor element.

```
a:hover {background: #00934a;
    color: white}
```

(a) HTML code with classes defined.

```

<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml/DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Figure 3-31</title>
<style type="text/css">
<!--
p.beginning {color: red;
font-size: 20pt}
p.middle {color: green;
font-size: 16pt;
font-weight: bold;
font-style: oblique}
p.end {color: navy;
font-size: 12pt;
font-weight: bolder;
font-style: normal;
font-variant: small-caps}
-->
</style>

```

class name middle
style for class named beginning applied to paragraph
style for class named middle applied to paragraph
style for class named end applied to paragraph
class name beginning
class name end
embedded style sheet used to define three paragraph classes
resulting styles as defined in embedded style sheet above; compare to HTML code above

(b) Resulting Web page.

In this first paragraph of text, I used the class name beginning to change the color of the text to red. The font weight is normal because I did not specify changes to that attribute. Finally, I changed the font size to 20 point.
In the second paragraph of text, I used the class name middle to make style changes. In this paragraph, the color of the text is green. I changed the font weight to bold and gave the paragraph a font style of oblique. This slants the font to the right.
In the last paragraph, I used the class name end to change style. The color of the text is changed to navy. The font weight in this paragraph is bolder, which makes it darker than the paragraph preceding it. I used a font variant of small caps for this text, in a normal style. The font size in this paragraph is 12 point, which is smaller than the paragraph before it.

Figure 3-31

BTW

Obtaining Images

The Web contains thousands of image files that can be downloaded for free and used for noncommercial purposes. Search for “free GIFs” or “free Web images” to find images. If you find a graphic you want to use, right-click the image, click Save Picture As on the shortcut menu, and then save the image to your computer. Many applications come with clip art that can be used on Web pages. Other types of digital images, such as images scanned by a scanner or pictures taken with a digital camera, also can be included on a Web page. You also can create images using a paint or image-editing program. Regardless of where you get the images, always follow copyright rules and regulations.

Adding an Image with Wrapped Text

As shown in Appendix A, the `` tag has many attributes, including attributes to specify height, width, and alternative text.

Alignment also is a key consideration when inserting an image. Alignment can give an image and the surrounding text completely different looks. Figure 3-32 shows two images, the first of which is left-aligned, which wraps any text to the right of the image. In this chapter, you use an embedded style sheet to align (float) the image to the left or right and wrap the text to the right or left of the positioned image. You also add some space (margins) around the image so that it is separated from the text. To accomplish these tasks, you use the float and margin properties.

The float property indicates in which direction (in this case left and right) to display (or float) an element being inserted on a Web page. **Floating** an element like an image allows the element to move to the side indicated in the float statement. As a result of that repositioning (floating), the other elements, like text, are moved up and allowed to wrap next to the floated element. When you first open the `greenhome.html` file in Internet Explorer (Figure 3-27 on page HTML 126), both images are left-aligned. Notice that the text is aligned beneath each image, leaving a lot of white space to the right of the images. The text does not surround the images in Figure 3-27, as it does in Figure 3-29 on page HTML 128. You achieve this text wrap by using the float property. In addition to floating the element, you should also provide some space around the image. The margin-left and

margin-right properties indicate how many pixels of space to put around each element. In this case, you will have five pixels of space around the right and left of each image. Figure 3–32 shows examples of images with margin spacing.

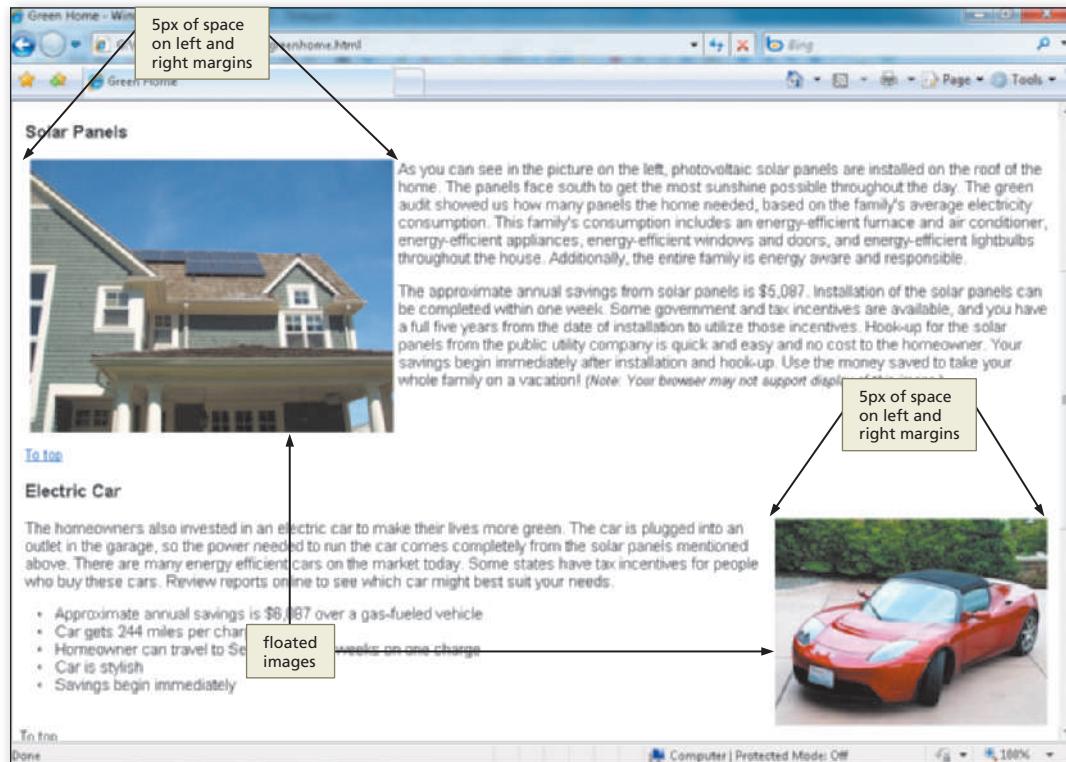


Figure 3–32 Left- and right-aligned images.

There are several ways to align text around images using styles. You can do this with an inline style (HTML code shown below) or with an embedded style sheet. The format of the HTML code to add the left- and right-aligned images with an inline style is:

```


```

where the float property tells the browser on which side to float the image element, and the margin properties tell the browser how much space (5 pixels) to add around the image. Using an inline style is a perfectly acceptable way to float an image element, wrap the text, and add margins of space around the image, but there may be a more efficient way to do it.

If you have numerous images to float on the page, it is better to use classes in an embedded style sheet than to use inline styles.

This project uses an embedded style sheet with a two-step approach. First, you will insert the class names align-left and align-right into the image `` tags that will use the classes. Then, you add those class names (align-left and align-right) to an image `` element in the embedded style sheet. To use this two-step approach, first you add the HTML code for the left-align and right-align class names in the `` tag itself within the Web page content:

```


```

which aligns the first image to the left, and the second image to the right.

Second, you have to insert the `img` (image) property in the embedded style sheet and add those two class names (`align-left` and `align-right`) where you define the style that you will use for images. This involves the following code that is inserted into the embedded style sheet:

```
img.align-left {float: left;
margin-left: 5px;
margin-right: 5px}
img.align-right {float: right;
margin-left: 5px;
margin-right: 5px}
```

where `img` is the property element and `align-right` and `align-left` are the class names. The class names `align-right` and `align-left` are arbitrary; you could name them anything. When naming classes, use names that make sense. Notice that the class names are separated from the element `img` with a period.

Another way to control space around images is to use the paragraph `<p>` tag. Remember that a paragraph tag inserts a blank line above the next object (text or image) after the paragraph tag. Figure 3–33a shows an example of using a `<p>` tag before inserting the Solar image, whereas Figure 3–33b shows an example of not using a `<p>` tag before the `` tag. In this project, we will use the paragraph tag before the `` tag to give more space between the image and the heading.

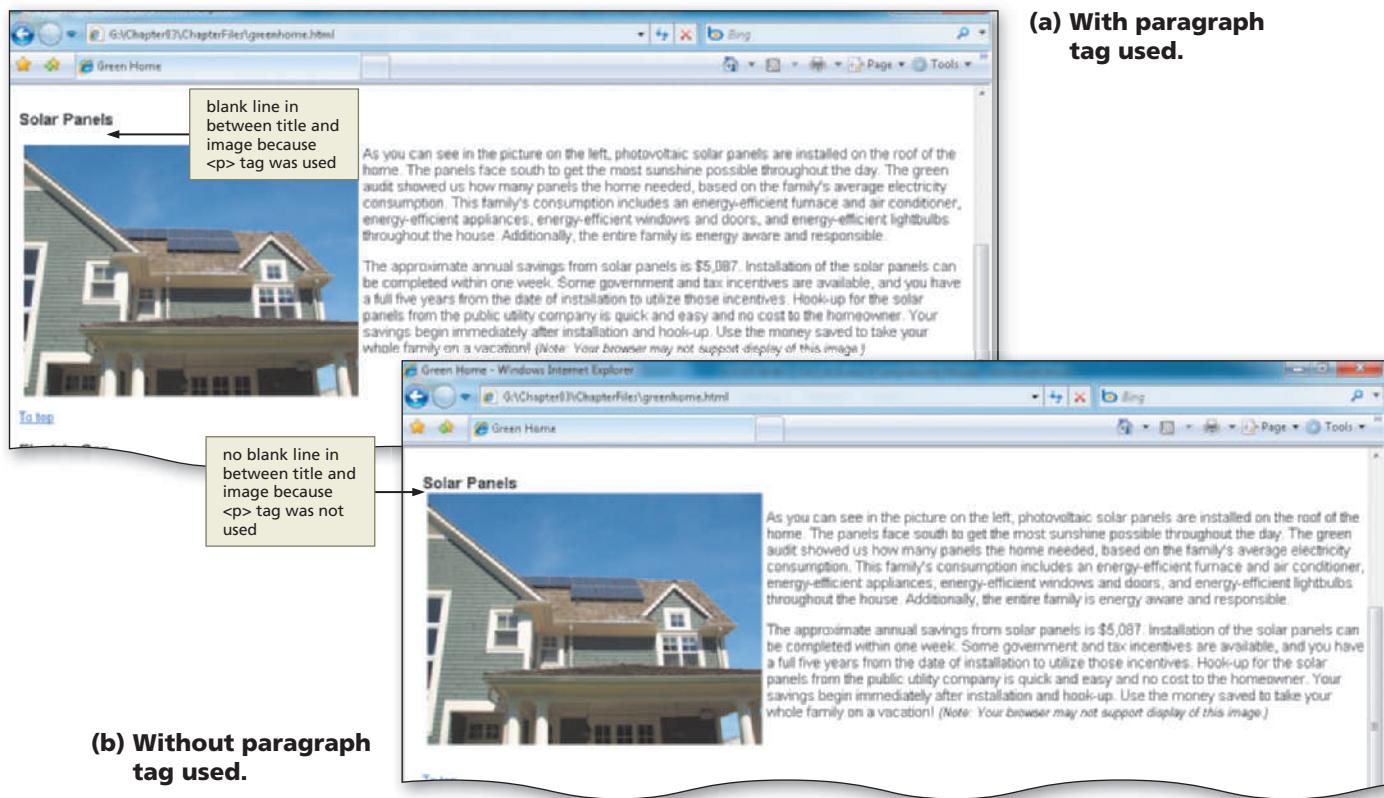


Figure 3-33

Using Thumbnail Images

Many Web developers use thumbnail images to improve page loading time. A **thumbnail image** is a smaller version of the image itself. The thumbnail is used as a link that, when clicked, will load the full-sized image. Figure 3–34a shows an example of a thumbnail image. When the image is clicked, the browser loads the full-sized image (Figure 3–34b). Loading images can take a long time, depending on the size and the complexity of the image. Using a thumbnail image gives a visitor the opportunity to decide whether to view the full-sized image.

To create a thumbnail version of an image, the image can be resized to a smaller size in a paint or image-editing program and then saved with a different file name. The thumbnail image then is added to a Web page as an image link to the larger version of the image. The HTML code to add a thumbnail image that links to a larger image takes the form:

```
<a href="largeimage.gif"></a>
```

where largeimage.gif is the name of the full-sized image and thumbnail.gif is the name of the smaller version of the image. In the case of this simple thumbnail example, a visitor clicks the thumbnail image to view the larger image, but there is no “return” button or link on the full-sized image Web page for the user to return to the original Web page. In this case, the visitor would have to use the Back button on the browser’s Standard toolbar to return to the original Web page displaying the thumbnail image. For most Web development projects, however, you always want to provide a link for the visitors and not force them to use the Back button.

BTW

Thumbnail Size

The size of a thumbnail varies depending on the clarity of the image and its purpose. Generally a 100–150 pixel height and 100–150 pixel width is an appropriate size. If you need the visitor to see more of the image even in the small size, then a larger thumbnail is fine. If a thumbnail is easily distinguishable at a very small size (100 x 100), then that size is appropriate. Also, you should never resize an image using the height and width attributes in the HTML code. You should resize the image in an editing program and save it with a new file name.

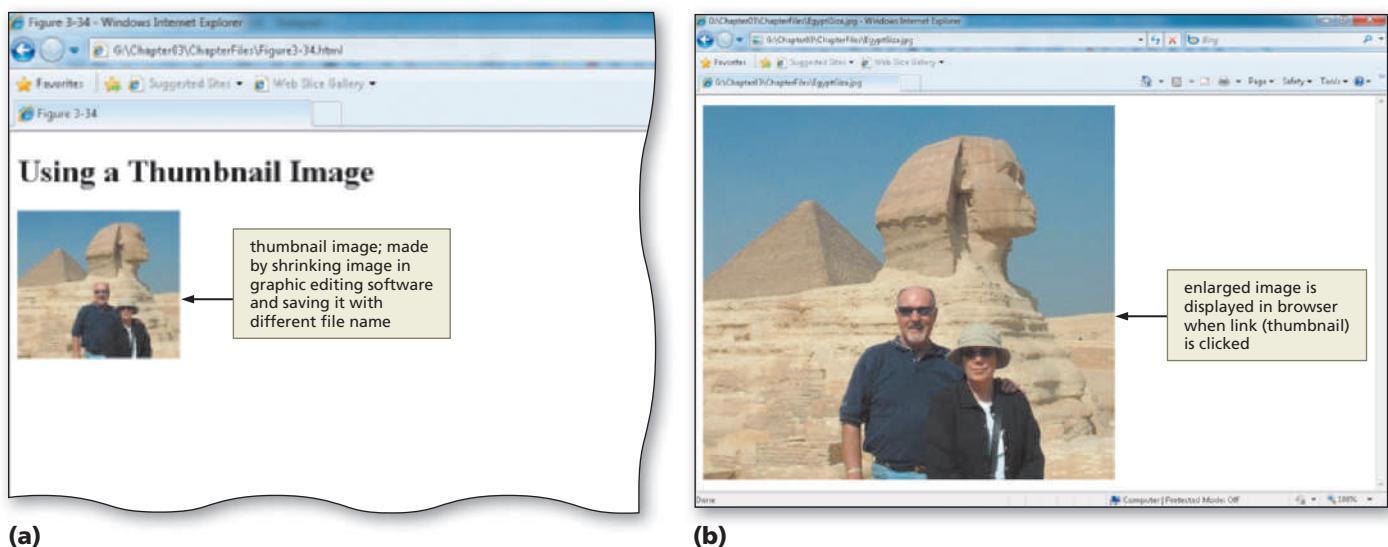


Figure 3–34

Wrap Text Around Images Using CSS Classes

Now you will use an embedded style sheet to wrap the text around the two images on the greenhome.html Web page. Remember that an embedded style sheet takes effect only for the Web page into which it is embedded. The embedded style sheet is placed within the `<head>` `</head>` container at the top of the Web page file.

The following steps show you how to insert left- and right-aligned images with wrapped text by adding class names to the img tags and then defining the classes within the embedded style sheets. Table 3–10 shows the code you'll need to define the classes in Step 3.

1

- With the greenhome.html file displayed in Notepad++, click immediately to the left of the s in src on line 44 to begin adding the class name to the first tag.
 - Type class="align-left" and press the SPACEBAR so that there is a space between what you just typed and src.
 - Click immediately to the left of the s in src on line 62 to begin adding the class name to the second tag.
 - Type class="align-right" and press the

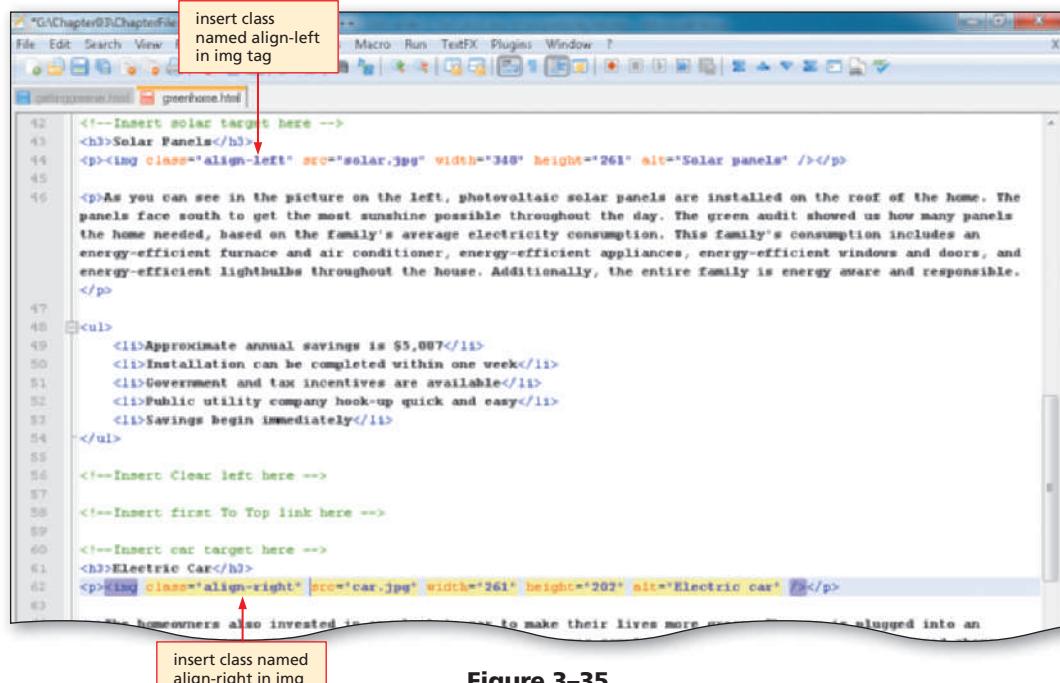


Figure 3-35

2

- Highlight the line <! Insert image style classes here --> on line 22, as shown in Figure 3–36, to begin adding image classes.

Q&A Do I have to press the **DELETE** key to delete the text that I highlighted in Step 2?

No, you do not have to press the **DELETE** key to delete the text on line 22. As long as the text is highlighted, the text is au

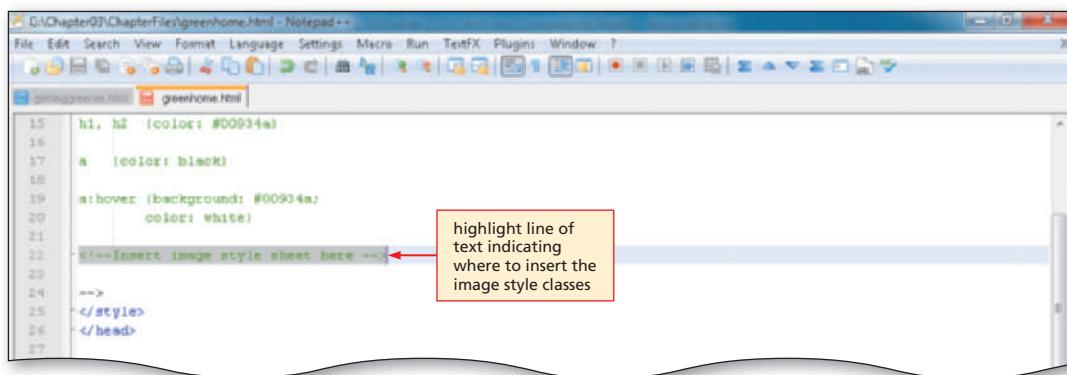


Figure 3–36

Table 3–10 CSS Code to Insert Class Definitions in Embedded Style Sheets

Line	CSS Selectors and Declarations
22	img.align-right {float: right;
23	margin-left: 5px;
24	margin-right: 5px}
25	
26	img.align-left {float: left;
27	margin-left: 5px;
28	margin-right: 5px}

3

- Type the HTML code in Table 3–10 but do not press the ENTER key at the end of line 28. This HTML code inserts the embedded style sheets that will be used by the align-left and align-right class names inserted in the previous step. This will align the images left or right on the Web page, with text wrapped to the right or left and with five pixels of horizontal space around the image (Figure 3–37).

Q&A

Why are we using the float property in these style sheets?

The float property allows you to position elements (in this case photographs) on the Web page.

Q&A

Why do we need margin-left and margin-right properties?

These properties provide five pixels (in this case) of space around the left and right sides of the image that is being inserted. If those are not entered, then the text aligns itself right next to the image. Allowing a bit of space between these two elements makes the image and text neater.

The screenshot shows the Notepad++ editor with the file 'greenhome.html' open. The code includes embedded CSS for styling headings, links, and images. Two specific CSS rules are highlighted with callouts:

- A callout points to the class definition for 'align-right': `img.align-right {float: right; margin-left: 5px; margin-right: 5px;}`. The callout text states: "insert image style with class name align-right that floats image to the right and adds 5px of space in left and right margins".
- A second callout points to the class definition for 'align-left': `img.align-left {float: left; margin-left: 5px; margin-right: 5px;}`. The callout text states: "insert image style with class name align-left that floats image to the left and adds 5px of space in left and right margins".

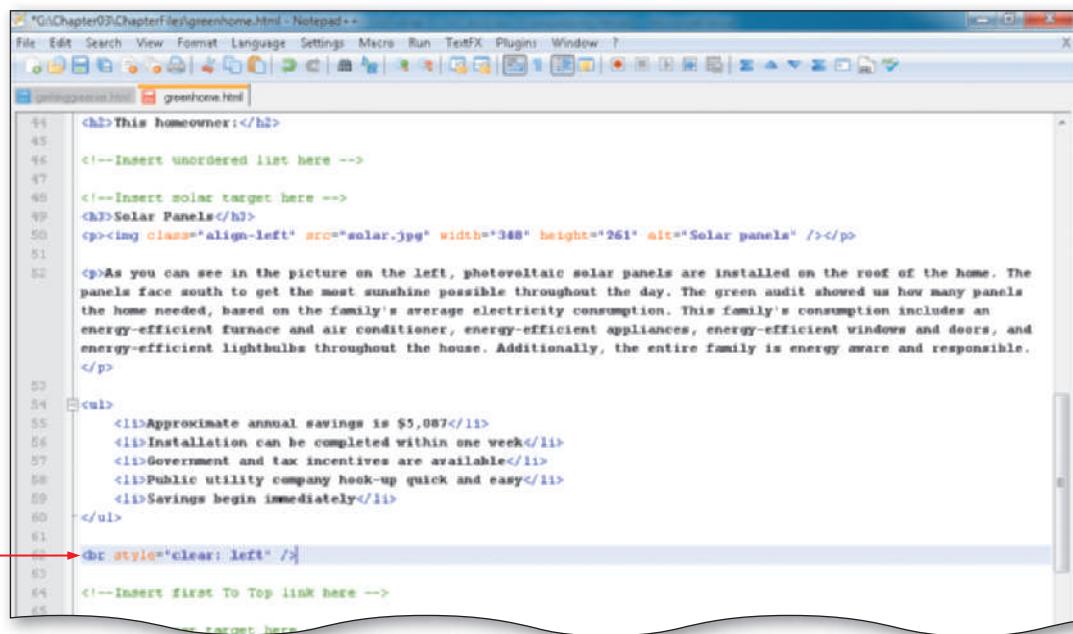
Figure 3–37

To Clear Text Wrapping

After specifying an image alignment and defining how text wraps, you must enter a break (`
`) tag to stop the text wrapping. You use the `<br style="clear: both" />`, `<br style="clear: left" />`, or `<br style="clear: right" />` tags to show where the text should stop wrapping. The following steps show how to enter code to clear the text wrapping.

1

- Highlight the line `<br style="clear: left" />` as the tag (Figure 3-38).



```

<h2>This homeowner:</h2>
<!--Insert unordered list here --&gt;
&lt;h3&gt;Solar Panels&lt;/h3&gt;
&lt;p&gt;&lt;img class="align-left" src="solar.jpg" width="388" height="261" alt="Solar panels" /&gt;&lt;/p&gt;
&lt;p&gt;As you can see in the picture on the left, photovoltaic solar panels are installed on the roof of the home. The panels face south to get the most sunshine possible throughout the day. The green audit showed us how many panels the home needed, based on the family's average electricity consumption. This family's consumption includes an energy-efficient furnace and air conditioner, energy-efficient appliances, energy-efficient windows and doors, and energy-efficient lightbulbs throughout the house. Additionally, the entire family is energy aware and responsible.&lt;/p&gt;
&lt;ul&gt;
    &lt;li&gt;Approximate annual savings is $5,087&lt;/li&gt;
    &lt;li&gt;Installation can be completed within one week&lt;/li&gt;
    &lt;li&gt;Government and tax incentives are available&lt;/li&gt;
    &lt;li&gt;Public utility company hook-up quick and easy&lt;/li&gt;
    &lt;li&gt;Savings begin immediately&lt;/li&gt;
&lt;/ul&gt;
&lt;br style="clear: left;" /&gt;
!--Insert first To Top link here --&gt;
</pre>

```

Figure 3-38

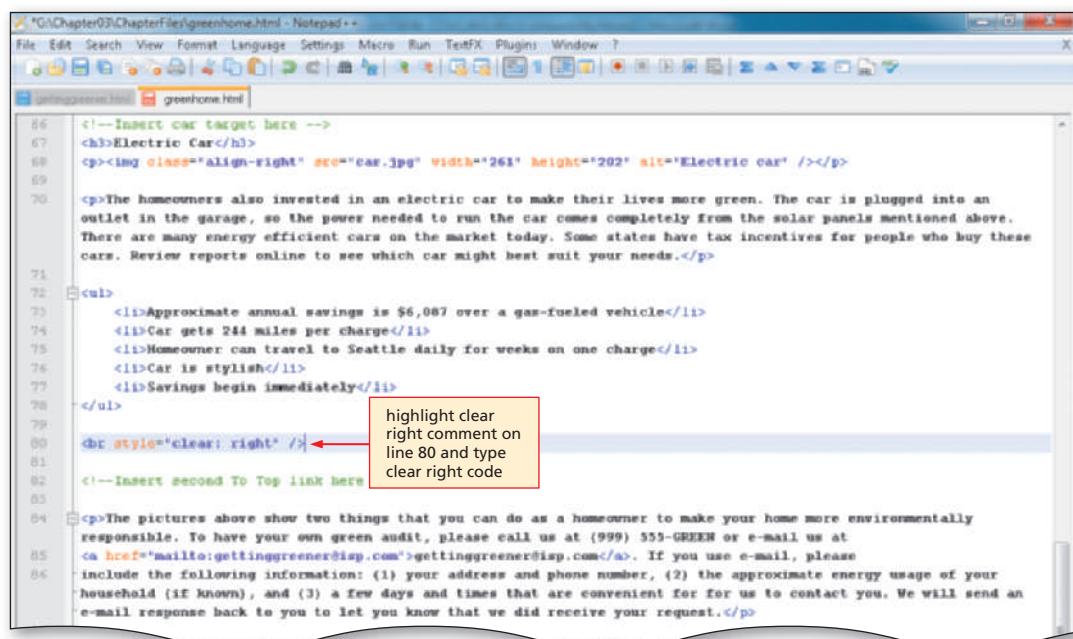
2

- Highlight the line `<br style="clear: right" />` as the tag to clear the text wrapping for both left- and right-aligned images, as displayed in Figure 3-39.

Q&A

What happens if you do not use the `<br style="clear: direction" />` tag?

Your text following the wrapped image will not be displayed as you intended. The following text will continue to wrap beyond the end of the text and image combination.



```

<!--Insert car target here --&gt;
&lt;h3&gt;Electric Car&lt;/h3&gt;
&lt;p&gt;&lt;img class="align-right" src="car.jpg" width="261" height="202" alt="Electric car" /&gt;&lt;/p&gt;
&lt;p&gt;The homeowners also invested in an electric car to make their lives more green. The car is plugged into an outlet in the garage, so the power needed to run the car comes completely from the solar panels mentioned above. There are many energy efficient cars on the market today. Some states have tax incentives for people who buy these cars. Review reports online to see which car might best suit your needs.&lt;/p&gt;
&lt;ul&gt;
    &lt;li&gt;Approximate annual savings is $6,087 over a gas-fueled vehicle&lt;/li&gt;
    &lt;li&gt;Car gets 244 miles per charge&lt;/li&gt;
    &lt;li&gt;Homeowner can travel to Seattle daily for weeks on one charge&lt;/li&gt;
    &lt;li&gt;Car is stylish&lt;/li&gt;
    &lt;li&gt;Savings begin immediately&lt;/li&gt;
&lt;/ul&gt;
&lt;br style="clear: right;" /&gt;
!--Insert second To Top link here --&gt;
</pre>

```

Figure 3-39

Q&A

Is there one tag to clear all alignments?

Yes. The `<br style="clear: both" />` tag clears all text alignments.

Adding Links Within a Web Page

The final links to be added in this project are links within the Green Home Web page. Because the Green Home Web page is quite long, it would be easier for visitors to have a menu or list at the top of the Web page that facilitates immediate movement to another section. Figure 3–40 shows how clicking the text link Installed solar panels in the bulleted list near the top of the page links to the Solar Panels section in another part of the Web page. When the mouse pointer is moved over the words Installed solar panels and is clicked, the browser repositions, or links, the page to the target named solar. Notice when the mouse hovers over the link, the link changes to green background and white text. That is because of the a:hover styles that are embedded on this Web page.

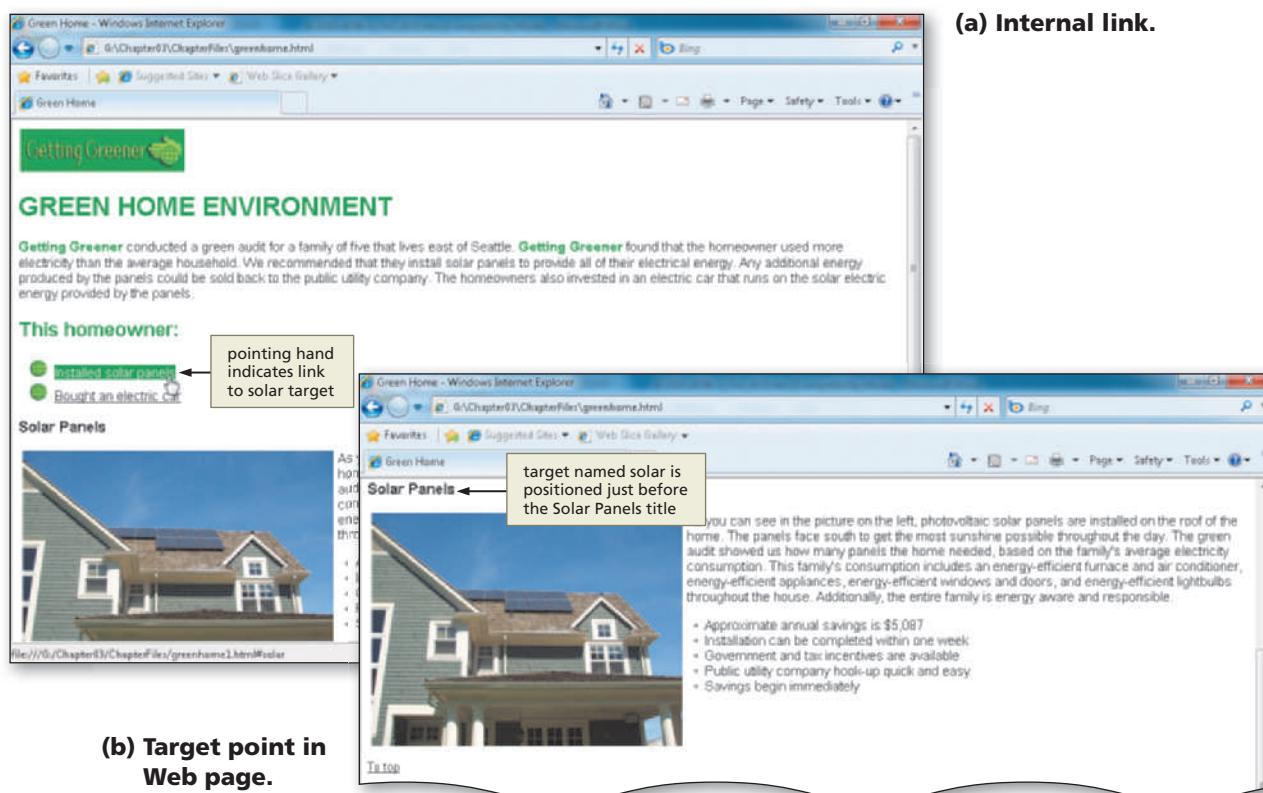


Figure 3–40

To create links within the same Web page, the targets for the links first must be created. Link targets are created using the `<a>` tag with the name attribute, using the form:

```
<a name="targetname"></a>
```

where targetname is a unique name for a link target within that Web page. Notice that the tag uses the name attribute, rather than the href attribute, and that no text is included between the start `<a>` and end `` tag, because the target is not intended to appear on the Web page as a clickable link. Instead, the link target is intended to mark a specific area of the Web page, to which a link can be directed.

Links to link targets are created using the `<a>` tag with the href attribute, using the form:

```
<a href="#targetname">
```

where targetname is the name of a link target in that Web page. Notice that the tag uses the href attribute, followed by the pound sign (#) and the target name enclosed in quotation marks.

To Set Link Targets

The next step is to set link targets to the Solar Panels and Electric Car sections of the Web page. The following steps show how to set the two link targets in the Green Home Web page.

1

- Highlight the line `<!-- Insert solar target here -->` on line 48.
- Type `` to create a link target named solar (Figure 3–41).

```

<!-- Insert solar target here -->
<a name="solar"></a>

```

Figure 3–41

2

- Highlight the line `<!--Insert car target here -->` on line 66.
- Type `` to create a link target named car (Figure 3–42).

Q&A

There is nothing between the start anchor and end anchor tags for these targets. Will they work?

These targets are just placeholders, so they do not need any words or phrases; they only need a target name, as shown in the anchor tag.

```

<a name="car"></a>

```

Figure 3–42

To Add Links to Link Targets Within a Web Page

The next step is to add link targets using the code shown in Table 3–11.

Table 3–11 HTML Code to Insert Bulleted List with Links to Link Targets

Line	HTML Tag and Text
46	<ul style="list-style-image: url(earthbullet.jpg)">
47	Installed solar panels
48	Bought an electric car
49	

The following step shows how to add the code to create an unordered (bulleted) list and then to use the list items as links to link targets within the Web page.

1

- Highlight the line
`<!--Insert unordered list here -->` on line 46.

- Type the HTML code in Table 3–11 but do not press the ENTER key at the end of line 49.

This HTML code inserts the bulleted list that provides links to the two targets (solar and car) inserted above.

Notice that you use the same earthbullet.jpg image that you used on the home page as the bullet (Figure 3–43). This helps with consistency across the Web site.

The screenshot shows a Microsoft Word document with the following HTML code:

```

34 <body>
35
36 <!--Insert Top target here -->
37
38 <!--Insert image and link back to home here -->
39
40 <h1>GREEN HOME ENVIRONMENT</h1>
41
42 <p><span style="color: #00933a; font-weight: bold">Getting Greener</span> conducted a green audit for a family of five that lives east of Seattle. <span style="color: #00933a; font-weight: bold">Getting Greener</span> found that the homeowner used more electricity than the average household. We recommended that they install solar panels to provide all of their electrical energy. Any additional power would be sold back to the utility company. The homeowners also invested in energy efficient windows and insulation. The panels could be sold back to the utility company if the homeowner moves or sells the house. The panels run on solar electric energy provided by the panels.</p>
43
44 <h2>This homeowner:</h2>
45
46 <ul style="list-style-image: url(earthbullet.jpg)">
47 <li><a href="#solar">Installed solar panels</a></li>
48 <li><a href="#car">Bought an electric car</a></li>
49 </ul>
50
51 <a name="#solar"></a>
52 <h3>Solar Panels</h3>
53 <p></p>
54

```

Annotations in the screenshot:

- A callout box points to the line 46 code: "use earthbullet.jpg image as bullet, just like on home page".
- A callout box points to the line 47 code: "inserted bulleted list to targets solar and car starting on line 46".
- A callout box points to the line 47 code: "link to href target named solar from this link".
- A callout box points to the line 47 code: "target named solar".

Figure 3–43

Q&A

Do I have to use a bulleted list for the links?

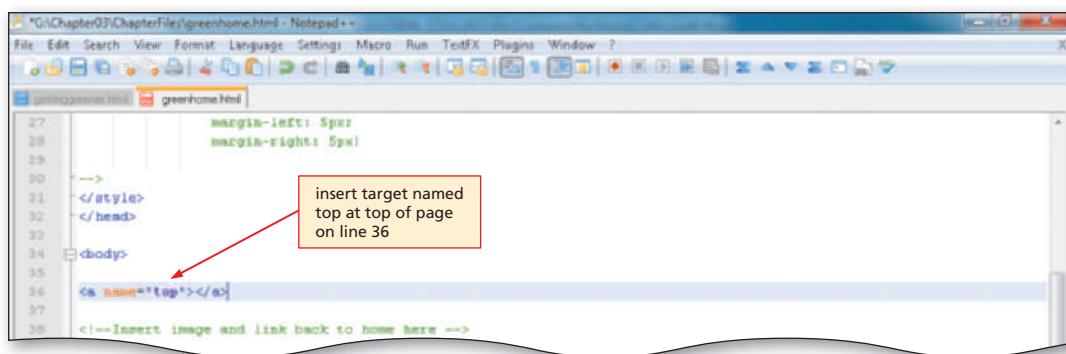
No, you can use any text for the links to the targets created in the step above. The bulleted list makes the links easy to use and keeps the links in one area of the Web page.

To Add Links to a Target at the Top of the Page

In this step, you add two To top links to provide a quick way to move back to the top of the Web page. To make these links, you first set the target at the top of the page, and then create the links to that target. You will also use an inline style to make the link text smaller than the regular font size. The following steps illustrate how to add links to a target at the top of the page.

1

- Highlight the line
`<!--Insert Top target here --> on line 36.
- Type `` as the tag that will create a target at the top of the Web page named top (Figure 3-44).



The screenshot shows a Notepad++ window with the file "greenhome.html" open. Line 36 contains the code ``. A red callout box points to this line with the text "insert target named top at top of page on line 36".

Figure 3-44

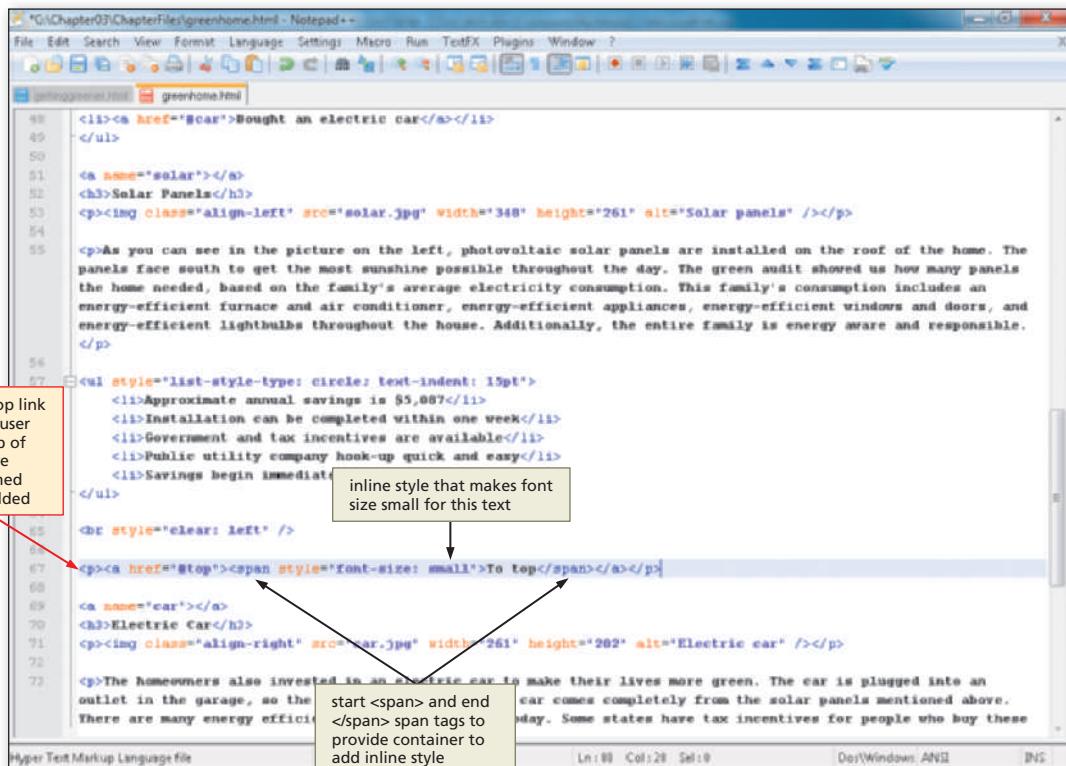
2

- Highlight the line
`<!--Insert first To top link here --> on line 67.
- Type `<p>To top</p>` as the tag (Figure 3-45).

Q&A

Why do you use a small size font for this link?

The link back to the top of the page should be subtle yet noticeable as distinguished from the other text on the page. Notice that the text link at the very bottom of the Web page to return to the home page is the same small size.



The screenshot shows a Notepad++ window with the file "greenhome.html" open. Line 67 contains the code `<p>To top</p>`. A red callout box points to this line with the text "insert To top link that takes user back to top of page where target named top was added". Another callout box points to the `style="font-size: small"` part with the text "inline style that makes font size small for this text". A third callout box points to the `` and `` tags with the text "start and end span tags to provide container to add inline style".

Figure 3-45

To Copy and Paste HTML Code

The copy and paste feature can be very useful for entering the same code in different places. The following step shows how to copy and paste the link code to three other lines in the HTML code.

1

- Highlight the HTML code `<p>To top` on line 67.
- Click Edit on the menu bar and then click Copy.
- Highlight the line `<!--Insert second To top link here -->` on line 85 to position the pointer.
- Click Edit on the menu bar and then click Paste to paste the HTML code that you copied into line 85 (Figure 3-46).

```

70 <a href="#top"><span style="font-size: small">To top</span></a>
71 <h3>Electric car</h3>
72 <p></p>
73 <p>The homeowners also invested in an electric car to make their lives more green. The car is plugged into an outlet in the garage, so the power needed to run the car comes completely from the solar panels mentioned above. There are many energy efficient cars on the market today. Some states have tax incentives for people who buy these cars. Review reports online to see which car might best suit your needs.</p>
74 <ul style="list-style-type: none; text-indent: -0pt">
75   <li>Approximate annual savings is $6,887 over a gas-fueled vehicle</li>
76   <li>Car gets 244 miles per charge</li>
77   <li>Homeowner can travel to Seattle daily for weeks on one charge</li>
78   <li>Car is stylish</li>
79   <li>Savings begin immediately</li>
80 </ul>
81 <br style="clear: right" />
82 <p><a href="#top"><span style="font-size: small">To top</span></a></p>
83 <!--Insert second To Top link to take user to target named top-->
84 <p>The pictures above show two things that you can do as a homeowner to make your home more environmentally responsible. To have your own green audit, please call us at (999) 555-GREEN or e-mail us at <a href="mailto:gettinggreener@sp.com">gettinggreener@sp.com</a>. If you use e-mail, please include the following information: (1) your address and phone number, (2) the approximate energy usage of your household (if known), and (3) a few days and times that are convenient for us to contact you. We will send an e-mail response back to you to let you know that we did receive your request.</p>

```

Figure 3-46

To Add an Image Link to a Web Page

The last step is to add an image link from the Green Home Web page back to the Getting Greener home page. The style for links that you set with an embedded style sheet earlier says that link background color is green and text is white. For an image link, however, this would not be appropriate. To override the embedded style sheet for just this one instance, you will use an inline style to set a background-color that is transparent with text-decoration of none (no underline). Remember that according to the style precedence rules, an inline style takes precedence over an embedded style sheet. These two styles (background-color: transparent; text-decoration: none) ensure that no border appears around the image and there is no line under the link image. Table 3-12 shows the code used to insert the image link.

Table 3-12 HTML Code to Insert Image Link to Home

Line	HTML Tag and Text
38	<code><div></code>
39	<code></code>
40	<code></code>
41	<code></div></code>

The following step shows how to create an image link at the top of the Green Home Web page.

1

- Highlight the line
 <!--Insert image and
 link back to home
 here --> on line 38.
 - Type the HTML code
 in Table 3–12 but
 do not press the
 ENTER key at the end
 of line 41.

This HTML code inserts a link back to home from the image earthsm.jpg.

This image is a smaller version of the earthlogo.jpg image that you used on the home page (Figure 3-47).

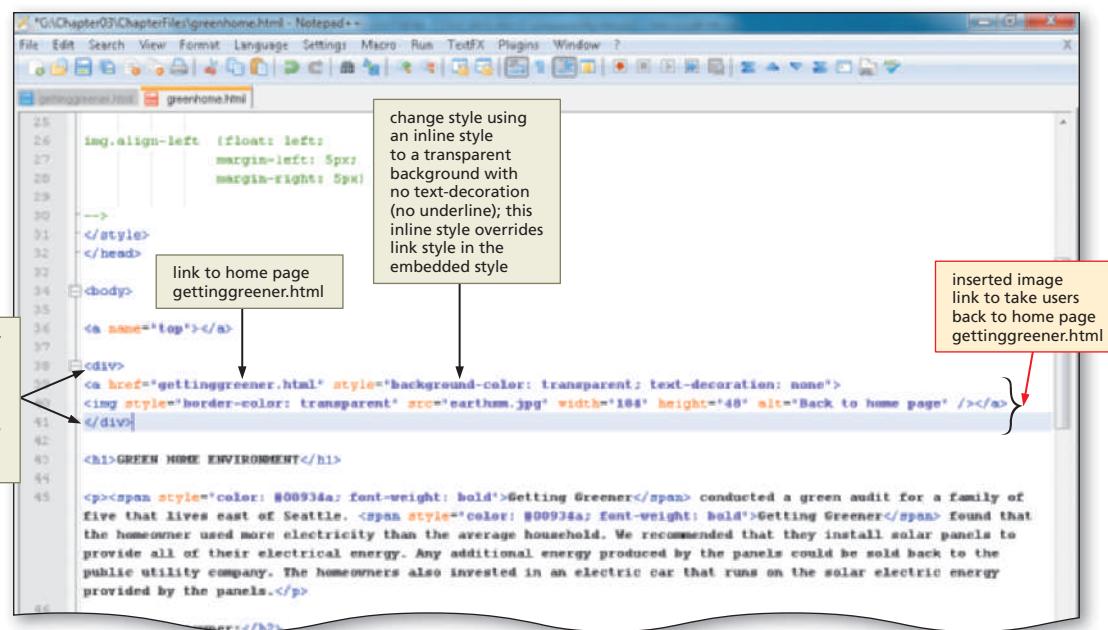


Figure 3-47

To Save, Validate, and Print the HTML File

With the HTML code for the Green Home Web page complete, the HTML file should be saved, the Web page should be validated at w3.org, and a copy of the file should be printed as a reference.

- 1 If necessary, activate the Notepad++ window.
 - 2 Click File on the menu bar, and then Save on the File menu to save the HTML file as greenhome.html.
 - 3 Open a new browser window and go to validator.w3.org.
 - 4 Click the Validate by File Upload tab, browse to the greenhome.html Web page, and then click Open.
 - 5 Click the Check button to determine if the Web page is valid. If the file is not valid, make corrections, re-save, and revalidate.
 - 6 Click the Notepad++ button on the taskbar to display the greenhome.html code. Click File on the menu bar, click Print on the File menu, and then click the Print button in the Print dialog box to print the HTML code.

To View and Test a Web Page

With the HTML code validated and saved, you should view the Web page and test the links.

- 1** Click the Internet Explorer button on the taskbar to view the page in your browser.
- 2** Click the Refresh button on the Standard toolbar to display the changes made to the Web page, which should now look like Figure 3–1b on page HTML 91.
- 3** Verify that all internal links work correctly by clicking the two links in the bulleted list at the top of the Web page. Also make sure to check the two To top links. Finally, verify that the image link to the home page works.

Q&A How can I tell if internal links are working when the link and target are displayed in the same browser window?

To see movement to a link, you might need to restore down and resize the browser window so that the target is not visible, then click the link.

To Print a Web Page

- 1** Click the Print icon on the Command bar to print the Web page (Figure 3–48).

Green Home Page 1 of 2

Getting Greener conducted a green audit for a family of five that lives east of Seattle. **Getting Greener** found that the homeowner used more electricity than the average household. We recommended that they install solar panels to provide all of their electrical energy. Any additional energy produced by the panels could be sold back to the public utility company. The homeowners also invested in an electric car that runs on the solar electric energy provided by the panels.

This homeowner:

- [Installed solar panels](#)
- [Bought an electric car](#)

Solar Panels

Government and tax incentives are available
Public utility company hook-up quick and easy
Savings begin immediately

[To top](#)

Electric Car

The homeowners also invested in an electric car to make into an outlet in the garage, so the power needed to run panels mentioned above. There are many energy efficient have tax incentives for people who buy these cars. Review suit your needs.

Green Home Page 2 of 2

Approximate annual savings is \$6,087 over a gas-fueled vehicle
Car gets 244 miles per charge
Homeowner can travel to Seattle daily for weeks on one charge
Car is stylish
Savings begin immediately



[To top](#)

The pictures above show two things that you can do as a homeowner to make your home more environmentally responsible. To have your own green audit, please call us at (999) 555-GREEN or e-mail us at gettinggreener@isp.com. If you use e-mail, please include the following information: (1) your address and phone number, (2) the approximate energy usage of your household (if known), and (3) a few days and times that are convenient for us to contact you. We will send an e-mail response back to you to let you know that we did receive your request.

[Back to home page](#)

Figure 3–48

Quick Reference

For a list of HTML tags and their associated attributes, see the HTML Quick Reference (Appendix A) at the back of this book, or visit the Quick Reference Web page for this book (scsite.com/HTML6e/qr). For a list of CSS properties and values, see Appendix D, or visit scsite.com/HTML6e/qr.

To Quit Notepad++ and a Browser

- 1** In Notepad++, click the File menu, then Close All.
- 2** Click the Close button on the Notepad++ title bar.
- 3** Click the Close button on all open browser windows.

Chapter Summary

In this chapter, you have learned how to develop a two-page Web site with links, images, and formatted text. You learned how to use inline and embedded style sheets and style classes to format elements in each Web page. The items listed below include all the new HTML and CSS skills you have learned in this chapter.

1. Add a Center-Aligned Banner Image Using an Inline Style (HTML 102)
2. Create an Unordered (Bulleted) List Using Images as Bullets (HTML 106)
3. Add a Text Link to Another Web Page Within the Same Web Site (HTML 108)
4. Add an E-Mail Link (HTML 109)
5. Add a Text Link to a Web Page in Another Web Site (HTML 111)
6. Add Embedded Style Statements (HTML 119)
7. Add an Inline Style for Color (HTML 121)
8. Test Links on a Web Page (HTML 125)
9. Wrap Text Around Images Using CSS Classes (HTML 136)
10. Clear Text Wrapping (HTML 138)
11. Set Link Targets (HTML 140)
12. Add Links to Link Targets Within a Web Page (HTML 141)
13. Add Links to a Target at the Top of the Page (HTML 142)
14. Copy and Paste HTML Code (HTML 143)
15. Add an Image Link to a Web Page (HTML 143)

Learn It Online

Test your knowledge of chapter content and key terms.

Instructions: To complete the Learn It Online exercises, start your browser, click the Address bar, and then enter the Web address scsite.com/html6e/learn. When the HTML Learn It Online page is displayed, click the link for the exercise you want to complete and read the instructions.

Chapter Reinforcement TF, MC, and SA

A series of true/false, multiple choice, and short answer questions that test your knowledge of the chapter content.

Flash Cards

An interactive learning environment where you identify chapter key terms associated with displayed definitions.

Practice Test

A series of multiple choice questions that test your knowledge of chapter content and key terms.

Who Wants To Be a Computer Genius?

An interactive game that challenges your knowledge of chapter content in the style of a television quiz show.

Wheel of Terms

An interactive game that challenges your knowledge of chapter key terms in the style of the television show, *Wheel of Fortune*.

Crossword Puzzle Challenge

A crossword puzzle that challenges your knowledge of key terms presented in the chapter.

Apply Your Knowledge

Reinforce the skills and apply the concepts you learned in this chapter.

Adding Text Formatting to a Web Page Using Inline Styles

Instructions: Start Notepad++. Open the file apply3-1.html from the Chapter03\Apply folder of the Data Files for Students. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files.

The apply3-1.html file is a partially completed HTML file that you will use for this exercise. Figure 3-49 shows the Apply Your Knowledge Web page as it should be displayed in a browser after the additional HTML tags and attributes are added.

Perform the following tasks:

1. Enter g:\Chapter03\Apply\apply3-1.html as the URL to view the Web page in your browser.
2. Examine the HTML file in Notepad++ and its appearance in the browser.
3. In Notepad++, change the HTML code to make the Web page look similar to the one shown in Figure 3-49.
4. Use an inline style to create a left-aligned h1 heading, Dorm Room Doctors, and make it blue.
5. Using the <div></div> container, right-align the image garbage.jpg (width=600, height=428). Give the image a right-and left-margins of 10 pixels. (Make sure to use the alt attribute.)
6. Make the first paragraph a large font. Color the words YES red in that paragraph. (*Hint:* Use the tag.)
7. The second paragraph should be normal font, but make the word sanity blue and italic.
8. In the third paragraph, make the phone number and e-mail address red and in a bolder font-weight.
9. Save the revised HTML file in the Chapter03\Apply folder using the file name apply3-1solution.html.

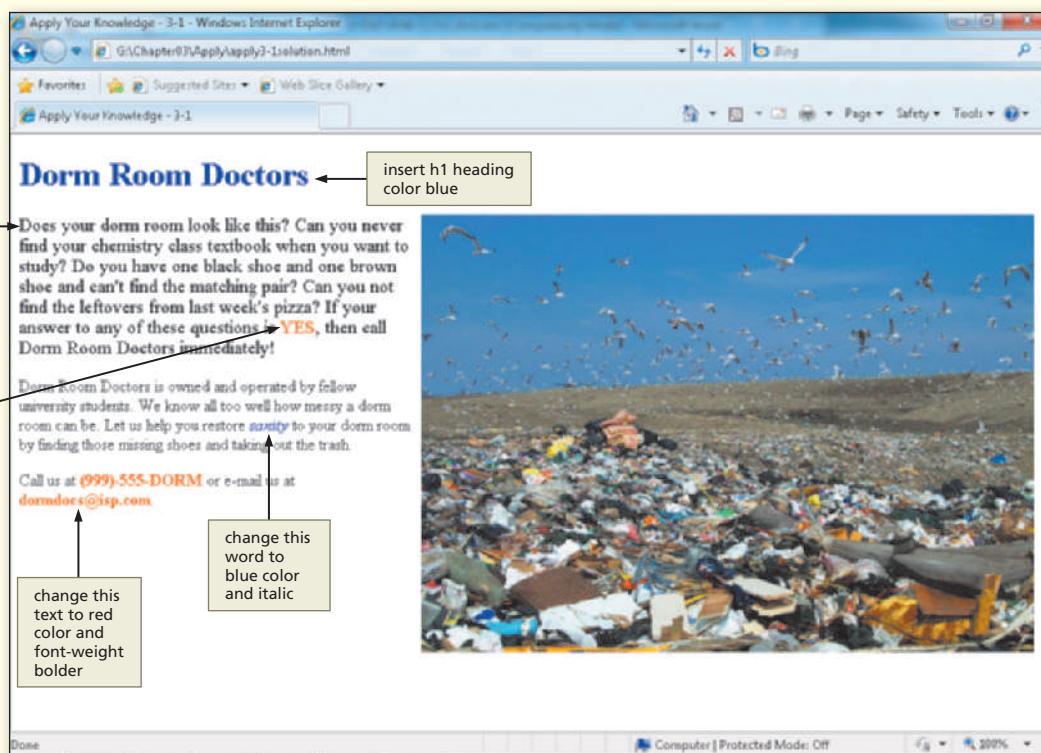


Figure 3-49

Continued >

Apply Your Knowledge *continued*

10. Validate your code.
11. Print the revised HTML file.
12. Enter g:\Chapter03\Apply\apply3-1solution.html as the URL to view the revised Web page in your browser.
13. Print the Web page.
14. Submit the revised HTML file and Web page in the format specified by your instructor.

Extend Your Knowledge

Extend the skills you learned in this chapter and experiment with new skills.

Creating Targets and Links

Instructions: Start Notepad++. Open the file extend3-1.html from the Chapter03\Extend folder of the Data Files for Students. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files. This sample HTML file contains all of the text for the Web page shown in Figure 3–50. You will add the necessary tags to make this Web page appear with left- and right-aligned images, text formatting, and links, as shown in Figure 3–50.

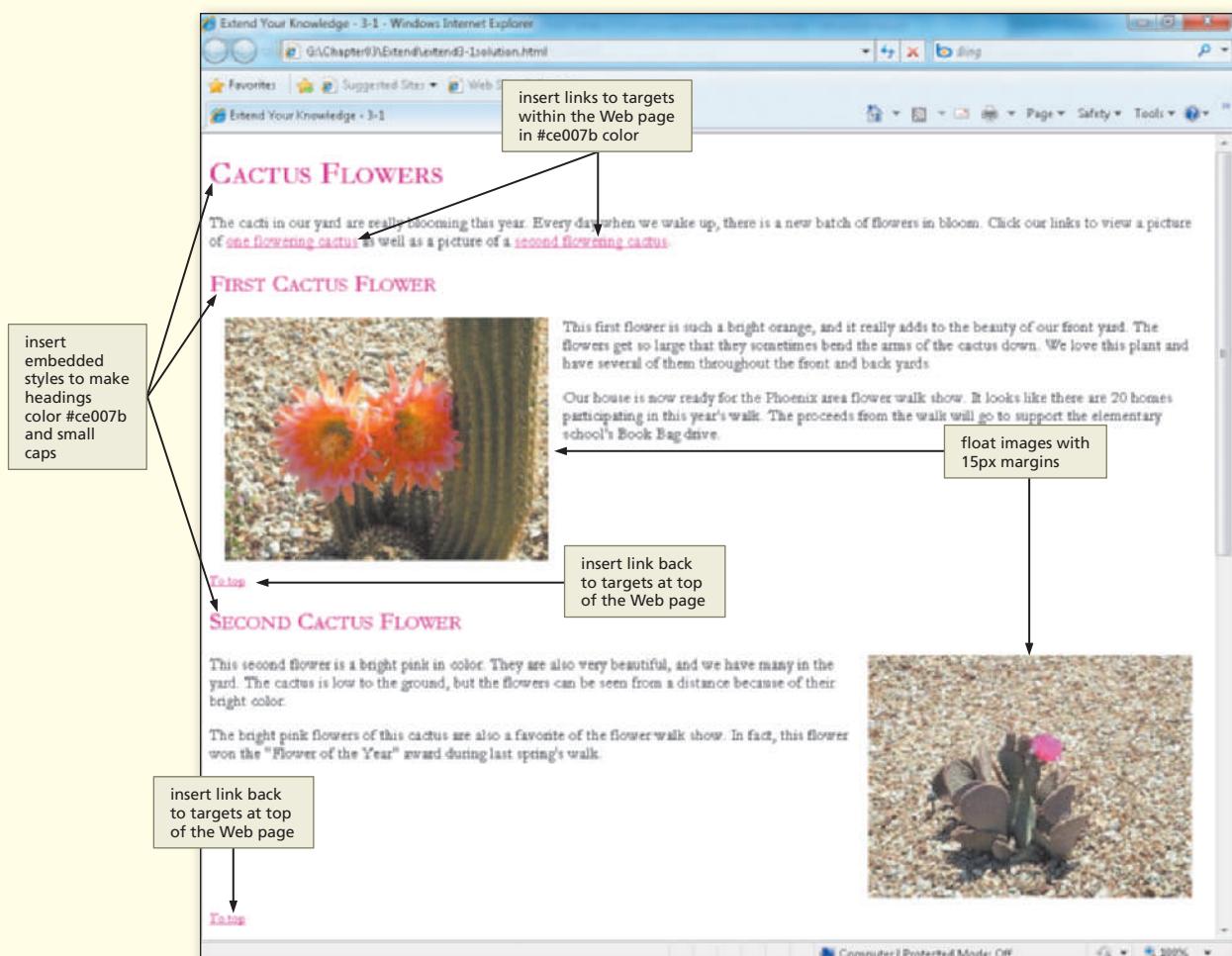


Figure 3–50

Perform the following tasks:

1. Insert the following embedded style sheets:

```
body {font-family: Garamond, Arial, Verdana;
      font-size: 12pt}
h1, h2 {color: #ce007b;
         font-variant: small-caps}
img {hspace: 20}
.align-right {float: right;
              margin-left: 15px;
              margin-right: 15px}
.align-left {float: left;
              margin-left: 15px;
              margin-right: 15px}
a {text-decoration: underline;
   color: #ce007b}
a:hover {background: #ce007b;
          color: white}
```

2. Make sure to use inline styles for all other styles. Add code to align the second picture on the right, also with margins of 15px. (*Hint:* Remember to clear alignment for both images.)
3. Add the HTML code to create three targets (one at the top of the Web page, one near the first h2 heading, and the last near the second h2 heading). Also create two link(s) back to the top with font size small, as shown in Figure 3–50.
4. Validate your HTML code and test all links.
5. Save the revised document as extend3-1solution.html and submit it in the format specified by your instructor.

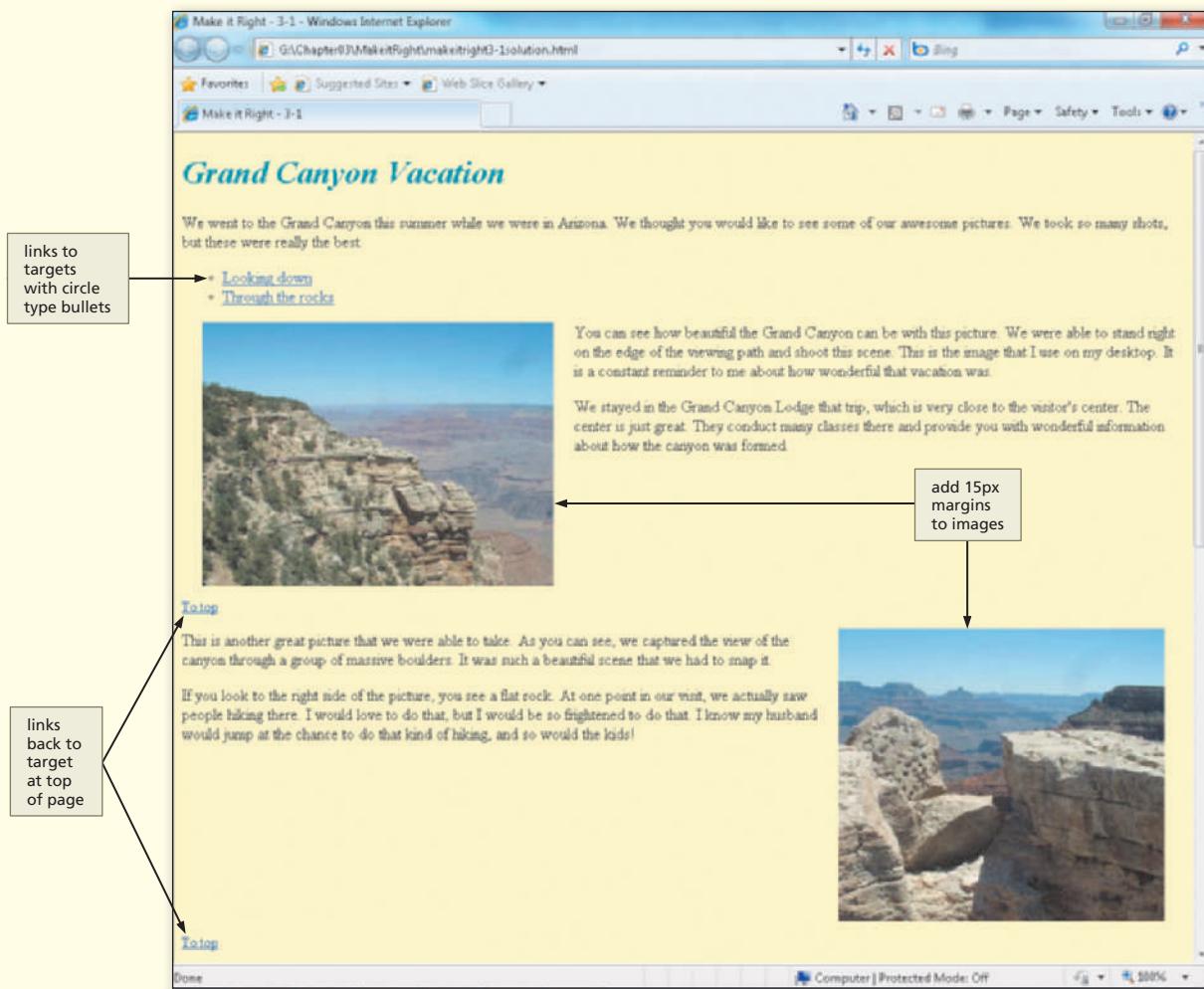
Make It Right

Analyze a document; correct all errors and improve the design.

Correcting the Grand Canyon Web Page

Instructions: Start Notepad++. Open the file makeitright3-1.html from the Chapter03\MakeItRight folder of the Data Files for Students. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files. The Web page is a modified version of what you see in Figure 3–51 on the next page. Make the necessary corrections to the Web page to make it look like the figure. The background color is #ffffc5 for this Web page. Use an inline style to float the images and provide margins. The Web page uses the images grandcanyon1.jpg and grandcanyon2.jpg, which have widths and heights of 346, 259, and 321, 288, respectively.

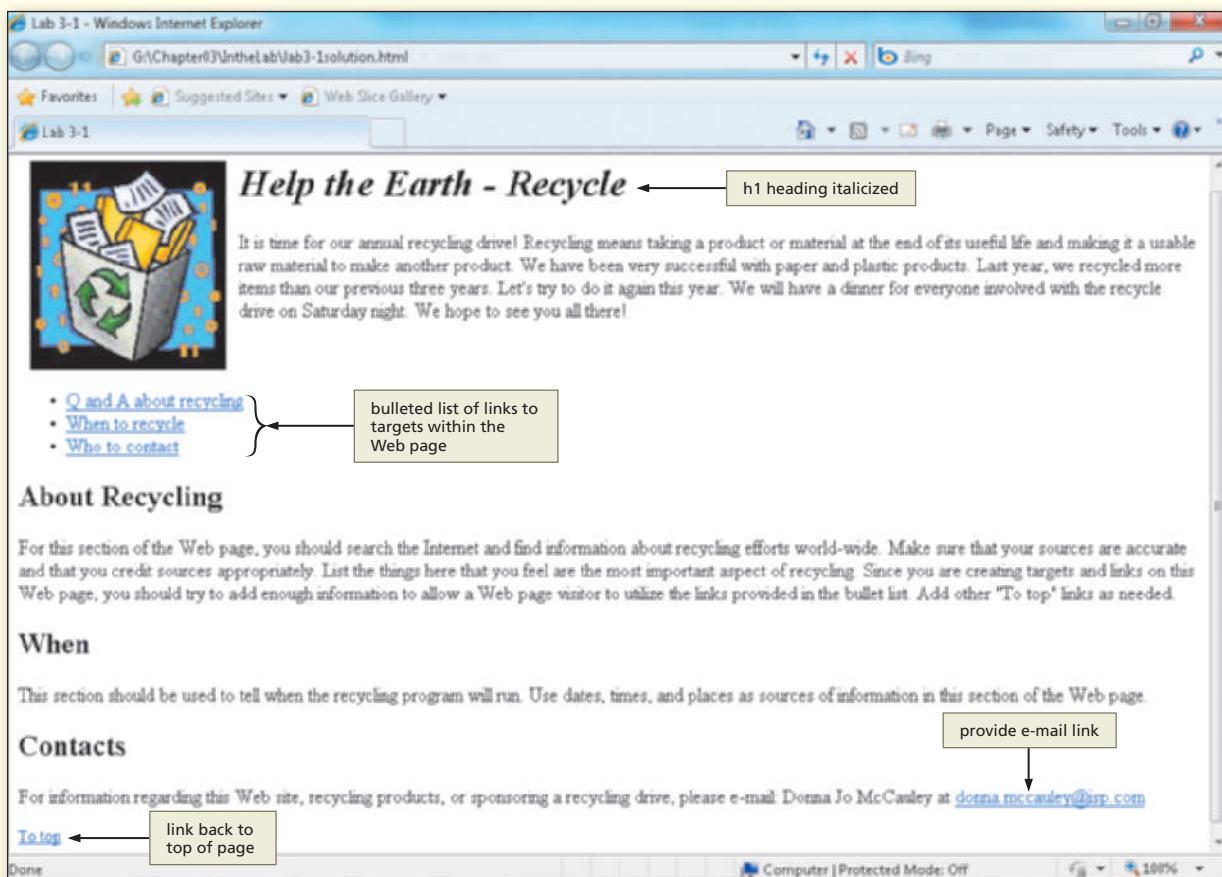
Continued >

Make It Right *continued***Figure 3–51****In the Lab****Lab 1: Creating a Web Page with Links**

Problem: Your instructor wants you to create a Web page demonstrating your knowledge of link targets. You have been asked to create a Web page to demonstrate this technique, similar to the one shown in Figure 3–52. Use inline styles for all styles in the Web page.

Instructions: Perform the following steps:

1. Start Notepad++ and create a new HTML file with the title, Lab 3-1, in the main heading section.
2. Begin the body section by adding the image recycle.gif and aligning it to the left. Use the margin-left-and-right with values of 10 pixels.
3. Add the heading Help the Earth - Recycle. Format the heading to use the Heading 1 style, left-aligned, italic, with the font color black.
4. Add an unordered list with the three list items, as shown in Figure 3–52. These three items will be used to link to the three sections of text below them.

**Figure 3-52**

5. Add a Heading 2 style heading, About Recycling, and set a link target named `about`. Type a paragraph of text based on your research of the topic, as shown in Figure 3-52.
6. Add a Heading 2 style heading, When, and set a link target named `when`. Type a paragraph based on your research of the topic, as shown in Figure 3-52.
7. Add a Heading 2 style heading, Contacts, and set a link target named `contacts`. Type the paragraph, as shown in Figure 3-52.
8. Create a link target at the top of the page named `top`.
9. Create a top link at the bottom of the page, as shown in Figure 3-52. Set the link to direct to the `top` target at the top of the page.
10. Create links from the bulleted list to the three targets.
11. Create an e-mail link, as shown in Figure 3-52.
12. Save the HTML file in the Chapter03\IntheLab folder using the file name lab3-1solution.html.
13. Validate the lab3-1solution.html file.
14. Print the lab3-1solution.html file.
15. Enter the URL `g:\Chapter03\IntheLab\lab3-1solution.html` to view the Web page in your browser.
16. Print the Web page.
17. Submit the HTML file and Web page in the format specified by your instructor.

In the Lab

2: Creating a Web Page with Links

Problem: Your instructor wants you to create a Web page demonstrating your knowledge of link targets. You have been asked to create a Web page to demonstrate this technique, similar to the one shown in Figure 3–53.

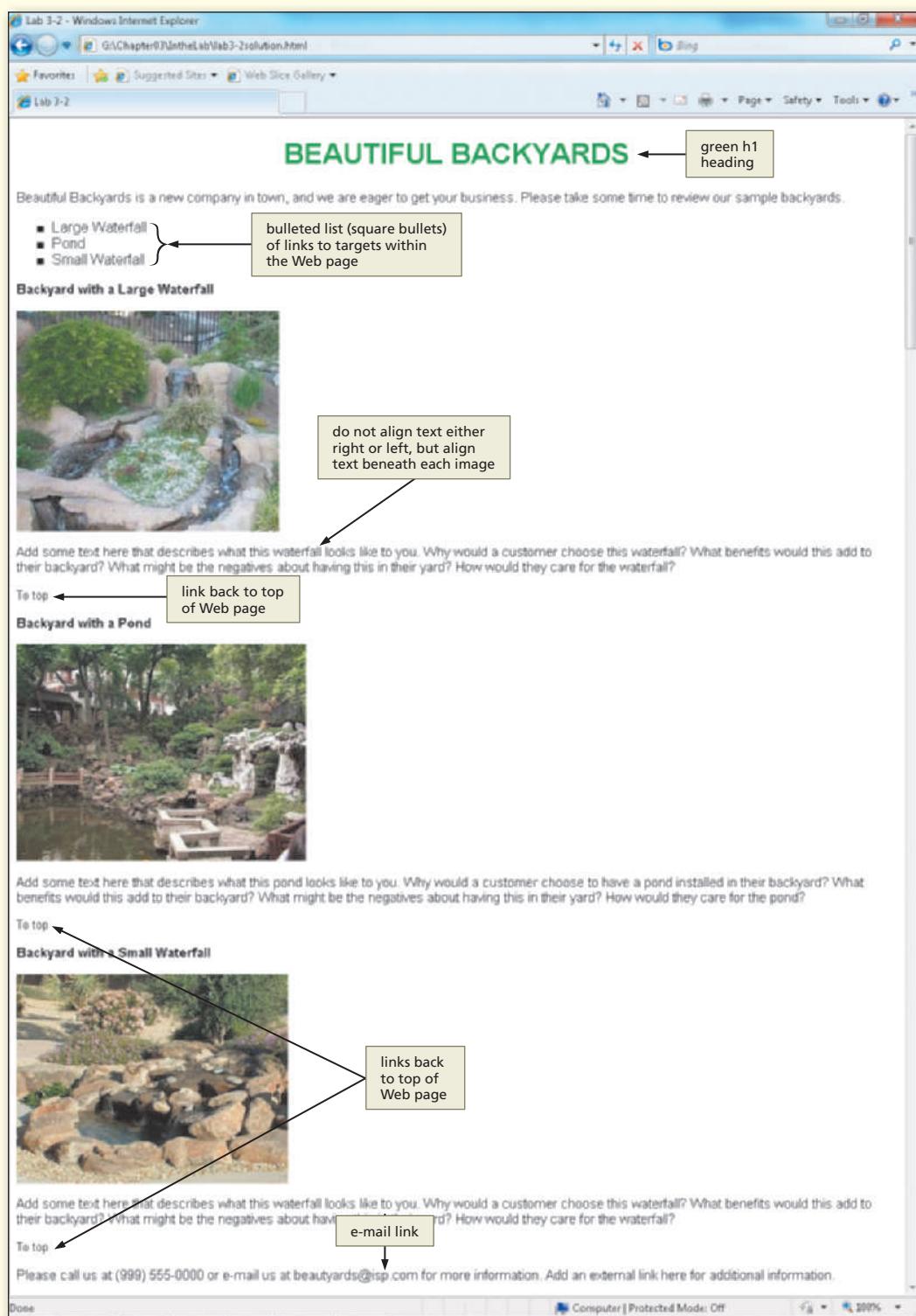


Figure 3–53

Instructions: Perform the following steps:

1. Start Notepad++ and create a new HTML file with the title, Lab 3-2, in the main heading section.
2. Begin the body section by adding an h1 heading that is centered.
3. Add the following code into an embedded style. Notice that all text links have the same background color as the h1 heading at the top of the Web page. This is because of the code that you put in the embedded style, as follows:

```
h1 {color: #00934a;  
    font-family: Arial, Verdana, Garamond}  
  
h2 {color: #00934a;  
    font-family: Arial, Verdana, Garamond}  
  
p {font-family: Arial, Verdana, Garamond;  
    font-size: 11pt}  
  
a {text-decoration: none;  
    color: black;  
    font-family: Arial, Verdana, Garamond}  
  
a:hover {background: #00934a;  
          color: white}  
  
ul {font-family: Arial, Verdana, Garamond;  
    list-style-type: square}
```

4. Add an unordered list with the three list items, as shown in Figure 3–53. These three items will be used to link to the three sections of text below them.
5. Add a title (not a heading) Backyard with a Large Waterfall and use a font-weight of bolder. Add the image largewaterfall.jpg that has a width of 294 and a height of 247.
6. Add a second title, Backyard with a Pond, and image pond.jpg (width=324; height=243). Set a link target named pond. Type a paragraph based on your research of the topic, as shown in Figure 3–53.
7. Add a third title, Backyard with a Small Waterfall, and image smallwaterfall.jpg (width=304; height=234).
8. Create a link target at the top of the page named top.
9. Add three To top links, one after each section, as shown in Figure 3–53. Set the link to direct to the top target at the top of the page.
10. Create links from the bulleted list to the three targets.
11. Create an e-mail link, as shown in Figure 3–53.
12. Save the HTML file in the Chapter03\IntheLab folder using the file name lab3-2solution.html.
13. Validate the lab3-2solution.html file.
14. Print the lab3-2solution.html file.
15. Enter the URL g:\Chapter03\IntheLab\lab3-2solution.html to view the Web page in your browser.
16. Print the Web page.
17. Submit the HTML file and Web page in the format specified by your instructor.

In the Lab

3: Creating Two Linked Web Pages

Problem: Your Communications instructor has asked each student in the class to create a two-page Web site to help students in the class get to know more about the area in which you are majoring in school. She suggested using the basic template shown in Figures 3–54a and 3–54b as a starting point. The first Web page (Figure 3–54a) is a home page that includes basic information about your major. Really try to answer the questions listed (for example, why you chose this major), as shown on the Web page. If you can, add an image related to your chosen field somewhere on the Web page. Add a link to the second Web page. The second Web page (Figure 3–54b) includes a paragraph of text and numbered lists with links.

Instructions: Perform the following steps:

1. Start Notepad++ and create a new HTML file with the title Lab 3-3 in the main heading section.
2. In the first Web page, include a Heading style 1 heading, similar to the one shown in Figure 3–54a, and a short paragraph of text. Experiment and use any color for the heading (navy is shown).
3. Create a text link to the second Web page, lab3-3specifics.html.
4. Save the HTML file in the Chapter03\IntheLab folder using the file name lab3-3solution.html. Validate the lab3-3solution.html file. Print the lab3-3solution.html file.
5. Start a new HTML file with the title Lab 3-3 Specifics in the main heading section.
6. In the second Web page, include a Heading style 1 heading, similar to the one shown in Figure 3–54b, a short paragraph of text, and two Heading style 2 headings. Use any color for the headings; navy is

(a) First Web page.

This two-page Web site tells the reader something about the subject area that you are focusing on in school. In this section of the page, please explain the reason why you chose to major in this area. What interested you in this field? Who (if anyone) has influenced you to work in the area? Think of the reasons why you would work so hard in school to pursue a dream job in this field.

(b) Second Web page.

Insert a picture or two related to this field.

Click this link to see [specifics](#) about this field.

provide link to second page of Web site

use blue colored h1 headings

Add some information about the types of careers that you can pursue after you graduate from school with a degree in this major. Comment on reasons why you are well-suited for that career.

Careers available in this field:

1. [List one type of career, list URL for information](#)
2. [List a second career, list URL for information](#)
3. [List a third career, list URL for information](#)

use blue colored h1 headings

add two ordered lists as shown

Other information

1. [List range of salaries in the field, list URL for information](#)
2. [List areas in the country \(world\) that have available jobs, list URL for information](#)
3. [List the cost of living in those areas of the country \(world\), list URL for information](#)

Click this link to return to the [home page](#)

provide link back to home page

use blue colored h1 headings

Figure 3-54

shown in the h1 and black is used in the h2 headings. From the standpoint of consistency, you may want to make those headings all the same color. (*Hint:* Use an embedded style for this.)

7. Create two ordered (numbered) lists with at least two items that serve as links to Web pages on another Web site. Add a link back to the first Web page, as shown in Figure 3–54b.
8. Save the HTML file in the Chapter03\IntheLab folder using the file name lab3-3specifics.html. Validate the lab3-3specifics.html file. Print the lab3-3specifics.html file.
9. Enter the URL g:\Chapter03\IntheLab\lab3-3solution.html to view the Web page in your browser. Click the text link to the second Web page. Click the links in the lists to test them.
10. Print the Web pages.
11. Submit the HTML file and Web page in the format specified by your instructor.

Cases and Places

Apply your creative thinking and problem-solving skills to design and implement a solution.

• EASIER •• MORE DIFFICULT

• 1: Add a Web Page to the Getting Greener Site

Laquisha Carter is very impressed with the Getting Greener Web pages and now would like to add a Web page listing other “green” things that you can do to help the environment. Search the Web to find at least four Web sites that contain information about other environmentally responsible things that you can do. Create a Web page that includes a Heading 1 style heading, a brief paragraph of descriptive text, and list links to those Web sites. Modify the Getting Greener home page to include a link to the new Web page.

• 2: Create a Web Page with Text Formatting

You recently got an internship developing Web pages for Jose Sanchez Painters. Mr. Sanchez has asked you to update the home page for his Web site to make it more visually appealing. As a first step, you plan to create a Web page with sample text formats, such as the ones shown in Figure 3–20 on page HTML 116, to share with Mr. Sanchez and get his input on which types of formatting he prefers. Create such a Web page and include text formatted as bold, italic, underlined, superscript, and subscript; use different colors and sizes for each type of text. Use CSS styles for most of your styles, but also use some of the text formatting tags shown in the chapter. Determine why you would choose to use one over the other (i.e., styles versus formatting tags). Be sure to include one sample using the font-weights of bold tag and one using bolder to see how they compare when displayed together. Compare those to using `` or `` as formatting tags. Be creative and make sure that the Web page looks good but that the new styles do not distract from the content. Be prepared to explain to your client, Mr. Sanchez, why some font-families (e.g., something very cursive style) might not be appropriate.

•• 3: Add Image Links to External Sites

To update the Getting Greener Web site further, you want to add image links so the pictures on the Green Home Web page also are links to Web pages in an external Web site. Search the Web for information specific to each of the two Green Home images used in the project. Modify the Green Home Web page so each “green” image is used as a link to a Web page in an external Web site. After adding the links, you decide the text paragraphs on the page are too close to the pictures. Modify the Green Home Web page to use the properties of float together with margins in the `` element to add space around each image. Also try two variations of other properties you can use with the `` element in addition to float and margin. (*Hint:* See Appendix D for ideas.)

Continued >

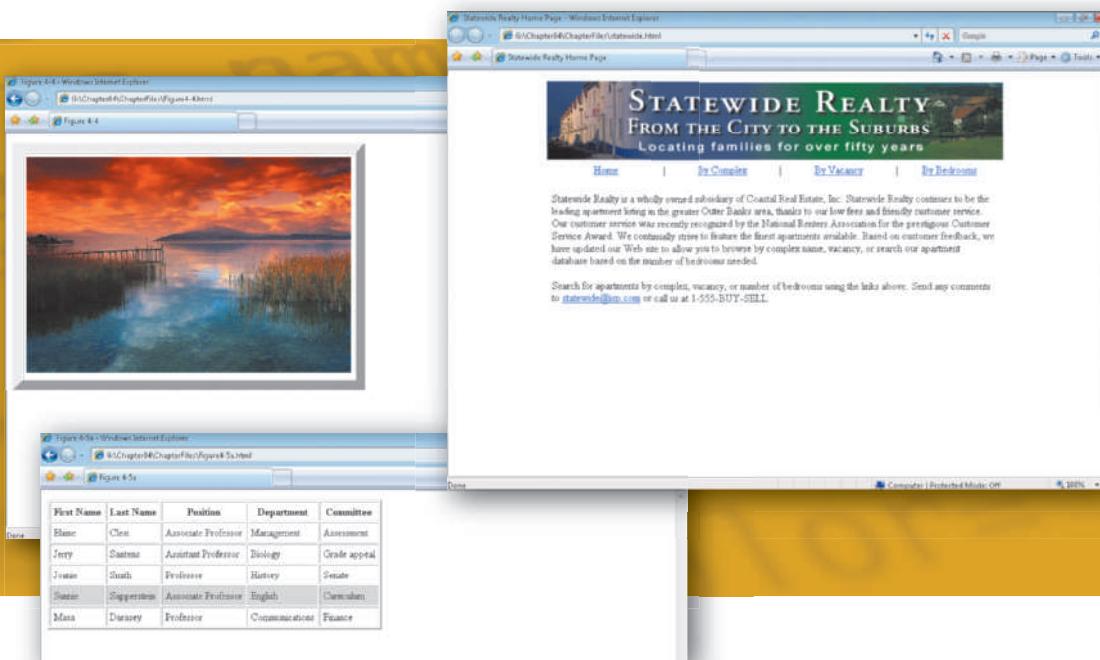
Cases and Places *continued***•• 4: Create a Web Page with Text Links and Define Link Colors****Make It Personal**

Your sister owns a clothing store and recently had a Web site developed for her company. She is unhappy that the links on the company Web pages appear in blue when unvisited and purple when visited, because those colors do not match the company logo. She has asked you to update the Web pages. Create a Web page similar to Figure 3–4 on page HTML 94, with three text links to a Web page in an external Web site. Add the appropriate link styles to define the link colors requested by your sister. Also, explain why you might not want to change the colors of the links from the standard blue and violet but show ways that you can accommodate different colors and not confuse users.

•• 5: Create a Prototype Web Site with Five Pages**Working Together**

Your manager at Uptown Enterprises has asked your team to create a simple five-page prototype of the Web pages in the new Entertainment section for the online magazine CityStuff. The home page should include headings and brief paragraphs of text for Arts, Music, Movies, and Dining. Within each paragraph of text is a link to one of the four detailed Web pages for each section (for example, the Arts link should connect to the Arts Web page). The home page also includes an e-mail link at the bottom of the page. Add a To top link that connects to a target at the top of the page. The four detailed Web pages should include links to external Web sites of interest and a link back to the home page. If possible, also find appropriate images to use as a background or in the Web page, and set text to wrap around the images. Remember to use CSS. Determine during the design phase whether it would be better to use inline or embedded style sheets or both for this Web site.

4 Creating Tables in a Web Site Using an External Style Sheet



Objectives

You will have mastered the material in this chapter when you can:

- Define table elements
- Describe the steps used to plan, design, and code a table
- Create a borderless table for a horizontal menu bar with text links
- Create an external style sheet to define styles across a Web site
- Utilize classes to give you more control over styles
- Link an external style sheet to Web pages where you want its styles applied
- Copy and paste HTML code to a new file
- Create a table with borders and insert text
- Alter the spacing between and within cells using the cellspacing and cellpadding attributes
- Utilize inline styles to alter the style of individual elements on a Web page
- Add background color to rows and cells
- Insert a caption below a table
- Create headings that span rows using the rowspan attribute

4 Creating Tables in a Web Site Using an External Style Sheet

Introduction

So far, you have learned how to make a basic Web page and how to link one Web page to another, both within the same Web site and external to the Web site. You also learned how to create inline styles and embedded style sheets to alter the appearance of Web page elements. In this chapter's project, you create an external (linked) style sheet to set the style for elements across multiple Web pages. The project also adds to your HTML knowledge by teaching you how to organize and present information on a Web page using tables with rows and columns. In this chapter, you learn about the elements used in a table and how to plan, design, and code a table. You learn how to use a table to create a horizontal menu bar with text links, and to create tables to organize text and images. You will enhance the Web site by manipulating the properties and attributes of tables and paragraphs, altering borders, colors, and spacing, and adding a table caption.

Project — Sabatina's Pizza Web Site

Having a reliable Web site makes it easier for a company's customers to find the establishment, provides a way to communicate the company's brand, and allows the company to provide additional services. As advertising director for Sabatina's Pizza, you want to enhance Sabatina's Web site to increase the company's exposure to current and new customers, and to incorporate ideas gathered from customer feedback surveys. The new site will allow customers to browse through tables of information that outline the appetizers, salads, and pizzas offered by the restaurant.

As shown in Figure 4–1a, the Sabatina's Pizza home page includes a company logo banner and a borderless table that contains a menu bar under the logo. The borderless table gives users easy access to all pages in the Web site. This table is available on every Web page in the Web site. The Appetizers, Salads, and Pizza Web pages (Figures 4–1b, 4–1c, and 4–1d) each include the company logo banner and the same borderless table at the top, as well as one table with borders that displays the contents of that particular Web page. In this project, you will create the `sabatinas.html` and `appetizers.html` Web pages. You will edit the `salads.html` Web page (Figure 4–1c) to add `cellspacing` and `cellpadding` attributes, thereby adjusting the spacing between cells. The `pizza.html` Web page file (Figure 4–1d) is also edited to add a caption with information about the table and to use the `rowspan` attribute to create headings that span several rows.

As you read through this chapter and work on the project, you will learn how to plan, design, and code tables to create a user-friendly Web site. You also will learn to format tables and to combine table features to make the pages more readable. In addition, you will learn to create a menu bar with text links.

(a) Sabatina's Pizza home page.

company logo banner image inserted at top of all Web pages in Web site

menu bar for navigation inserted in all Web pages

50-pt margin added on right and left sides of all paragraphs

e-mail link added to home page

Sabatina's Pizza recently opened in the Highlands. It has already become a popular place for family and friends to meet and have a great meal. Sabatina's offers many kinds of pizza, both thin and thick crust. They also have delicious appetizers and salads. There is ample seating on the inside of the restaurant and more seating on our lovely shaded patio outside. We have carry-out orders and also make deliveries.

Please look through our Web site to learn more about our menu. Want to place a carry-out order? Call us at (999) 555-SABAS or e-mail us at sabatinas@isp.com.

(b) Appetizers Web page.

link back to Home page

Appetizers link bold and italicized to indicate user is on Appetizers Web page

Item	Size	Cost
Breadsticks	4 pieces	\$3.95
Calamari	Serves 6	\$8.95
Mozzarella sticks	6 pieces	\$5.95
Stuffed mushrooms	Serves 6	\$6.95

table with 5-px border and header with background color and white text

Salads menu option bold and italicized to indicate user is on Salads Web page

(c) Salads Web page.

Item	Medium	Large
Caesar	\$4.95	\$8.95
Mediterranean	\$4.95	\$6.95
Greek	\$4.95	\$8.95
Japanese	\$4.95	\$7.95

cellpadding (5px) and cellspacing (10px) added to enhance style of table

Home | Appetizers | **Salads** | Pizza

Pizza menu option bold and italicized to indicate user is on Pizza Web page

(d) Pizza Web page.

Thin Crust Pizza	Cheese	Medium - \$7.95	Large - \$9.95
	Sausage	Medium - \$8.95	Large - \$10.95
	Veggie	Medium - \$8.95	Large - \$10.95
Thick Crust Pizza	Cheese	Medium - \$9.95	Large - \$11.95
	Sausage	Medium - \$10.95	Large - \$12.95
	Veggie	Medium - \$10.95	Large - \$12.95

Additional ingredients include mushrooms, olives, pepperoni - \$2.00 per item

same cellpadding and cellspacing used as in Salads Web page

caption inserted beneath table to clarify information

rowspan attributes used twice to span three rows each

Figure 4-1

Overview

As you read this chapter, you will learn how to create the Web pages shown in Figures 4–1a through 4–1d on the previous page by performing these general tasks:

- Enter HTML code into the Notepad++ window.
- Save the file as an HTML file.
- Enter basic HTML tags and add text to the file.
- Add a horizontal menu bar with text links.
- Create a table with borders to display information in an organized manner.
- Create an external style sheet to set the style for all Web pages in the Web site.
- Use classes to give more control over the styles used.
- Link an external style sheet to Web pages.
- Utilize inline styles to alter the style of individual elements on the Web page.
- Add HTML tags that enhance a table with cellpadding and cellspacing.
- Enhance a Web table with rowspanning.
- Add a caption to a table.
- Print the HTML code and Web pages

Plan Ahead

General Project Guidelines

When creating a Web page, the actions you perform and decisions you make will affect the appearance and characteristics (the styles) of the finished page. As you create Web pages, such as those shown in Figures 4–1a through 4–1d, you should follow these general guidelines:

1. **Complete Web page planning.** Before developing a Web page, you must know the purpose of the Web site, identify the users of the site and their computing environments, and decide who owns the information on the Web page.
2. **Analyze the content and organization of the Web page.** In the analysis phase of the Web development life cycle, you should analyze what content to include on the Web page and how to organize that information. In this phase, you need to determine what information you want to convey so that you can highlight that information on the Web page using different techniques. Refer to Table 1–4 on page HTML 15 for information on the phases of the Web development life cycle.
3. **Choose the content and organization for the Web page.** Once you have completed the analysis, you need to determine specifically what content to include on the Web page. With tables, you are able to display the Web page content in a very organized manner. Tables can be used to display text only, as well as graphical images or combinations of text and images. Some text is better highlighted by using different colors for column or row headings. Other information is displayed more effectively with row- and column-spanning techniques. This should all be determined before coding the Web pages.
4. **Identify how to format various elements of the Web page.** The overall appearance or style of a Web page significantly affects its ability to communicate clearly. Additionally, you want to provide easy navigation for your Web site visitors. Adding images and color helps to communicate your message and adding a menu bar with links to the other Web pages within the Web site makes it easy to navigate the Web site. Determine what style sheets to use for the overall appearance or style, including external, embedded, and inline style sheets. Also determine what formatting tag attributes need to be implemented.

(continued)

5. **Determine where to save the Web page.** You can store a Web page permanently, or save it on a variety of storage media, including a hard disk, USB flash drive, CD, or DVD. You also can indicate a specific location on the storage media for saving the Web page. Recognize the appropriate absolute and relative addressing that you will need as determined by your analysis.
6. **Create the Web page and links.** After analyzing and designing the Web site, you need to develop the individual Web pages. It is important to maintain a consistent look throughout the Web site. Use graphics and links consistently so that your Web site visitor does not become confused.
7. **Test all Web pages within the Web site.** An important part of Web development is testing to assure that you are following XHTML standards. This book uses the World Wide Web Consortium (W3C) validator that allows you to test your Web page and clearly explains any errors you have. Additionally, you should check all content for accuracy and test all links.

When necessary, more specific details concerning the above guidelines are presented at appropriate points in the chapter. The chapter also will identify the actions performed and decisions made regarding these guidelines during the creation of the Web pages shown in Figures 4–1a through 4–1d on page HTML 159.

Plan Ahead

Planning and Designing a Multipage Web Site

The Web site that you create in this chapter consists of four Web pages: sabatinas.html, appetizers.html, salads.html, and pizza.html. With a multipage Web site, you need to design the overall look of the Web site itself, as well as the individual Web pages. You will use both formatting tag attributes and style sheets to create the overall appearance (or style) of the Web site. An **external style sheet** is used to define styles for multiple pages in a Web site. With external (linked) style sheets, you create the style sheet first in a separate file saved with a .css extension. You then link this style sheet into any Web page in which you want to use it.

In Chapters 2 and 3, you learned how to insert inline and embedded style sheets into your Web pages. Recall that inline style sheets are used to change the style of an individual HTML tag. An embedded style sheet is inserted between the <head> and </head> tags of a single Web page within the style container (<style> and </style>). Embedded style sheets are used to change the style for elements on an entire Web page. For the project in this chapter, where you have a multipage Web site, you will learn how to create an external, or linked, style sheet.

Creating Web Pages with Tables

Tables allow you to organize information on a Web page using HTML tags. Tables are useful when you want to arrange text and images into rows and columns in order to make the information straightforward and clear to the Web page visitor. You can use tables to create Web pages with newspaper-type columns of text or structured lists of information. Tables can be complex, using the rowspan and colspan attributes to span rows and columns, background colors in cells, and borders to provide formatting (Figure 4–2a on the next page). Tables also can be simple, with a basic grid format and no color (Figure 4–2b). The purpose of the table helps to define what formatting is appropriate.

(a) Complex table.

The screenshot shows a table titled "Percentage of Online Courses Completed" with 6 rows and 8 columns. A column header "Module" spans 6 rows. A row header "Abner" spans 8 columns. Arrows point from callout boxes to specific features:

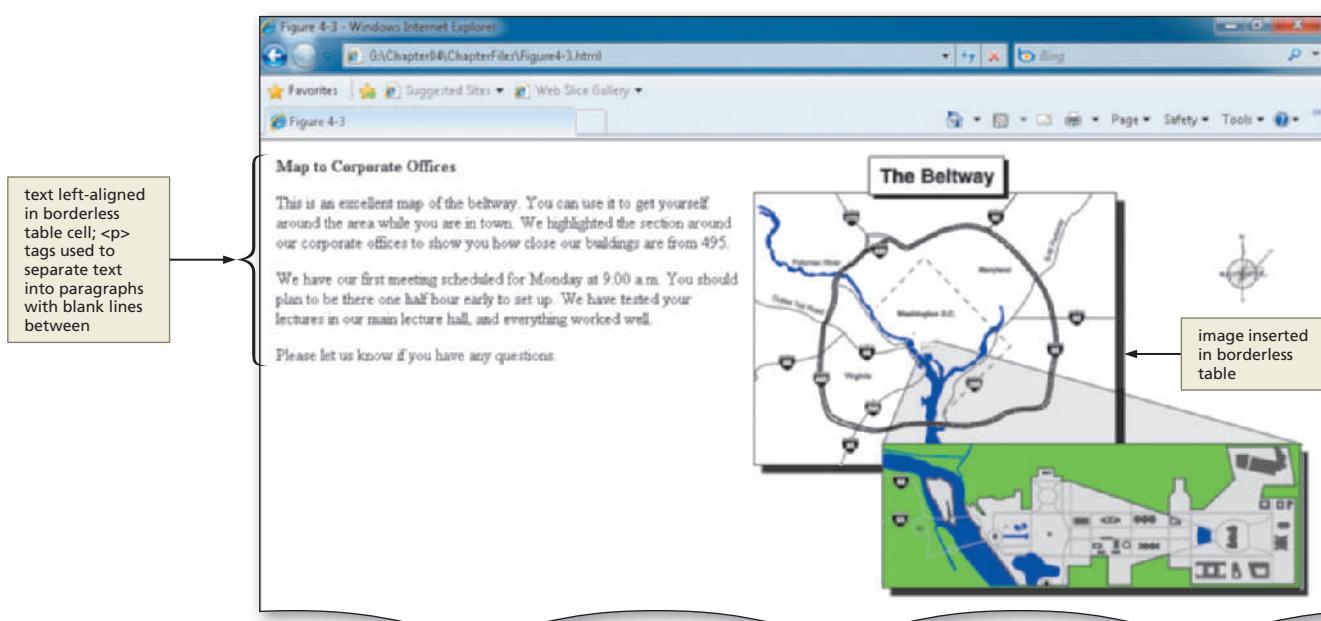
- A box labeled "background colors added to enhance style of table" points to the header row.
- A box labeled "colspan attribute used to span eight columns" points to the "Abner" header cell.
- A box labeled "border attribute used for cell borders" points to the border of the table.
- A box labeled "rowspan attribute used to span six rows" points to the "Module" header cell.

(b) Simple table.

The screenshot shows the same table structure as Figure 4-2a, but without any additional styling like background colors or borders. A callout box points to the text "basic grid; no color added to table".

Figure 4–2

In Chapter 3, you learned how to wrap text around an image. You also can use tables to position text and images, such as the one shown in Figure 4–3; this is a borderless table used to position text to the left of the map images. An advantage of using a table to position text and images instead of just wrapping the text around the image is that you have greater control over the placement of the text and image.

**Figure 4–3** Image and text positioned in table.

Tables also can be used to create a border or frame around an image. Figure 4–4 shows a Web page with an image inserted into a table with one row and one cell. The border is set to a pixel width of 15 to create the appearance of a frame. Using a table to create a frame is a simple technique that gives an image a polished look and highlights the image.

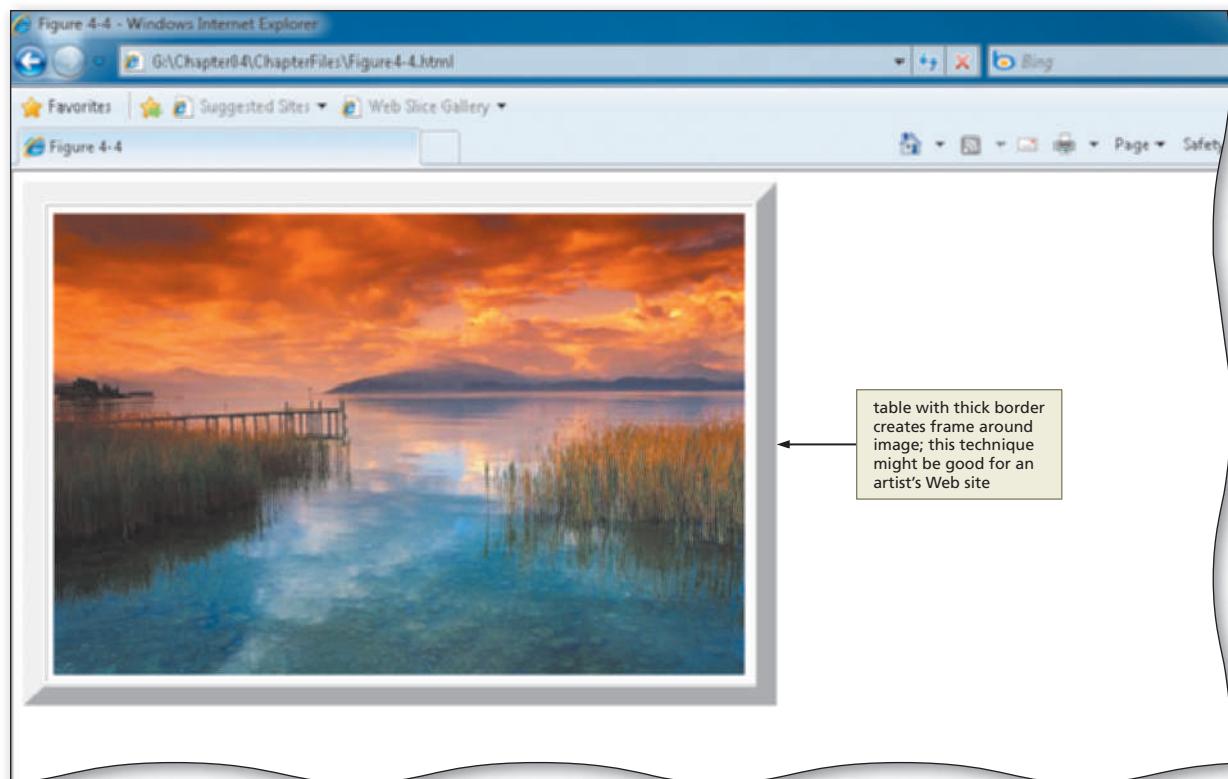


Figure 4–4 Table used as image frame.

Table Elements

Tables consist of rows, columns, and cells, much like spreadsheets. A **row** is a horizontal line of information. A **column** is a vertical line of information. A **cell** is the intersection of a row and a column. Figure 4–5 on the next page shows examples of these three elements. In Figure 4–5a, the fifth row in the table has a gray background. In Figure 4–5b, the fourth column has a peach background. In Figure 4–5c, the cell at the intersection of column 2 and row 6 has a gold background.

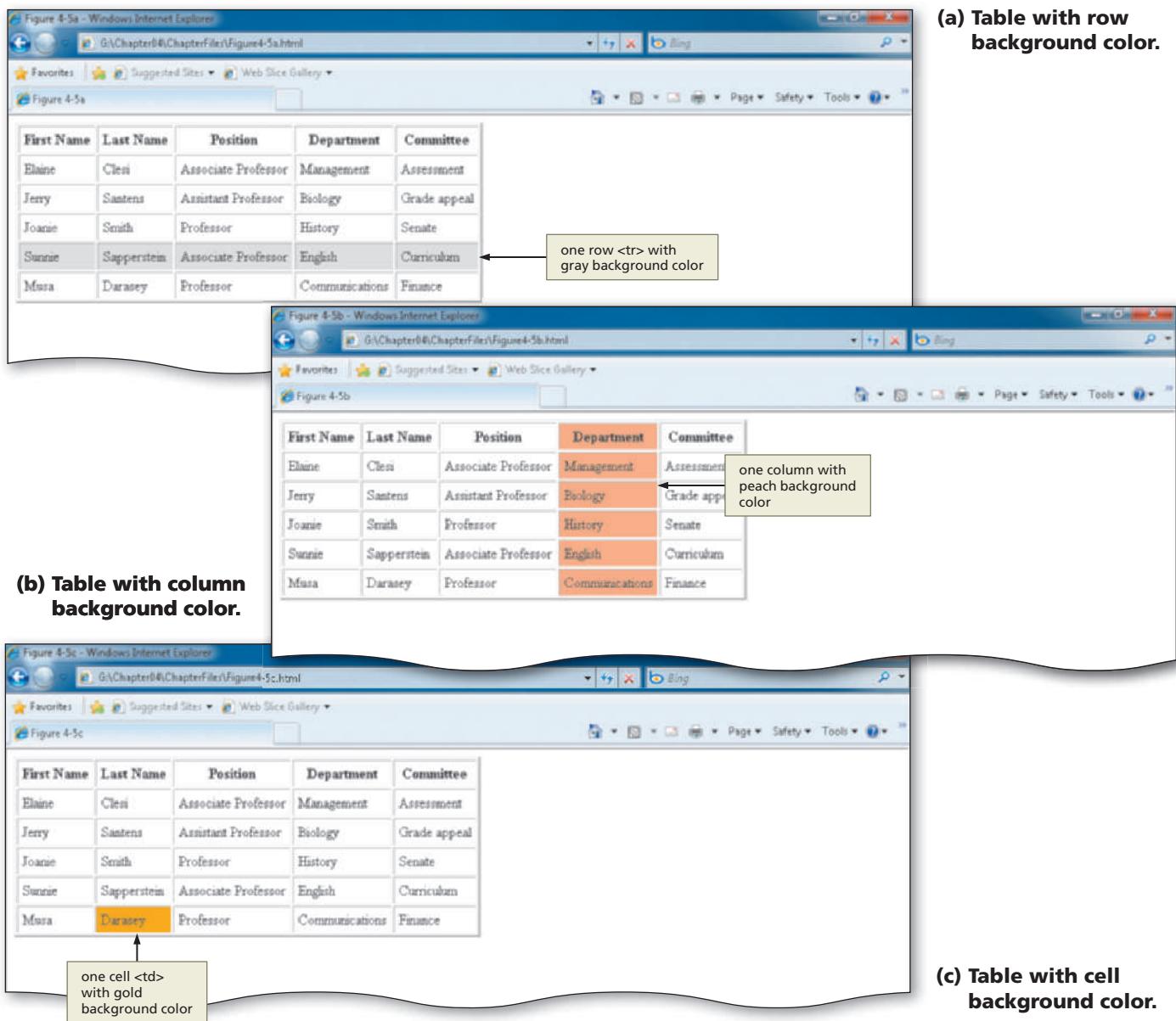
As shown in Figure 4–5c, a cell can be one of two types: a heading cell or a data cell. A **heading cell** displays text as bold and center-aligned. A **data cell** displays normal text that is left-aligned.

Understanding the row, column, and cell elements is important as you create a table using HTML. Properties and attributes are set relative to these table elements. For example, you can set attributes for an entire row of information, for a single cell, or for one or more cells within a row.

BTW

Tables

Tables are useful for a variety of purposes. They can store information in tabular form or create a layout on a Web page. Layouts created with tables give more flexibility to the Web developer. You have more control over the placement of information or images. Many popular Web sites use tables.

**Figure 4-5****BTW****Table Elements**

Many Web sources

discuss table parts, giving numerous examples and tips. For more information about HTML table parts, search the Web for key terms such as HTML Table Elements or Table Properties.

Adding Style to Table Elements

As discussed earlier, there are many ways to set the style (or appearance) of a Web page element. You can use formatting tag attributes (e.g., setting cellpadding and cellspacing on a table) to set the appearance or style. You also can use inline, embedded, or external (linked) style sheets to set the style. During the design phase of the Web development life cycle, you determine how to set the styles for the elements on your Web pages. In this chapter, you use an external (linked) style sheet to set styles for the tables across the entire Web site.

Table Borders, Headers, Captions, and Rules

Tables include features such as table borders, table headers, table captions, and rules attributes (Figure 4–6). A **table border** is the line that encloses the perimeter of the table. A **table header** is the same as a heading cell — it is any cell with bold text that indicates the purpose of the row or column. A header row is used to identify the meaning of the numbers in each column, and headings that span columns and rows are used to provide additional information. Headers also are used by non-visual browsers to identify table content. See the guidelines in Appendix C for specific information about making your Web pages available to the nearly 20% of the world population who have some sort of disability. A **table caption** is descriptive text located above or below the table that further describes the purpose of the table.

Tables can use these features individually or in combination. The purpose for the table dictates which of these features are used. For example, the table shown in Figure 4–6 lists columns of numbers. A header row is used to identify the meaning of the numbers in each column, and headings that span columns and rows are used to provide additional information. Finally, the table caption explains that each number is based on thousands (that is, the 10 listed in the table represents 10,000).

The screenshot shows a Microsoft Internet Explorer window displaying a table titled "Safety Equipment for Roller Blading". The table has a border and contains data representing units sold in thousands. Several features are annotated with callouts:

- A callout points to the first row of the table, labeled "table header with rowspan attribute used to span four rows".
- A callout points to the first column of the table, labeled "table header with colspan attribute used to span five columns".
- A callout points to the right edge of the table, labeled "table border".
- A callout points to the bottom of the table, labeled "table caption positioned beneath table by caption-side: bottom in style sheet".
- A callout points to the text "Numbers listed in thousands" located just below the table.

Safety Equipment for Roller Blading

Units Sold in First Quarter	Helmets	Wrist Protectors	Elbow Pads	Knee Pads
	10	35	7	36
	8	25	17	9
	36	7	20	15

Numbers listed in thousands

Figure 4–6 Table headers, border, and caption.

Another useful table feature is the **rules** attribute, which creates horizontal or vertical lines in a table. The **rules** attribute allows a Web developer to select which internal borders to show in a table. It supports several values to provide different formatting options. For example, using **rules="none"** creates a table with no internal rules. Using **rules="cols"** creates a table with vertical rules between each column in the table (Figure 4–7a), while **rules="rows"** creates a table with horizontal rules between each row in the table (Figure 4–7b). Appendix A provides additional information on values supported by the **rules** attribute.

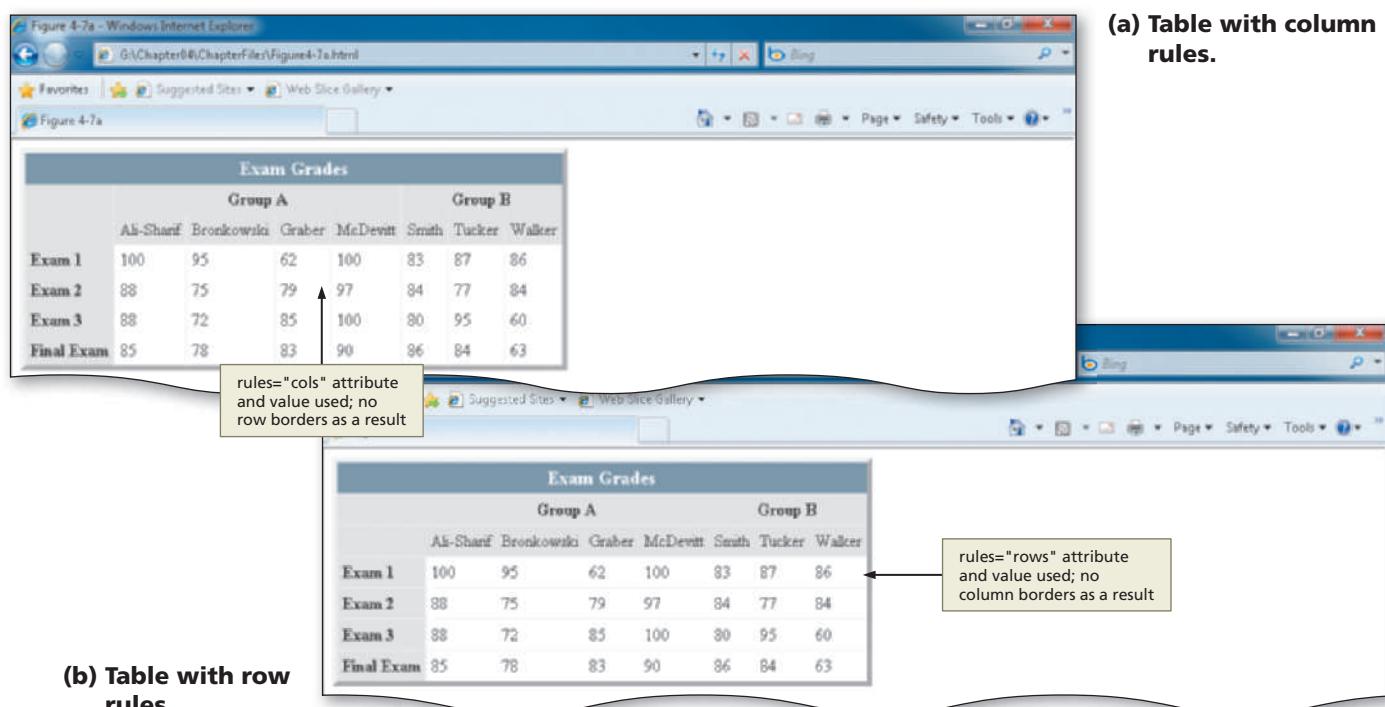


Figure 4–7

Determining the Need for, Planning, and Coding a Table

Creating tables for a Web page is a three-step process: (1) determining if a table is needed, (2) planning the table, and (3) coding the table. Each of these steps is discussed in detail in the following sections.

Determining If a Table Is Needed

First, you must determine whether a table is necessary. Not all Web pages require the use of tables. A general rule is that a table should be used when it will help organize information or Web page elements in such a way that it is easier for the Web page visitor to read. Tables generally are useful on a Web page if the Web page needs to display a structured, organized list of information or includes text and images that must be positioned in a very specific manner. Figures 4–8a and 4–8b show examples of information displayed as text in both a table and a bulleted list. To present this information, a table (Figure 4–8a) would be the better choice. The bulleted list (Figure 4–8b) might give the Web page an acceptable look, but the table presents the information more clearly.

(a) Web page with table of information.

The screenshot shows a Microsoft Internet Explorer window titled "Figure 4-8a - Windows Internet Explorer". The address bar shows the path "G:\Chapter04\Chapter files\Figure4-8a.html". The main content area displays a table titled "Metric Conversions" with two sections: "Weight" and "Volume Liquid". The "Weight" section contains three rows: 1 ounce (28.4 grams), 1 pound (454 grams), and 2.2 pounds (1 kilogram). The "Volume Liquid" section contains four rows: 1 teaspoon (4.7 milliliters), 1 tablespoon (14.2 milliliters), 1 cup (227 milliliters), and 1.06 quarts (1000 milliliters or 1 liter). A callout box labeled "table of information" points to the table.

(b) Web page with bulleted lists of information.

The screenshot shows a Microsoft Internet Explorer window titled "Figure 4-8b". The main content area displays the same "Metric Conversions" information as in figure 4-8a, but it is presented in bulleted lists under two sections: "Weight" and "Volume". The "Weight" section lists: 1 ounce = 28.4 grams, 1 pound = 454 grams, and 2.2 pounds = 1 kilogram. The "Volume" section lists: 1 teaspoon = 4.7 milliliters, 1 tablespoon = 14.2 milliliters, 1 cup = 227 milliliters, and 1.06 quarts = 1000 milliliters or 1 liter. A callout box labeled "same information presented in bulleted lists" points to the lists.

Figure 4-8

Planning the Table

To create effective tables, you must plan how the information will appear in the table and then create a good design. Before writing any HTML code, sketch the table on paper. After the table is sketched on paper, it is easier to see how many rows and columns to create, if the table will include headings, and if any of the headings span rows or columns. Conceptualizing the table on paper first saves time when you try to determine which HTML table tags to use to create the table.

For example, to create a simple table that lists the times run by various cross-country team members, you might sketch the table shown in Figure 4-9a on the next page. If runners participate in two different race lengths, such as 5K and 10K, that information can be included in a table designed as shown in Figure 4-9b. If the table needs to include different race dates for each race length, that information can be included in a table such as the one shown in Figure 4-9c. Finally, to make the table easier for the Web page visitor to understand, the table should include headings that span rows and columns and a caption. For instance, in Figure 4-9b, the headings 5K and 10K each span two columns of data. Because column spanning is used, you can easily see which runners ran in the 5K or 10K races. In Figure 4-9c, because of row spanning, you can easily

tell what date each race was run. Design issues such as these should be considered in the planning stage before any HTML code is entered. Figure 4–10 shows how the table might look after it is coded. You will use a variety of style sheets together with formatting tag attributes to create the tables for the project in this chapter.

BTW

Table Tutorial

Table tutorials are available through online sources. Tutorials take you step-by-step through the table creation process. Search the Web for the phrase HTML Table Tutorial to find excellent sources of information.

NAME1	NAME2	NAME3	NAME4
TIME	TIME	TIME	TIME

(a) Simple table.

5K		10K	
NAME1	NAME2	NAME3	NAME4
TIME	TIME	TIME	TIME

(b) Column spanning added.

		5K		10K	
		NAME1	NAME2	NAME3	NAME4
Meet Dates	MAY 5	TIME	TIME	TIME	TIME
	MAY 12	TIME	TIME	TIME	TIME
	MAY 19	TIME	TIME	TIME	TIME
	MAY 26	TIME	TIME	TIME	TIME

(c) Row spanning added.

Figure 4–9

CROSS COUNTRY TEAM					
Meet Dates	5 K		10 K		
	May 5	Hruskoczy	Rodriguez	Eegesberg	Franzle
	May 12	15:00	19:20	29:51	20:47
	May 19	18:03	18:12	28:53	22:37
	May 26	19:10	21:12	22:59	20:07

Figure 4–10 Table with row and column spanning.**Coding the Table**

After you have completed the table design, you can begin coding the table using HTML tags. Table 4–1 shows the four main HTML tags used to create a table. Each of these tags has a number of attributes, which are discussed later in this chapter.

Figure 4–11a shows an example of these tags used in an HTML file, and Figure 4–11b shows the resulting Web page. As shown in Figure 4–11b, the table has four rows (a table header and three rows of data cells) and two columns. The rows are indicated in the HTML file

Table 4-1 HTML Table Tags

Tag	Function
<table></table>	<ul style="list-style-type: none"> Indicates the start and end of a table All other table tags are inserted within these tags
<tr> </tr>	<ul style="list-style-type: none"> Indicates the start and end of a table row Rows consist of heading or data cells
<th> </th>	<ul style="list-style-type: none"> Indicates the start and end of a table heading (also called a heading cell) Table headings default to bold text and center-alignment
<td> </td>	<ul style="list-style-type: none"> Indicates the start and end of a data cell in a table Data cells default to normal text and left-alignment

in Figure 4–11a by the start `<tr>` tags and the end `</tr>` tags. For this simple table, the number of columns in the table is determined based on the number of cells within each row. As shown in Figure 4–11b, each row has two cells, which results in a table with two columns. (Later in this chapter, you will learn how to indicate the number of columns within the `<table>` tag.)

As shown in the HTML in Figure 4–11a, the first row includes table heading cells, as indicated by the start `<th>` tag and end `</th>` tag. In the second, third, and fourth rows, the cells contain data, indicated by the start `<td>` tag and end `</td>` tag. In the resulting table, as shown in Figure 4–11b, the table header in row 1 appears as bold and centered text. The text in the data cells in rows 2 through 4 is left-aligned and normal text. The table in Figure 4–11b has a border, and cellspacing of 5 pixels was added to highlight further differences between the cells. You learn about cellspacing later in the chapter.

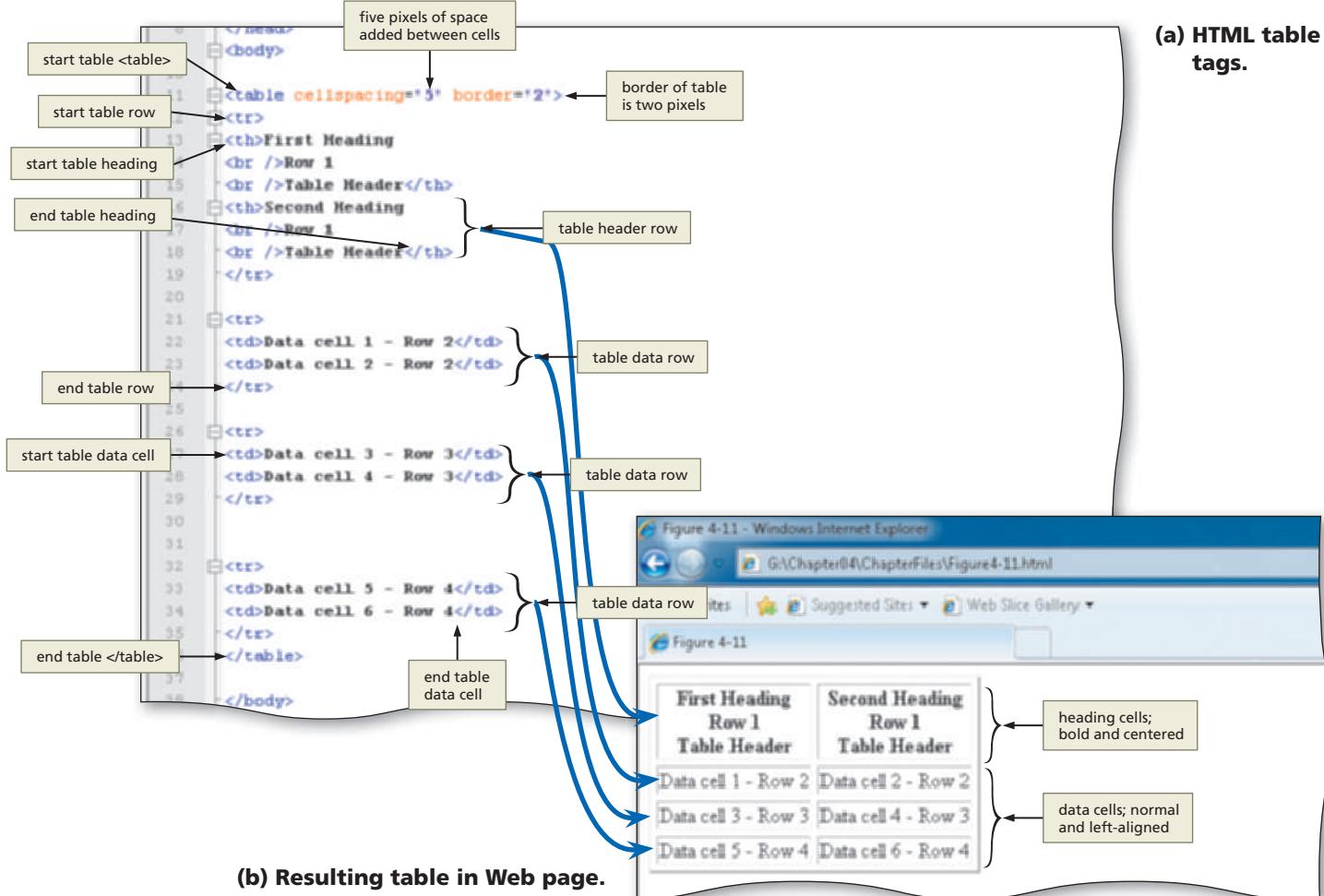
**Figure 4-11**

Table Borders

Table borders frame an image. You can insert a single image into a one-row, one-column table. Using a border gives the image a 3-D appearance, making the image appear to have a frame around it. A border of 1 pixel (`border="1"`) is too small to use as a frame, but `border="25"` is too large.

Table Tag Attributes

Each of the four main table tags listed in Table 4–1 on page HTML 169 has different attributes. Table 4–2 lists the table tags and the main attributes associated with each. The `<th>` and `<td>` tags, which are both used to specify the contents of a cell, have the same attributes. Many of the table tags' attributes listed in Table 4–2 are used in creating the Sabatina's Pizza Web site.

Table 4–2 Table Tag Attributes and Functions

Tag	Attribute	Function
<code><table> </table></code>	<code>border</code> <code>cellspacing</code> <code>cellpadding</code> <code>cols</code> <code>width</code>	<ul style="list-style-type: none"> Defines width of table border in pixels Defines space between cells in pixels Defines space between a cell's contents and its border in pixels Defines number of columns Sets table width relative to window width
<code><tr> </tr></code>	<code>valign</code>	<ul style="list-style-type: none"> Vertically aligns row (top, middle, bottom)
<code><th> </th></code> and <code><td> </td></code>	<code>colspan</code> <code>rowspan</code> <code>valign</code>	<ul style="list-style-type: none"> Sets number of columns spanned by a cell Sets number of rows spanned by a cell Vertically aligns cell (top, middle, bottom)

Plan Ahead

Identify the format of various Web page elements.

Before inserting tables or graphical elements in a Web page, you should plan how you want to format them. By effectively utilizing tables and graphics, you can better organize the most important topics on the Web page. Consider the following formatting suggestions.

- **Effectively utilize graphics.** An important part of Web development is the use of graphics to call attention to a Web page. Generally, companies utilize the same logo on their Web site as they use on print material associated with the company, such as business cards and letterheads. Using the same graphical image on all marketing materials, including the Web site, is a good way to provide a consistent visual image and brand message to customers. Colorful company logos can also add an attention-grabbing element to a Web page.
- **Format tables to organize Web page content.** Sometimes it is better to have no border around the table, while other times borders enhance the look of the table, depending on the content and purpose of the table. In this chapter, you will use both bordered and borderless tables. Another consideration is where to place the table (left-, right-, or center-aligned).
- **Determine what table formatting to use.** When using a table to organize text links, it is important to first decide how to format the table. Although you may not want to distract from the text links by creating a table with a heavy border, you need to separate the text links in such a way that they are easy to find. Creating a borderless table with separators (e.g., pipe symbols) between the text links helps organize but not distract from the links.
- **Identify what links are needed.** Each Web page in a multipage Web site should have a link back to the home page of the Web site. Web developers often use the company logo to link back to the home page. In this project, the logo is also the central image of the Web pages. Because of that, a better option might be to provide a text link called Home that visitors can use to return to the home page. Each Web page should include links to the other pages on the Web site. Putting these links in a table at the top of each Web page helps visitors navigate easily, and providing the navigation menu bar across all Web pages in the Web site is also important for consistency. If a Web page is very long, it also might be a good idea to put the same text link table at the bottom of the Web page. Again, the purpose of providing links is to make it easy to navigate the Web site.

Creating a Home Page with Banner Logo and Borderless Table

The first Web page developed in this chapter's project is the home page of the Sabatina's Pizza Web site. As you have learned, the home page is the main page of a Web site and is what Web site visitors generally view first. Visitors then click links to move from the home page to the other Web pages in the site. The Sabatina's Pizza home page includes the company logo as a banner image and a borderless table that contains three links to other pages: the Appetizers Web page, the Salads Web page, and the Pizza Web page. The home page also provides an e-mail link, so visitors can contact Sabatina's Pizza easily.

To Start Notepad++

The first step in creating the Sabatina's Pizza Web site is to start Notepad++ and ensure that word wrap is enabled. The following steps, which assume Windows 7 is running, start Notepad++ based on a typical installation. You may need to ask your instructor how to start Notepad++ for your computer.

- 1** Click the Start button on the Windows taskbar to display the Start menu.
- 2** Click All Programs at the bottom of the left pane on the Start menu to display the All Programs list.
- 3** Click Notepad++ in the All Programs list.
- 4** Click Notepad++ in the list to display the Notepad++ window.
- 5** If the Notepad++ window is not maximized, click the Maximize button on the Notepad++ title bar to maximize it.
- 6** Click View on the menu bar.
- 7** If the Word wrap command does not have a check mark next to it, click Word wrap.

To Enter Initial HTML Tags to Define the Web Page Structure

Just as you did in Chapters 2 and 3, you start your file with the initial HTML tags that define the structure of the Web page. Table 4–3 contains the tags and text for this task.

Table 4–3 Initial HTML Tags

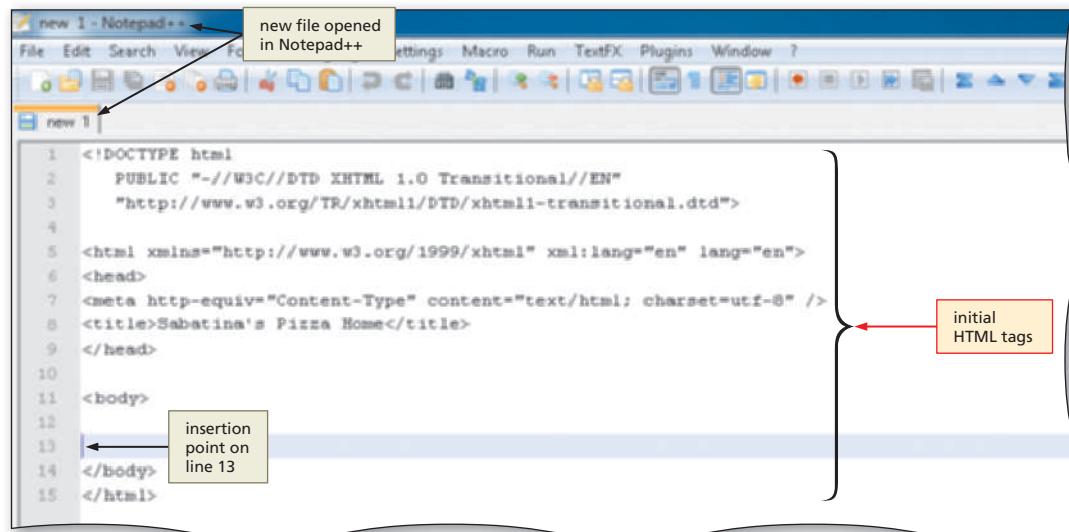
Line	HTML Tag and Text
1	<!DOCTYPE html
2	PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3	"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4	
5	<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
6	<head>
7	<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
8	<title>Sabatina's Pizza Home</title>
9	</head>
10	

Table 4–3 Initial HTML Tags (continued)

Line	HTML Tag and Text
11	<body>
12	
13	</body>
14	</html>

The following steps illustrate how to enter the initial tags that define the structure of the Web page.

- 1 Enter the HTML code shown in Table 4–3. Press ENTER at the end of each line. If you make an error as you are typing, use the BACKSPACE key to delete all the characters back to and including the incorrect characters, then continue typing.
- 2 Position the insertion point on the blank line between the <body> and </body> tags (line 12) and press the ENTER key to position the insertion point on line 13 (Figure 4–12).
- 3 Compare what you typed to Figure 4–12. If you notice errors, use your mouse pointer or ARROW keys to move the insertion point to the right of each error and use the BACKSPACE key to correct the error.

**Figure 4–12**

To Save an HTML File

With the initial HTML code for the Sabatina's Pizza home page entered, you should save the file. Saving the file frequently ensures you won't lose your work. Saving a file in Notepad++ also adds color to code that can help you identify different elements more easily. The following steps illustrate how to save an HTML file in Notepad++.

- 1 With a USB flash drive connected to one of the computer's USB ports, click File on the Notepad++ menu bar and then click Save.
- 2 Type `sabatinas.html` in the File name text box (do not press ENTER).
- 3 Click Computer in the left pane of the Save As dialog box to display a list of available drives.
- 4 If necessary, scroll until UDISK 2.0 (G:) or the name of your storage device is displayed in the list of available drives.

- 5** Open the Chapter04\ChapterFiles folder.
- 6** Click the Save button in the Save As dialog box to save the file on the USB flash drive in the Chapter04\ChapterFiles folder with the name sabatinas.html.

To Insert and Center an Image

The first task for the Sabatina's Pizza home page is to insert the company logo banner, sabatinaslogo.jpg. As stated earlier in the book, the company logo is generally used in all communication that represents the company, including on the Web site, on business cards, and on company letterheads. Table 4–4 contains the HTML code to add the centered logo banner image.

Table 4–4 HTML Code for Adding and Centering an Image

Line	HTML Tag and Text
13	<div style="text-align: center">
14	
15	</div>

The following step shows how to add a centered banner image.

1

- With the insertion point on line 13, enter the HTML code shown in Table 4–4, pressing ENTER at the end of each line. Make sure to indent the second line of code by using the TAB key. This separates the start and end <div> tag from the tag, highlighting the image insertion. Press the ENTER key twice at the end of line 15 to position the insertion point on line 17 (Figure 4–13).

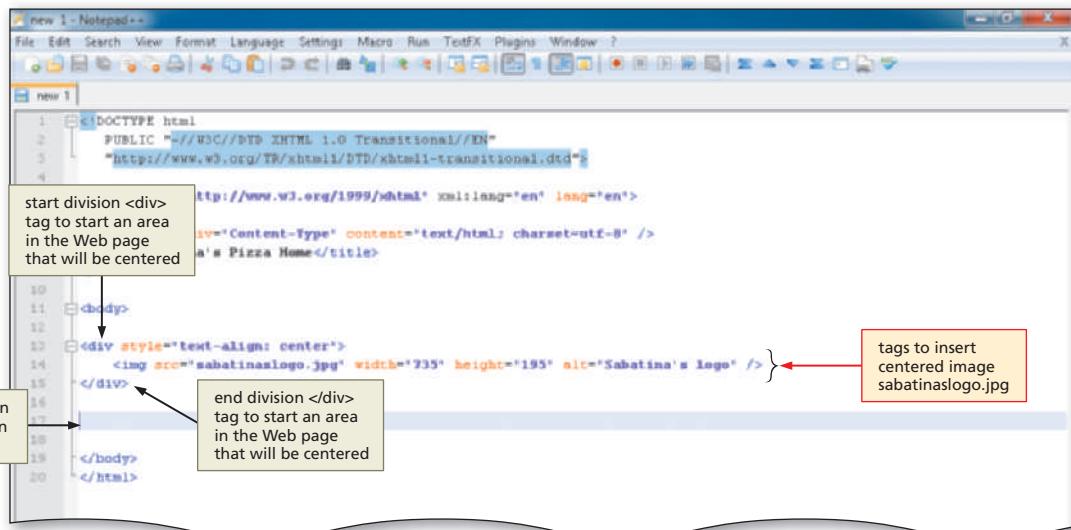


Figure 4–13

Q&A When I pressed ENTER at the end of line 14, Notepad++ indented line 15 also. How do I remove the indent?

You have to press the left arrow key or Backspace to get back to the left margin before you insert the </div> tag.

Q&A How can I determine the height and width of an image?

You can determine the height and width of an image by opening the image in a paint or image-editing program. Once you know the height and width, you also can adjust the width and height by using the width and height attributes in the tag. Be aware that, in doing so, you might cause the image to look distorted on the Web page.

Q&A Why can't I just put the tag within the <center></center> container to center this image?

If you look up the <center> tag in Appendix A, you see that it is deprecated. This book uses Cascading Style Sheets and the <div> </div> container rather than deprecated tags.

Using a Table to Create a Horizontal Menu Bar

The Web site created in this project consists of four Web pages. Visitors should be able to move easily from one Web page to any of the other three Web pages. Providing a menu bar prominently across the top of the Web page (Figure 4–14) gives the visitor ready access to navigation links. You will create a table to hold the menu bar links.



Figure 4–14

BTW

Navigation

Studies have been conducted to assess the best location on a Web page to place navigation bars and lists. The research results are varied, with indications that navigation options on the top, side, and bottom of a Web page show slight differences in visitor usability. The most important aspect of Web page navigation is to make the options easy enough to locate so visitors do not have to search for them.

All of the styles for this and other tables are set in the external style sheet that you will create in the next section of this chapter. The horizontal menu bar table is borderless (no border property is used), and has one row and seven columns (each divider is a column, as well as the text links). To better align the menu bar with the Sabatina's Pizza logo, the table is set to 60% of the window's width, so that it is not as wide as the logo. The menu bar has four links — Home, Appetizers, Salads, and Pizza — that link to the Web pages `sabatinas.html`, `appetizers.html`, `salads.html`, and `pizza.html`, respectively. Each link is inserted in a single column (cell). The | (pipe) symbol is included in a column between each of the four links to separate them visually.

The width of each column (cell) in the table is specified in the external style sheet, which will be linked to the home page and all other Web pages in this Web site later in the chapter. (If you do not define the width for these cells, the width defaults to the size of the word or symbol in the cell.) Classes that are used in the home page and in the other three Web pages are also defined in the external style sheet. The menuword and menupipe classes need to be inserted in the menu bar in order to adopt the styles as defined in the external style sheet. The class menuword will be used to set each of the cells in which there are words (i.e., Home, Appetizers, Salads, and Pizza) to 23% of the width of the table. The menupipe class will be used to set each of the cells in which there are pipe symbols to 1% of the size of the table. When you specify sizes in percentages, it is generally best not to set the entire 100% width. The menuwordselect class is used to highlight the link to the page being viewed in the browser. For example, when you click the link to the Appetizers Web page, the word Appetizers in the menu bar is bold and italicized. Each Web page uses the menuwordselect class for one link word.

Just as you did in Chapter 3, you add the class names to your Web pages before you create the classes, in this case, in the external style sheet. All menu bar styles are controlled with the external style sheet. If you didn't use an external style sheet, you would have to type each style into each menu bar on each Web page in the Web site. The advantage of using external (linked) style sheets is that if you need to make a change across the entire Web site,

you make that change only once — in the external style sheet. The change then takes effect in every Web page into which the external (linked) style sheet has been linked. Using an external style sheet, you create the .css file once and link it with one line of code into each Web page.

To Create a Horizontal Menu Bar with Text Links

Table 4–5 shows the HTML code for the horizontal menu bar table.

Table 4–5 HTML Code to Insert a Menu Bar

Line	HTML Tag and Text
17	<table class="menu">
18	<tr>
19	<td class="menuwordselect">Home</td>
20	<td class="menupipe"> </td>
21	<td class="menuword">Appetizers</td>
22	<td class="menupipe"> </td>
23	<td class="menuword">Salads</td>
24	<td class="menupipe"> </td>
25	<td class="menuword">Pizza</td>
26	</tr>
27	</table>

The following step shows how to create a table that contains text links to four pages on the Web site, separated by pipe symbols. The pipe symbol is usually found above the ENTER key; it is inserted when you press Shift and the \ (backslash) key.

1

- With the insertion point on line 17, enter the HTML code, as shown in Table 4–5, pressing ENTER after each line. Use the TAB key to indent the code, as shown in the table. Press the ENTER key once more after line 28 (Figure 4–15) to position the insertion point on line 29.

Q&A

Why indent my code with the TAB key?

Indenting is a good way to organize your code so sections are recognizable. You can immediately see what lines of

code are related to a single row in the table, as contained within the indented <tr> </tr> tags. This is helpful when you have many rows in the table, as in the pizza.html file.

```

<!DOCTYPE PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Sabatina's Pizza Home</title>
</head>
<body>
<div>

<table class="menu">
<tr>
<td class="menuwordselect"><a href="sabatinas.html">Home</a></td>
<td class="menupipe">|</td>
<td class="menuword"><a href="appetizers.html">Appetizers</a></td>
<td class="menupipe">|</td>
<td class="menuword"><a href="salads.html">Salads</a></td>
<td class="menupipe">|</td>
<td class="menuword"><a href="pizza.html">Pizza</a></td>
</tr>
</table>
</div>

```

Figure 4–15

To Add Paragraphs of Text

Next, two paragraphs of text must be added to the Web page. The text is displayed beneath the menu bar of links you just inserted. You use a paragraph `<p>` tag to insert a blank line between the menu bar and the text. Table 4–6 contains the code to add the paragraphs of text.

Table 4–6 HTML Code to Add Paragraphs of Text

Line	HTML Tag and Text
29	<code><p>Sabatina's Pizza recently opened in the Highlands. It has already become a popular place for family and friends to meet and have a great meal. Sabatina's offers many kinds of pizza, both thin and thick crust. They also have delicious appetizers and salads. There is ample seating on the inside of the restaurant and more seating on our lovely shaded patio outside. We have carry-out orders and also make deliveries.</p></code>
30	
31	<code><p>Please look through our Web site to learn more about our menu. Want to place a carry-out order? Call us at (999) 555-SABAS or e-mail us at sabatinas@isp.com.</p></code>

The following step illustrates how to add paragraphs of text.

- With the insertion point on line 29, enter the HTML code, as shown in Table 4–6, to insert the paragraphs of text, pressing the ENTER key after each line, including line 31 (Figure 4–16).

```

17 <table class="menu">
18   <tr>
19     <td class="menupipe"><a href="#">Home</a></td>
20     <td class="menupipe">|</td>
21     <td class="menupipe"><a href="appetizers.html">Appetizers</a></td>
22     <td class="menupipe">|</td>
23     <td class="menupipe"><a href="salads.html">Salads</a></td>
24     <td class="menupipe">|</td>
25     <td class="menupipe"><a href="pizza.html">Pizza</a></td>
26   </tr>
27 </table>
28
29 <p>Sabatina's Pizza recently opened in the Highlands. It has already become a popular place for family and friends to meet and have a great meal. Sabatina's offers many kinds of pizza, both thin and thick crust. They also have delicious appetizers and salads. There is ample seating on the inside of the restaurant and more seating on our lovely shaded patio outside. We have carry-out orders and also make deliveries.</p>
30
31 <p>Please look through our Web site to learn more about our menu. Want to place a carry-out order? Call us at (999) 555-SABAS or e-mail us at <a href="mailto:sabatinas@isp.com"><span style="text-decoration: underline">sabatinas@isp.com</span></a>.</p>
32
33
34

```

two paragraphs of text inserted

Figure 4–16

To Save the HTML File

With the HTML code for the Sabatina's Pizza home page complete, you should re-save the file. The following step shows how to save an HTML file that has been previously saved.

- 1** Click the Save icon on the Notepad++ toolbar to save the most recent version of sabatinas.html on the same storage device and in the same folder as the last time you saved it (Figure 4–17).

```

<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Sabatina's Pizza Home</title>
</head>
<body>
<div style="text-align: center">

</div>
<table class="menu">
<tr>
<td class="menupipe"><a href="sabatinas.html">Home</a></td>
<td class="menupipe"><a href="#">|</a></td>
<td class="menupipe"><a href="appetizers.html">Appetizers</a></td>
<td class="menupipe"><a href="#">|</a></td>
<td class="menupipe"><a href="#">|</a></td>
</tr>
</table>

```

Figure 4–17

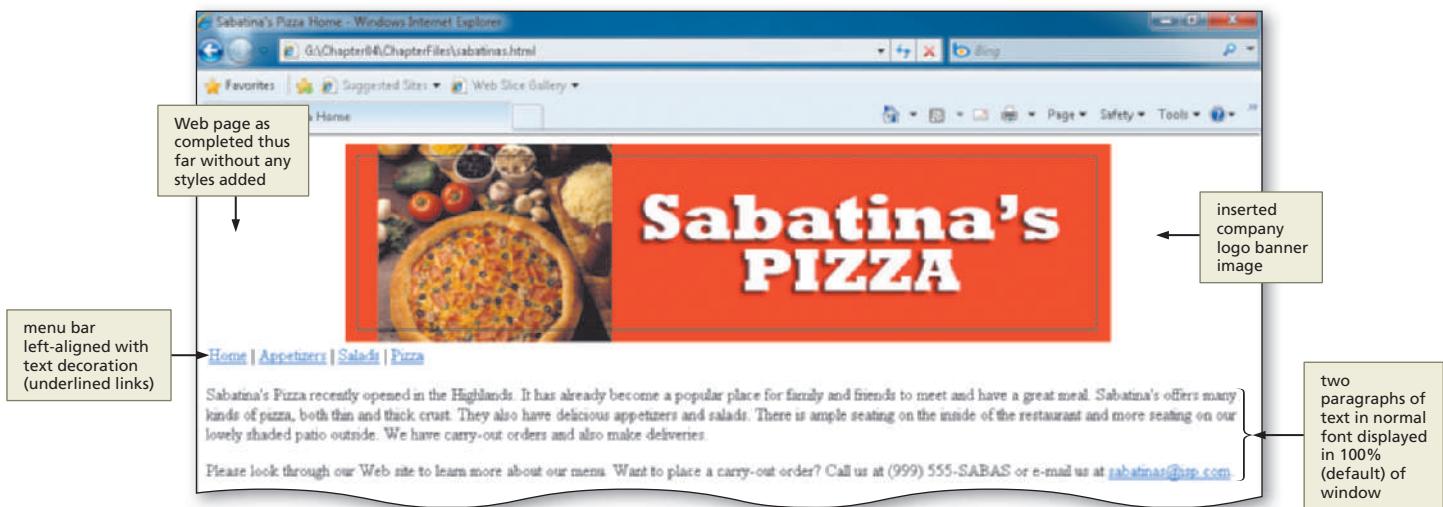
Viewing the Web Page and Testing Links

After you save the HTML file for the Sabatina's Pizza home page, it should be viewed in a browser to confirm the Web page appears as desired. You do not validate or print the Web page yet because you still have one statement (the link) to add to the file. It also is important to test the four links on the Sabatina's Pizza home page to verify they function as expected.

To View a Web Page

The following steps illustrate how to view the HTML file in a browser.

- In Internet Explorer, click the Address bar to select the URL on the Address bar.
- Type g:\Chapter04\ChapterFiles\sabatinas.html or the location of your file on the Address bar of your browser and press ENTER to display the Web page (Figure 4–18).

**Figure 4–18**

To Test Links on a Web Page

The following steps show how to test the links on Sabatina's Pizza home page to verify that they work correctly.

- 1** With Sabatina's home page displayed in the browser, point to the e-mail link, `sabatinas@isp.com` and click the link to open the default e-mail program with the address `sabatinas@isp.com` in the To: text box.
- 2** Click the Close button in the New Message window. If a dialog box asks if you want to save changes, click No.
- 3** With the USB flash drive in drive G, click the Salads link from the home page just created. Click back to the Home page from the Salads page. Next, click the Pizza link from the Home page. Click back to the Home page from the Pizza page. The link for the Appetizers page will not work because that Web page is not yet created; you will create it later in this chapter.

External Style Sheets

External style sheets are the most comprehensive type of style sheet and can be used to control the consistency and look of many Web pages within a Web site. Adding an external style sheet to a Web page involves a two-step process of creating an external style sheet and then linking this style sheet to the desired Web pages. The most beneficial feature of the external style sheet is that you can easily change the style (appearance) of all Web pages into which the style sheet is linked just by changing the external style sheet. For instance, the font-family and font-size for all four Web pages in this chapter's project are set in the external style sheet. If the owners of the Web site decide that they do not like the look (or style) of that font-family or font-size, you only have to make the change in one file — the external style sheet. Compare that process to having the font-family and font-size inserted into every single Web page in a Web site either with embedded style sheets or (worst case) inline styles. To make a change to all font-family and font-size styles, you would have to change those in every single place that those styles reside. External (linked) style sheets are the most efficient and powerful way to change styles for an entire Web site.

An external style sheet is a text file that contains the selectors and declarations for the styles you want to apply across the Web site. The sample code that follows shows an example of an external style sheet used to set table, paragraph, and link formatting. Note the use of classes (e.g., menu, menuword, menupipe) in this external style sheet that provides a finer level of control within a table used for navigation.

BTW

Classes

Note that the classes are named with a period (.) after the element is defined. Thus, the `table.menu` statement identifies a class named menu that will be used with the table elements.

```
table {width: 65%; margin-left:auto; margin-right:auto; font-family: Arial, Verdana, Garamond; font-size: 12 pt}

.menu {text-align: center; width: 60%}

.menuword {width: 23%}

.menupipe {width: 1%}
```

```
th      {color: white;  
         background-color: #ff1828}  
  
caption {caption-side: bottom;  
          font-style: italic}  
  
p      {margin-left: 50pt;  
         margin-right: 50pt;  
         font-family: Arial, Verdana, Garamond;  
         font-size: 12pt}  
  
a      {text-decoration: none;  
         font-family: Arial, Verdana, Garamond;  
         font-size: 12pt;  
         color: black}
```

The format of the external style sheet is very similar to the format of the embedded style sheet. An external style sheet, however, does not need `<style> </style>` tags to start and end the style sheet; it only needs the style statements.

To create an external style sheet, enter all of the style statements in a text file using Notepad++ or another text editor, and then save the text file with a .css (for Cascading Style Sheet) extension. The code shown above, for example, can be saved with the file name styles1.css and then linked onto multiple Web pages.

Remember that the `<head>` tag is used for a variety of purposes. The information contained within the `<head></head>` container of your HTML document provides information to browsers and search engines but is not displayed on the Web page itself. The following tags can be used within the `<head></head>` container: `<base>`, `<link>`, `<meta>`, `<script>`, `<style>`, and `<title>`. For each Web page to which you want to apply the styles in an external style sheet, a `<link />` tag similar to the sample code below must be inserted within the `<head></head>` tags of the Web page:

```
<link rel="stylesheet" type="text/css" href="styles1.css" />
```

The `<link />` tag indicates that the style sheet styles1.css should be applied to this Web page. The property-value statement `rel="stylesheet"` defines the relationship of the linked document (that is, it defines it as a style sheet). The property-value statement `type="text/css"` indicates the content and language used in the linked document. The property-value statement `href="styles1.css"` indicates the name and location of the linked style sheet, styles1.css. To apply this style sheet to other pages in the Web site, you would insert the same `<link />` tag between the `<head></head>` tags of each Web page.

**Plan
Ahead****Determine what type of style sheets or other formatting to use in your Web pages.**

- **Determine which styles will be common across the Web pages in the Web site.** Some things should be constant across Web pages in a Web site. For instance, if tables are used, their style is generally common or consistent throughout the Web site. A font-family style is also something that is generally common across all Web pages in a Web site. Consider using external (or linked) style sheets for styles that will encompass all Web pages in a multipage Web site. Because it is a four-page Web site, the project in this chapter is a perfect application for external (or linked) style sheets. You use an external style sheet in this project to set styles for the body, paragraphs, links, and some table styles.
- **Identify elements that need to differ from the style used across the Web site.**
 - Consider using embedded style sheets for Web pages in which the styles apply only to one particular Web page. You can use embedded style sheets when you want elements within one Web page to look similar to one another. In the project in Chapter 4, however, you do not use embedded style sheets.
 - Consider using inline styles for any style that is unique for a single element, such as when you want one word or paragraph on one Web page to look different than all others. In this project, you use inline styles to align images and vary some text.
 - Consider using formatting tag attributes to give style to a single element. In this project, you use the cellpadding and cellspacing attributes, and others.

Adding an External Style Sheet

The next step is to create an external style sheet (.css file) and link it to the Web pages where it will be used. The external (linked) style sheet will set the style for body text, paragraphs, links, and table format.

Because the font-family and font-size style is used for all text on all Web pages, you can put that style in the body element, identified on the external style sheet as body. All links (identified as *a* in the external style sheet) will be black through all states (normal, visited, and active) and use no text decoration (that is, the text links will not be underlined). Most of the tables will have a 65% width relative to the window. The exception is the menu bar table that you created earlier, which will be slightly smaller (60% width). Finally, you will use classes, as discussed in Chapter 3, to have more control over particular elements of the tables in the Web site. You use a class named menu for the menu bar links. You also want to separate the menu bar table cells with pipe symbols. To do that, you use classes named menuword and menupipe to specify the width of each of those cells. You also insert a class named menuwordselect, which is used later for the menu tables in the Web site. Note that the classes are named with a period (.) after the element is defined. Thus, the table.menu statement identifies a class named menu that will be used with the table elements.

**Plan
Ahead****Create and link an external style sheet.**

The external style sheet is the most powerful and lowest precedence style sheet. With this style sheet, you can easily create a common look across a Web site by creating the external (.css) style sheet and linking it onto all Web pages.

- **Create the .css file.** The first step is to create the external style sheet itself. This file, which contains all of the style statements that you want, has to be saved with a file name extension of .css. Make sure to store this file in the same folder as the other Web pages.
- **Link the external style sheet onto the Web pages.** The second step is to link the external style sheet (.css file) onto the Web pages where you want the styles to be applied. The link statement is placed between the `<head>` and `</head>` tags.

To Create an External Style Sheet

Table 4–7 shows the style statements for an external style sheet for the Sabatina’s Pizza Web site. To create an external style sheet, you open a new text file and enter CSS code for the style statements that define the Web page style. After coding the style statements, you save the file with the file extension .css to identify it as a CSS file.

Table 4–7 Code for an External Style Sheet

Line	CSS Properties and Values
1	body {font-family: Arial, Verdana, Garamond;
2	font-size: 12 pt}
3	
4	p {margin-left: 50pt;
5	margin-right: 50pt}
6	
7	a {text-decoration: none;
8	color: black}
9	
10	table {width: 65%;
11	margin-left:auto;
12	margin-right:auto}
13	
14	.menu {text-align: center;
15	width: 60%}
16	
17	.menuword {width: 23%}
18	
19	.menuwordselect {width: 23%; font-weight: bolder; font-style: italic}
20	
21	.menupipe {width: 1%}
22	
23	th {color: white;
24	background-color: #ff1828}
25	
26	caption {caption-side: bottom;
27	font-style: italic}

The following steps illustrate how to create and save an external style sheet.

1

- If necessary, click the Notepad++ button on the taskbar to display *sabatinas.html*. Click File on the menu bar and then click New.
- Enter the CSS code, as shown in Table 4-7, using the TAB key to align text, as shown (Figure 4-19).

Q&A

Why did this step open a new file but not close the old file that I had in Notepad++?

```

sabatinas.html
new 2
View Format Language Settings Macro Run TextFX Plugins Window ?
sabatinas.html
1 body (font-family: Arial, Verdana, Garamond;
2   font-size: 12 pt)
3
4 { p
5   (margin-left: 50pt;
6   margin-right: 50pt)
7
8 a (text-decoration: none;
9   color: black)
10
11 table (width: 65%;
12   margin-left: auto;
13   margin-right: auto)
14
15 .menu (text-align: center;
16   width: 60%)
17
18 .menuword (width: 23%)
19
20 .menuwordselect (width: 23%; font-weight: bolder; font-style: italic)
21
22 .menupipe (width: 1%)
23
24 th (color: white;
25   background-color: #ff1828)
26
27 caption (caption-side: bottom;
28   font-style: italic)
  
```

Figure 4-19

2

- With the USB drive plugged into your computer, click File on the menu bar and then click Save As. Type *styles1.css* in the File name text box. If necessary, navigate to the G:\Chapter04\ChapterFiles folder. Click the Save button in the Save As dialog box to save the file as *styles1.css*.
- Click the File menu, click Print on the File menu, and then click the Print button in the Print dialog box to print a hard copy of the style sheet (Figure 4-20).

```

body{font-family: Arial, Verdana, Garamond;
  font-size: 12 pt}

p {margin-left: 50pt;
  margin-right: 50pt}

a {text-decoration: none;
  color: black}

table{width: 65%;
  margin-left: auto;
  margin-right: auto}

.menu {text-align: center;
  width: 60%}

.menuword {width: 23%}

.menuwordselect {width: 23%; font-weight: bolder; font-style: italic}

.menupipe {width: 1%}

th {color: white;
  background-color: #ff1828}

caption {caption-side: bottom;
  font-style: italic}
  
```

Figure 4-20

Examining the External Style Sheet

Because the CSS code for the external style sheet is complex, a review is necessary to learn what it does. The CSS code that you entered and is shown in Table 4-7 on page HTML 181 defines a new style for four main elements on a Web page: body, paragraphs, links, and tables. It is a good idea (but not a requirement) to insert your styles in order in the external style sheet.

The first style statement on lines 1 and 2 is entered as:

```
body {font-family: Arial, Verdana, Garamond;
      font-size: 12 pt}
```

to change the font-family and font-size for the text throughout the Web site. You use the body element because you want these styles to apply to text across the Web site (Figure 4-19). If you wanted to apply one font-family or font-size to paragraphs of text and another font-family and font-size to the text in links, you would use the elements *p* and *a* rather than body to create those styles. This project uses three different font-family styles (Arial, Verdana, and Garamond) just in case the computer on which the Web page is viewed does not have the first (Arial) or second (Verdana) font-family. If the computer does not have any of the three font-families, then the normal (default browser) font-family is used.

The next styles are applied to the paragraph (*p*) element on lines 4 and 5:

```
p {margin-left: 50pt;
   margin-right: 50pt}
```

With this style, you are adding right and left margins that are 50 points wide. This pulls the paragraph text in 50 points both from the left and right. To see what that style statement does to the look of the home page for this Web site, look at Figure 4-18 on page HTML 177 compared to Figure 4-1a on page HTML 159.

Lines 7 and 8 define the styles for all links by using the link (*a*) element:

```
a {text-decoration: none;
   color: black}
```

This statement sets links to have no text-decoration (underlines) and makes all link states (normal, visited, and active) black in color. With a text-decoration setting of none, the browser will not display lines under any links. Setting the link color to black throughout eliminates the blue and purple (normal and visited) link colors that you would normally have. You can see in Figure 4-1a through 4-1d that users can tell what page they are on by the style of the link text in the menu bar; the name of the page they are visiting appears bold and italic in the menu bar. This is accomplished with the menuwordselect class that was defined in the styles1.css file that you just created. For instance, you can see that the style of the Home link (Figure 4-21 on the next page) when you are on the Home Web page is darker (bolder) and italicized. As you move from page to page, that bolder and italicized style moves to the respective page text link. For this feature, the menuwordselect class is inserted into the appropriate text link. The menuwordselect class is used once on each Web page.

BTW

Colors

To find the exact color, you can open the logo in a graphic image editing program and use one of the tools (such as the eye dropper tool) to click on the logo itself. If you then look at the color box, you should see the six-digit hexadecimal code for that color.

**Figure 4-21**

In this external style sheet, all table-related styles are inserted together from lines 10 through 27:

table	{width: 65%;
	margin-left:auto;
	margin-right:auto}
.menu	{text-align: center;
	width: 60%}
.menuword	{width: 23%}
.menuwordselect	{width: 23%; font-weight: bolder;
	font-style: italic}
.menupipe	{width: 1%}
th	{color: white;
	background-color: #ff1828}
caption	{caption-side: bottom;
	font-style: italic}

Lines 10 through 12 identify the general style for all tables on the Web pages. The width of all tables (with the exception of the menu bar) will be 65% of the browser window. You control the width of the menu bar table by naming a class called menu (remember this can be any arbitrary name), as shown in the code in the styles1.css Web page file. The margin:auto statements horizontally center the element table with respect to the edges of the window. The values used for each side are equal because of the auto designator.

On lines 14 through 21, beneath the table element, are four named classes: menu, menuword, menuwordselect, and menupipe. Those lines are indented so that you can immediately see that they are related to the table element. On line 15, the style of the menu class is designated to be 60% rather than the 65% of all other tables. Line 14 tells the browser to display the text within the cells of the table in the center. See the differences created by these styles in Figure 4–22.

The menuword class sets the width of that column to 23%. The menupipe class sets the column with the pipe symbol to 1%. The menuwordselect class specifies a word to highlight by making it bolder and italicized. In Figure 4–22, the selected word is Appetizers. See how that word in the menu bar is bolder and italicized. You can utilize these classes (menu, menuword, menuwordselect, and menupipe) with the HTML code that you just typed in the sabatinas.html file. For the Sabatina's Pizza Home page, the menuwordselect class is used for the Home link (Figure 4–19, page HTML 182). When you click the Appetizers link though, the word Appetizers will be in bold and italics because you will use the menuwordselect class for that link. See the effect of that code in Figure 4–22.

In lines 23 and 24, you identify styles for all table headers (<th>). You designate a background color of #ff1828 (which is a red) with text color that is white (Figure 4–22). The red that is used is the same red that is in the Sabatina's Pizza logo banner.

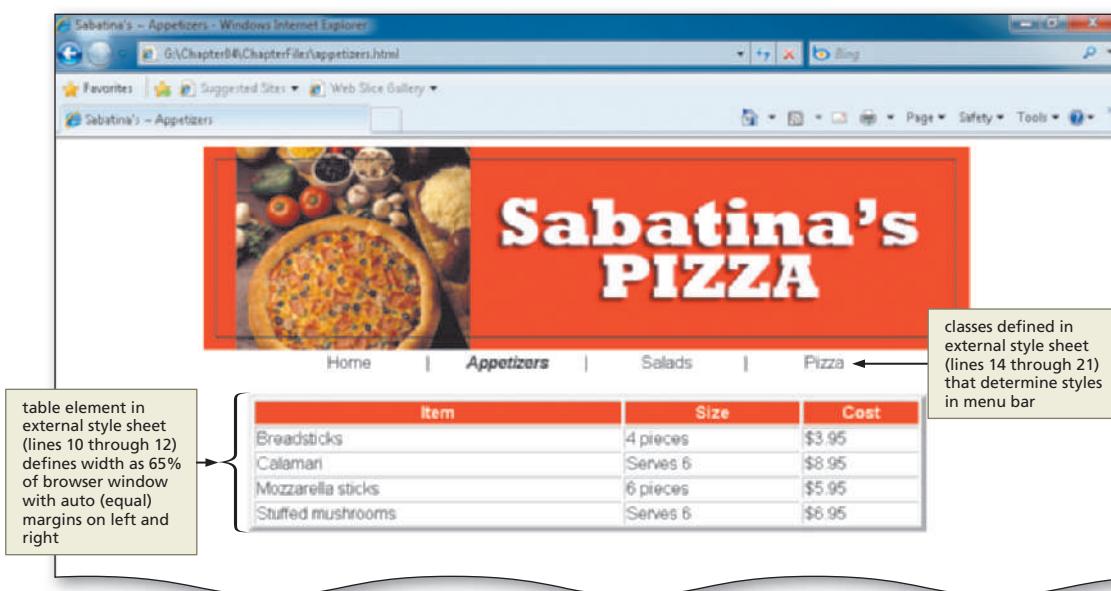


Figure 4–22

The final section of CSS code, lines 26 and 27, defines the styles to be applied to table captions. You want all captions to be aligned beneath the table and italicized.

Linking to the External Style Sheet

Four Web pages in the Sabatina's Pizza Web site require the same style: sabatinas.html, appetizers.html, salads.html, and pizza.html. Linking the external style sheet to each of these Web pages gives them the same styles for margins, paragraph text, links, and tables.

To link to the external style sheet, a `<link />` tag must be inserted onto each of these four Web pages. The `<link />` tag used to link an external style sheet is added within the `<head></head>` tag of the Web page HTML code. The general format of the `<link />` tag is:

```
<link rel="stylesheet" type="text/css" href="styles1.css" />
```

where `rel="stylesheet"` establishes that the linked document is a style sheet, `type="text/css"` indicates that the CSS language is used in the text file containing the style sheet, and `href="styles1.css"` provides the name and location (URL) of the linked style sheet. To link a style sheet to a Web page, the `<link />` tag must use "stylesheet" as the value for the `rel` property and `text/css` as the value for the `type` property. The URL used as the value for the `href` property varies, based on the name and location of the file used as the external style sheet. The URL used here indicates that the external style sheet, `styles1.css`, is located in the main or root directory of the Web site.

To Link to an External Style Sheet

After creating and saving the external style sheet, `.css` file, you use a `<link />` tag to link the external style sheet to any Web pages to which you want to apply the style. Notice that the link tag is one of those (like the `` tag) that does not have a separate end tag (e.g., `<body>` and `</body>`). You therefore should use the `/` after a space and before the `>` in the tag to indicate the end of the tag.

The following step illustrates how to add a link to an external style sheet using a `<link />` tag and then save the HTML file.

1

- Click the `sabatinas.html` tab in Notepad++ to make it the active window.
- With the `sabatinas.html` file open, click the end of line 8 after the `>` and press the ENTER key twice. Your insertion point should be positioned on line 10.
- Type `<link rel="stylesheet" type="text/css" href="styles1.css" />` as the HTML code and then press the ENTER key, as shown in Figure 4–23.
- Click the Save button on the Notepad++ toolbar.

```
1 <!DOCTYPE html
2 PUBLIC "-//IUC//DTD XHTML 1.0 Transitional//EN"
3 "http://www.w3.org/TR/xhtml/DTD/xhtml-transitional.dtd">
4
5 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
6 <head>
7 <meta http-equiv="Content-Type" content="text/html; charset=utE-8" />
8 <title>Sabatina's Pizza Home</title>
9
10 <link rel="stylesheet" type="text/css" href="styles1.css" /> ←
11
12 </head>
13
14 <body>
15
16 <div style="text-align: center">
17   
18 </div>
```

Figure 4–23

Q & A Is that all it takes to use an external style sheet — to insert that link statement?
 Yes, that is all you need to do to use the styles identified in the external style sheet. The styles specified in the external style sheet will apply to that page, unless an embedded or inline style sheet takes precedence. Remember too that if you want to change a style, you just change it in the external (linked) style sheet itself. It automatically takes effect in any Web page that is linked to the external style sheet. (Remember to click the Refresh button if that Web page is already open in the browser.)

Q & A Will the table styles from the styles1.css file take effect for all tables within the Web site?
 As long as you insert the style sheet link statement onto the Web page, then the table styles will take effect. Remember that you can override those styles with either an embedded or an inline style sheet. You would do this if there is a table that you want to vary from all other tables in the Web site.

Q & A Why is an external style sheet sometimes called a linked style sheet?
 The style sheet that you created above is external to (as opposed to inline or embedded in) the Web page in which it is used. That's why it is called an external style sheet. The external style sheet is used by linking it into a Web page. It is called linked because you use a `<link />` tag to insert it.

Validating and Printing the HTML, Viewing the Web Page, and Testing Links

After you save the HTML file for the Sabatina's Pizza home page, it should be validated to ensure that it meets current XHTML standards and viewed in a browser to confirm the Web page displays as desired. It also is important to test the links in the Sabatina's Pizza home page to verify that they function as expected.

To Validate a Web Page

The following steps illustrate how to validate an HTML file.

- 1 Open Internet Explorer and navigate to the Web site validator.w3.org.
- 2 Click the Validate by File Upload tab.
- 3 Click the Browse button.
- 4 Locate the `sabatinas.html` file on your storage device and click the file name.
- 5 Click the Open button in the Choose File to Upload dialog box and the file name will be inserted into the File box.
- 6 Click the Check button.

Q & A What if my HTML code does not pass the validation process?
 If your file does not pass validation, make changes to the file to correct your errors. You should then revalidate the file.

To Print an HTML File

After your HTML code has passed validation, it is a good idea to make a hard copy printout of it.

- 1** Click the Notepad++ button on the taskbar to activate the Notepad++ window.
- 2** Click File on the menu bar, click the Print command, and then click the Print button to print a hard copy of the HTML code (Figure 4–24).



The screenshot shows the Notepad++ code editor with the following details:

- File Path:** C:\ShellyCashman\HTML6\Chapter04\Solutions\ChapterFiles\sabatinas.html
- Date and Time:** Monday, December 07, 2009 11:44 AM
- Code Content:**

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Sabatina's Pizza Home</title>

<link rel="stylesheet" type="text/css" href="styles1.css" />

</head>

<body>

<div style="text-align: center">
    
</div>

<table class="menu">
<tr>
    <td class="menuwordselect"><a href="sabatinas.html">Home</a></td>
    <td class="menupipe">|</td>
    <td class="menuword"><a href="appetizers.html">Appetizers</a></td>
    <td class="menupipe">|</td>
    <td class="menuword"><a href="salads.html">Salads</a></td>
    <td class="menupipe">|</td>
    <td class="menuword"><a href="pizza.html">Pizza</a></td>
</tr>
</table>

<p>Sabatina's Pizza recently opened in the Highlands. It has already become a popular place for family and friends to meet and have a great meal. Sabatina's offers many kinds of pizza, both thin and thick crust. They also have delicious appetizers and salads. There is ample seating on the inside of the restaurant and more seating on our lovely shaded patio outside. We have carry-out orders and also make deliveries.</p>

<p>Please look through our Web site to learn more about our menu. Want to place a carry-out order? Call us at (999) 555-SABAS or e-mail us at <a href="mailto:sabatinas@isp.com"><span style="text-decoration: underline">sabatinas@isp.com</span></a>.</p>

</body>
</html>
```

Figure 4–24

To View, Test, and Print a Web Page

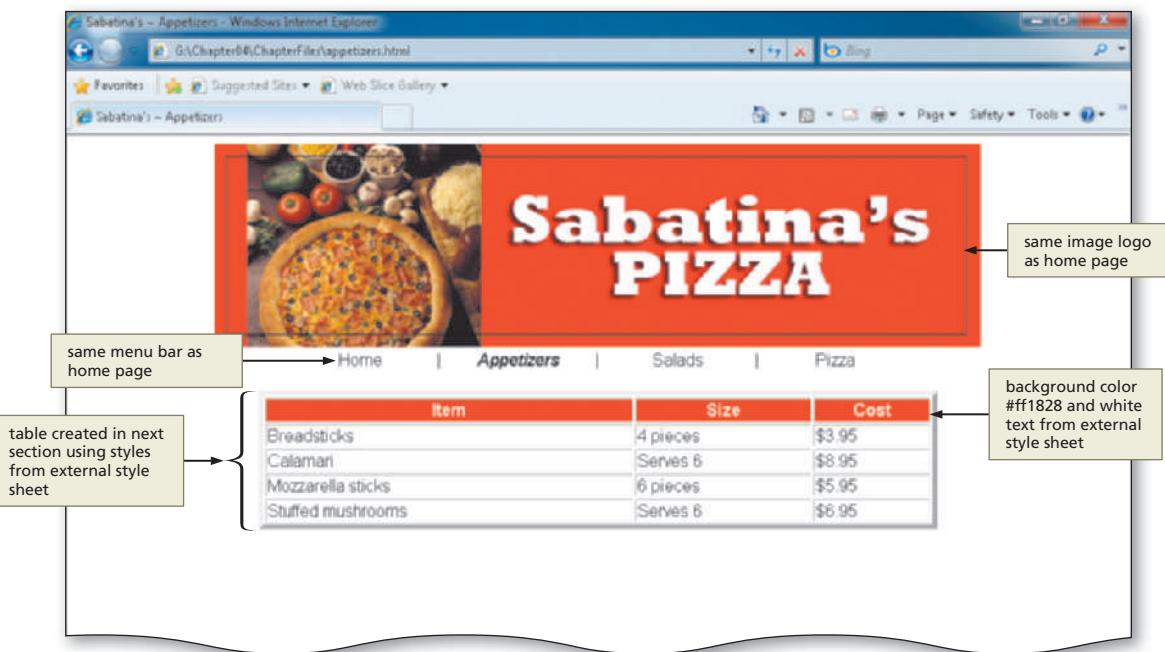
- 1** Click the Internet Explorer button on the Windows taskbar to activate Internet Explorer.
- 2** In Internet Explorer, click the Address bar to select the URL in the Address bar.
- 3** Type `g:\Chapter04\ChapterFiles\sabatinas.html` (or the specific path to your file) to display the new URL in the Address bar and then press the ENTER key.
- 4** Click the Salads and Pizza links to test that they work correctly.
- 5** Click the Print button on the Internet Explorer Command bar to print the Web page (Figure 4–25).



Figure 4–25

Creating a Second Web Page

You have created the Sabatina's Pizza home page with a horizontal menu bar of text links for easy navigation to other pages in the site and an external style sheet. Now it is time to create one of those linked pages — the Appetizers page (Figure 4–26 on the next page). Like the home page, the Appetizers page includes the logo image and a horizontal menu bar of text links. Having the Sabatina's Pizza logo and the horizontal menu bar at the top of each page provides consistency throughout the Web site. The menu bar lists the four Web pages — Home, Appetizers, Salads, and Pizza — with a | (pipe) symbol between links. Beneath the menu bar is a table listing the appetizers that are available at Sabatina's Pizza.

**Figure 4-26**

The first step in creating the Appetizers Web page is to add the HTML tags to define the Web page structure, the Sabatina's Pizza logo banner image, and the horizontal menu bar. Because the logo banner image and menu bar are the same as on the home page, you can copy and paste HTML code from the Home page and then edit it for the Appetizers page. You would copy/paste this code because you have already tested it by opening the `sabatinas.html` file in the browser, and you know the code works. Rather than retying the code, and possibly getting errors, a copy/paste will assure that it is correct.

To Copy and Paste HTML Code to a New File

The following step shows how to copy the HTML tags to define the Web page structure and the horizontal menu table from the HTML file, `sabatinas.html`, to a new HTML file.

- 1** If necessary, click the Notepad++ button on the taskbar and click the `sabatinas.html` tab to make it the active window.
- 2** Click immediately to the left of the `<` in the `<!DOCTYPE html>` tag on line 1.
- 3** Drag through the `</table>` tag on line 30 to highlight lines 1 through 30.
- 4** Press `CTRL+C` to copy the selected lines to the Clipboard.
- 5** Click File on the Notepad++ menu bar and then click New.
- 6** Press `CTRL+V` to paste the contents from the Clipboard into a new file. Press the `ENTER` key twice to position the insertion point on line 32.
- 7** Change the words `Pizza Home` in the `<title>` on line 8 to `~ Appetizers` (using the tilde `~` character).
- 8** Change the word `menuwordselect` on line 22 to `menuword`.
- 9** Change the word `menuword` on line 24 to `menuwordselect`.

To Save an HTML File

With the HTML code for the structure code and menu table added, the appetizers.html file should be saved.

- 1** With a USB drive plugged into the computer, click File on the menu bar and then click Save As. Type appetizers.html in the File name text box.
- 2** If necessary, click USB (G:) in the Save in list. Double-click the Chapter04 folder and then double-click the ChapterFiles folder in the list of available folders. Click the Save button in the Save As dialog box (Figure 4–27).

```

1 <!DOCTYPE html
2   PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
6   <head>
7     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
8     <title>Sabatina's ~ Appetizers</title>
9
10    <link rel="stylesheet" type="text/css" href="styles1.css" />
11
12  </head>
13
14 <body>
15
16   <div style="text-align: center">
17     
18   </div>
19
20   <table class="menu">
21     <tr>
22       <td class="menupipe"><a href="sabatinas.html">Home</a></td>
23       <td class="menupipe">|</td>
24       <td class="menuwordselected"><a href="appetizers.html">Appetizers</a></td>
25       <td class="menupipe">|</td>
26       <td class="menupipe"><a href="salads.html">Salads</a></td>
27       <td class="menupipe">|</td>
28       <td class="menuword"><a href="pizza.html">Pizza</a></td>
29     </tr>
30   </table>
31
32

```

HyperText Markup Language file nb char: 829 nb line: 32 Ln: 32 Col: 1 Sel: 0 Dos/Windows: ANSI INS

Figure 4–27

Determine what styles are needed for the second Web page.

Prior to adding more code to the Appetizers Web page, think through what styles you have defined and determine if there is anything in this Web page that needs to differ from the external style sheet. Any differences can be inserted as a formatting attribute, an inline style, or as an embedded style sheet that will override the styles defined in the external style sheet.

- **Review the table formatting in the external style sheet to see if it is appropriate for this page.** Borderless tables often are appropriate when the tables are used to position text and image elements. In other instances, such as when a table is used to structure columns and rows of information, borders are appropriate.
- **Identify any other styles that may need to be applied to tables on this page.** It is important to make your tables clear enough that users can easily identify the information that they need. You may want to start with the basic table format and add options as necessary.

Plan Ahead

Adding a Table with Borders

The borderless table style defined in the styles1.css external style sheet works well for the horizontal menu that appears on every page. However, the borderless format would be less effective in presenting the three columns and five rows of information about available appetizers at Sabatina's Pizza. Figure 4–28a shows this information in a table with borders. Figure 4–28b shows the same information in a table without borders. As shown in this figure, using a table with borders makes the information on the Appetizers Web page easier to read and provides a frame that gives the table a three-dimensional appearance.

(a) Table with borders.

Item	Size	Cost
Breadsticks	4 pieces	\$3.95
Calamari	Serves 6	\$8.95
Mozzarella sticks	6 pieces	\$5.95
Stuffed mushrooms	Serves 6	\$6.95

(b) Table with no borders.

Item	Size	Cost
Breadsticks	4 pieces	\$3.95
Calamari	Serves 6	\$8.95
Mozzarella sticks	6 pieces	\$5.95
Stuffed mushrooms	Serves 6	\$6.95

Figure 4–28

To Create a Table with Borders and Insert Text

Creating the table shown in Figure 4–28a involves first creating a table with three columns and five rows. The first row of the table is for column headings; the other rows are for data. As you have learned, table heading cells `<th>` differ from data cells `<td>` in their appearance. Text in a heading cell appears as bold and centered, while text in a data cell appears as normal and left-aligned. In the external style sheet `styles1.css` that you created in an earlier section of the project, you gave table header cells a background color of `#ff1828` and white text. Table 4–8 contains the HTML tags and text used to create the table of appetizers on the Appetizers Web page.

Table 4–8 HTML Code to Create a Table

Line	HTML Tag and Text
32	<code>
</code>
33	<code><table border="5"></code>
34	<code><tr></code>
35	<code> <th>Item</th></code>
36	<code> <th>Size</th></code>
37	<code> <th>Cost</th></code>
38	<code></tr></code>
39	
40	<code><tr></code>
41	<code> <td>Breadsticks</td></code>
42	<code> <td>4 pieces</td></code>
43	<code> <td>\$3.95</td></code>
44	<code></tr></code>
45	
46	<code><tr></code>
47	<code> <td>Calamari</td></code>
48	<code> <td>Serves 6</td></code>
49	<code> <td>\$8.95</td></code>
50	<code></tr></code>
51	
52	<code><tr></code>
53	<code> <td>Mozzarella sticks</td></code>
54	<code> <td>6 pieces</td></code>
55	<code> <td>\$5.95</td></code>
56	<code></tr></code>
57	
58	<code><tr></code>
59	<code> <td>Stuffed mushrooms</td></code>
60	<code> <td>Serves 6</td></code>
61	<code> <td>\$6.95</td></code>
62	<code></tr></code>
63	<code></table></code>
64	
65	<code></body></code>
66	<code></html></code>

The following step illustrates how to create a table with borders and insert text into heading and data cells.

1

- With the insertion point on line 32, enter the HTML code for the Appetizers table, as shown in Table 4–8, using TAB to create indents, and pressing ENTER after each line except the last line (Figure 4–29).

Q&A

What does the number in the table border line (line 33) represent?

It represents the number of pixels that you want the border to be. The higher the number, the wider the border.

Q&A

Are there other attributes that can be used in the `<table>`, `<tr>`, `<th>`, and `<td>` tags?

Many other attributes can be used in the `<table>` related tags (see Appendix A). The best way to review these attributes is to try them in simple tables to see the effect. Viewing tables with various attributes designated side by side on a Web page helps you to determine which attributes to use.

Q&A

Why are we using a white font color for the headings?

Because the background color is so dark (#ff1828), you could not easily read the heading if it was the default color of black. Changing the font color to white on a dark background color makes it easier to read.

```

<table border="1">
  <thead>
    <tr>
      <th>Item</th>
      <th>Size</th>
      <th>Cost</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Breadsticks</td>
      <td>4 pieces</td>
      <td>$3.95</td>
    </tr>
    <tr>
      <td>Calamari</td>
      <td>Serves 6</td>
      <td>$8.95</td>
    </tr>
    <tr>
      <td>Mozzarella sticks</td>
      <td>6 pieces</td>
      <td>$5.95</td>
    </tr>
    <tr>
      <td>Stuffed mushrooms</td>
      <td>Serves 6</td>
      <td>$6.95</td>
    </tr>
  </tbody>
</table>

```

Figure 4–29

To Save, Validate, View, and Print the Web Page

After adding the remaining HTML code, perform the following steps to save, validate, view, and print the Appetizers Web page.

- In Notepad++, click the Save icon on the toolbar to save appetizers.html.
- Click the Internet Explorer button on the taskbar.
- Use the W3C validator service to validate the appetizers.html Web page.
- Click the Notepad++ button to return to Notepad++.
- Print the appetizers.html Notepad++ file (Figure 4–30).
- Use the Back button or Internet Explorer tabs to return to the Sabatina's home page.
- Click the Appetizers link on the home page to show the most recent file.
- Click the Print button on the Command bar to print the Web page, as shown in Figure 4–31.

```

<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Sabatina's ~ Appetizers</title>

<link rel="stylesheet" type="text/css" href="styles1.css" />

</head>

<body>

<div style="text-align: center">

</div>

<table class="menu">
<tr>
<td class="menuword"><a href="sabatinas.html">Home</a></td>
<td class="menupipe">|</td>
<td class="menuwordselect"><a href="appetizers.html">Appetizers</a></td>
<td class="menupipe">|</td>
<td class="menuword"><a href="salads.html">Salads</a></td>
<td class="menupipe">|</td>
<td class="menuword"><a href="pizza.html">Pizza</a></td>
</tr>
</table>

<br />
<table border="5">
<tr>
<th>Item</th>
<th>Size</th>
<th>Cost</th>
</tr>

<tr>
<td>Breadsticks</td>
<td>4 pieces</td>
<td>$3.95</td>
</tr>

<tr>
<td>Calamari</td>
<td>Serves 6</td>
<td>$8.95</td>
</tr>

<tr>
<td>Mozzarella sticks</td>
<td>6 pieces</td>
<td>$5.95</td>
</tr>

<tr>
<td>Stuffed mushrooms</td>
<td>Serves 6</td>
<td>$6.95</td>
</tr>
</table>
</body>
</html>

```

Sabatina's ~ Appetizers

Page 1 of 1

Sabatina's PIZZA



Home | **Appetizers** | Salads | Pizza

Item	Size	Cost
Breadsticks	4 pieces	\$3.95
Calamari	Serves 6	\$8.95
Mozzarella sticks	6 pieces	\$5.95
Stuffed mushrooms	Serves 6	\$6.95

Figure 4-30

Figure 4-31

To Test Links on a Web Page

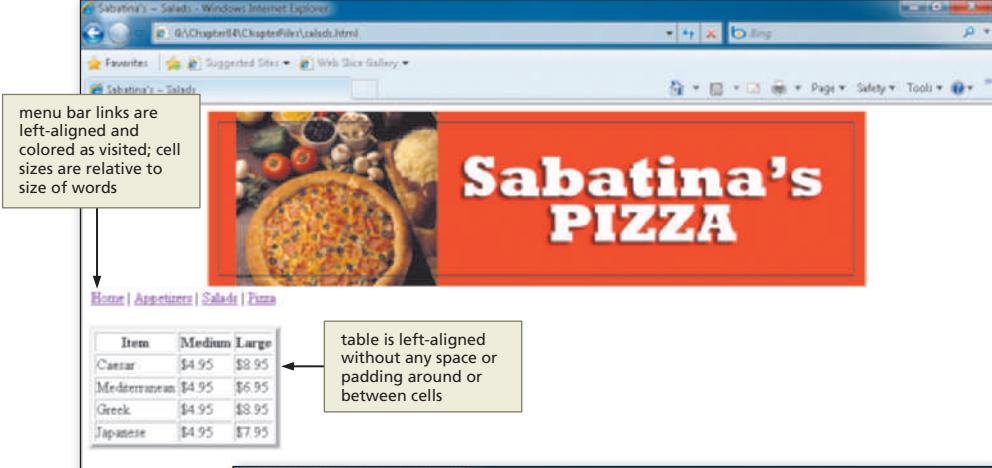
After confirming that the Web page appears as desired, the four links on the horizontal menu bar should be tested to verify that they function as expected. The following steps show how to test the links on the appetizers.html Web page. Compare Figure 4–32a (the starting Web page) to Figure 4–32b (the ending Web page after the HTML code is entered in the next section).

- 1** Click the Home link to change to the Sabatina's Pizza home page.
- 2** Click the Appetizers link to return to the appetizers.html Web page.
- 3** Click the Pizza link. (You will add a heading to the Pizza page later in the project.)
- 4** Click the Salads link (Figure 4–32a).

Q&A

Why isn't the Salads Web page formatted with the styles in the external style sheet?

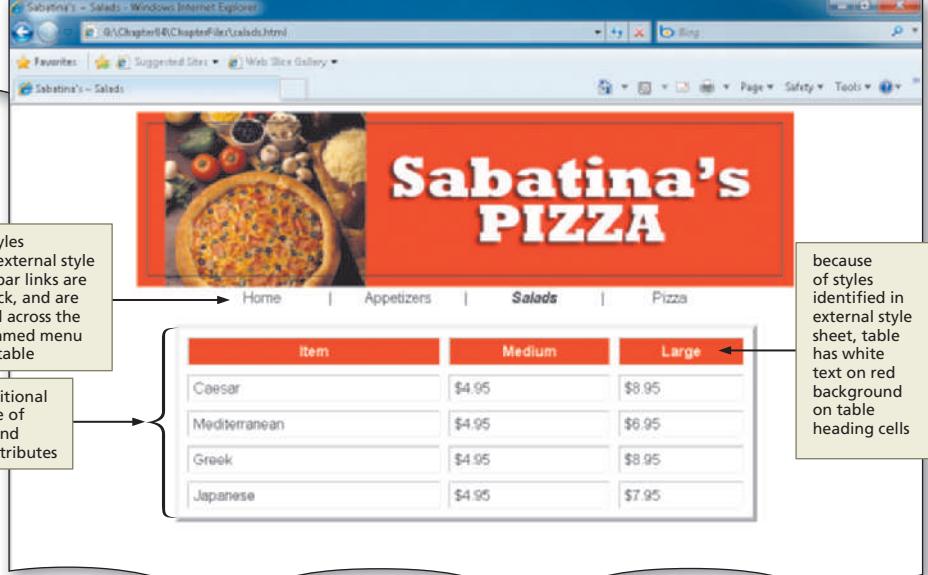
In order for the external style sheet to take effect, you have to insert the <link> statement into the HTML code in the Salads file.



menu bar links are left-aligned and colored as visited; cell sizes are relative to size of words

Item	Medium	Large
Caesar	\$4.95	\$8.95
Mediterranean	\$4.95	\$6.95
Greek	\$4.95	\$8.95
Japanese	\$4.95	\$7.95

(a) Tables with no external style sheet linked in.



because of styles identified in external style sheet, menu bar links are centered, black, and are evenly spaced across the table; class named menu used for this table

table has additional space because of cellpadding and cellspacing attributes

Item	Medium	Large
Caesar	\$4.95	\$8.95
Mediterranean	\$4.95	\$6.95
Greek	\$4.95	\$8.95
Japanese	\$4.95	\$7.95

because of styles identified in external style sheet, table has white text on red background on table heading cells

(b) Tables with external style sheet linked in.

Figure 4–32

Adding a Link, Cellspacing, and Cellpadding

The table of information on the Appetizers Web page did not use the cellspacing or cellpadding attributes. The size of each data cell, therefore, automatically was set to the minimum size needed for the text inserted in the data cell. The salads.html Web page, however, should be modified to use cellspacing and cellpadding by adding the cellspacing and cellpadding attributes to the <table> tag. **Cellspacing** defines the number of pixels of space between cells in a table. Additional cellspacing makes the borders around each cell look thicker (see the thickness of the borders in Figure 4–33b versus Figure 4–33a). Although there is no official default, browsers usually use a default of 2 for cellspacing. **Cellpadding** defines the number of pixels of space between a cell's contents and the cell wall. In other words, if you add cellpadding, you give more space around the content within that cell. The default for cellpadding is 1. Figures 4–33a and 4–33b illustrate how using the cellspacing and cellpadding <table> tag attributes can affect a table's appearance.

BTW

Cellspacing

The cellspacing attribute adds pixels between cells. The purpose of cellspacing is to add additional space in the separation between the cells, whether or not a border exists. With a border, cellspacing increases the size of the border between the cells. Without a border, cellspacing increases the amount of white space between the cells.

BTW

Cellpadding

The cellpadding attribute adds pixels within a cell border. The purpose of cellpadding is to keep the content within each cell from looking like it is too close to the content of another cell. Cellpadding will set a margin for the right, left, top, and bottom of the cell all at once with the specification of one tag.

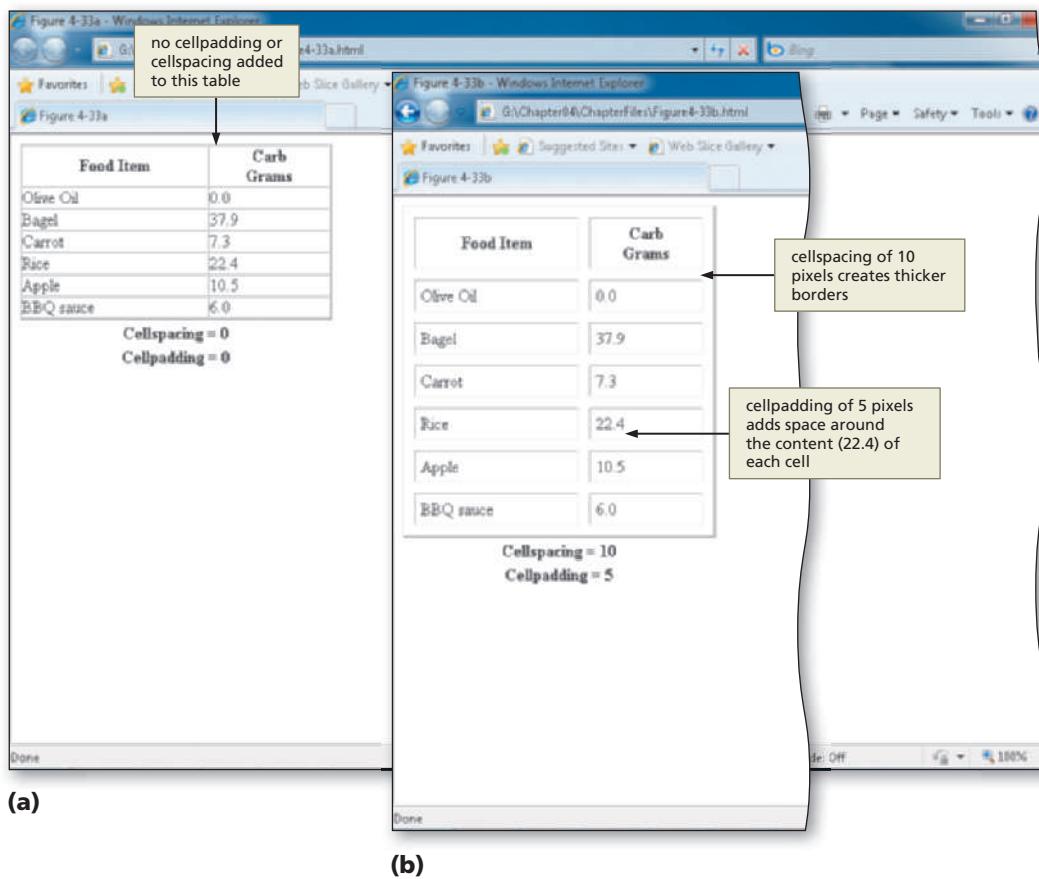


Figure 4–33 Tables with and without cellspacing and cellpadding.

**Plan
Ahead**

- **Determine if you need to use cellpadding, cellspacing, or both.** The first thing you should consider is if you need these spacing attributes at all. If your content is getting across to the users without any modification to the spacing, then maybe you do not need to do this. Look again at the appetizers.html Web page. The information there is completely readable, and there is no cellpadding or cellspacing.
- **Determine what table spacing to use.** If you decide that you do need to insert space within or around the cells of data, then you should next consider how much space to provide within the table. Cellspacing is the space between the borders of each cell. Cellpadding is the space between a cell's content and its border. Both attributes serve the purpose of making the table of information easier to read. No rule of thumb says how much cellpadding or cellspacing should be used. Try various values to see the effect on the table.

To Open an HTML File

In the following steps, you activate Notepad++ and open the salads.html Web page file.

- 1 Click the - Notepad++ button on the taskbar.
- 2 With a USB drive plugged into your computer, click File on the menu bar and then click Open on the File menu.
- 3 If necessary, navigate to the Chapter04\ChapterFiles folder on the USB drive.
- 4 Double-click salads.html in the list of files to open the file in Notepad++ (Figure 4–34).

```

<!DOCTYPE html
PUBLIC "-//IUC//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Sabatina's ~ Salads</title>
<!--Insert link statement here -->
</head>
<body>
<div style="text-align: center">

</div>
<table class="menu">
<tr>
<td class="menupword"><a href="sabatinas.html">Home</a></td>
<td class="menupipe">|</td>
<td class="menupword"><a href="appetizers.html">Appetizers</a></td>
<td class="menupipe">|</td>
<td class="menupwordselect"><a href="salads.html">Salads</a></td>
<td class="menupipe">|</td>
<td class="menupword"><a href="pizza.html">Pizza</a></td>
</tr>
</table>
<br />
<table border="5">

```

Figure 4–34

To Link to an External Style Sheet

The first thing you need to do in this Web page is link to the external style sheet that contains the overall styles you want this Web page to have.

- 1 Highlight the phrase <!--Insert link statement here --> on line 10.
- 2 Type `<link rel="stylesheet" type="text/css" href="styles1.css" />` to enter the link to the external style sheet. Do not press the ENTER key.

Q&A Remind me, what does this link statement do?

This statement links this Web page to the external style sheet that you created earlier. You need this link to the external (or linked) style sheet in order to apply those styles to the Web page.

To Add Cellspacing and Cellpadding to a Table

With the salads.html file open, the HTML code to add cellspacing and cellpadding can be added. The following step shows how to add cellspacing and cellpadding to a table.

1

- Click immediately to the right of the second " (quotation mark) in the border="5" statement on line 33 and then press the SPACEBAR.

- Type `cellpadding="5"` `cellspacing="10"` as the attributes and values but do not press the ENTER key (Figure 4-35).

Experiment

- Change the values in the cellpadding and cellspacing attributes to see what that does to the table. Try values that are smaller or much larger to see the effect.

```

10  <link rel="stylesheet" type="text/css" href="styles1.css" /> ← link to external style sheet, styles1.css, added on line 10; this is same link statement inserted into home page and will be added to other Web pages
11  </head>
12
13  <body>
14
15  <div style="text-align: center">
16    
17  </div>
18
19  <table class="menu">
20    <tr>
21      <td class="menupipe"><a href="#">Home</a></td>
22      <td class="menupipe">|</td>
23      <td class="menupipe"><a href="#">Appetizers</a></td>
24      <td class="menupipe">|</td>
25      <td class="menupipe"><a href="#">Salads</a></td>
26      <td class="menupipe">|</td>
27      <td class="menupipe"><a href="#">Pizza</a></td>
28    </tr>
29  </table>
30
31  <br />
32  <table border="5" cellpadding="5" cellspacing="10"> ← cellspacing attribute and value added on line 33
33    <tr> ← cellpadding attribute and value added on line 33
34      <th>Item</th>
35      <th>Medium</th>
36      <th>Large</th>
37    </tr>
38
39

```

Figure 4-35

Q&A What is the amount of cellpadding and cellspacing if I do not specify this in the table tag?

The default value for cellpadding is 1, while the default value for cellspacing is 2.

Q&A Can I set the cellpadding and cellspacing differently for different cells?

No, you cannot set cellpadding and cellspacing differently for various cells. This attribute is only available for the `<table>` (whole table) tag.

To Save, Validate, Print, and View the HTML File and Print the Web Page

- 1** With the USB drive plugged into your computer, click File on the menu bar and then click Save to save the salads.html file.
 - 2** Click the Internet Explorer button on the taskbar to display the Sabatina's ~ Salads page.
 - 3** Validate the Web page using the W3C validator service.
 - 4** Use the Back button or Internet Explorer tabs to return to the Sabatina's ~ Salads page.
 - 5** Click the Refresh icon on the Address bar to show the most recent file.
 - 6** Click the Print button on the Command bar to print the Web page.
 - 7** Click the Notepad++ button to return to Notepad++. Print the file.
-

BTW

Row and Column Spanning

Creating headings that span rows and columns defines tables more clearly. Many Web sites contain information about row and column spanning. For more information about row and column spanning, search the Web. Many tutorials have good suggestions for the use of column and row spanning.

Adding a Caption and Spanning Rows

If you need to add information to a table that does not fit into the table data cells, a caption can be a good option. For example, look at the pizza.html Web page in Figure 4–1d on page HTML 159. The caption indicates additional ingredients that are available for the pizzas listed in the table. That “additional ingredients” information does not fit into either the heading or data cells for the table. It is also not appropriate to have that information as an h1 or h2 heading. Putting that information in a caption is a perfect solution.

When you want to merge several cells into one, you can use row or column spanning. You can span rows or columns anywhere in a table. Generally, row and column spanning is used to create headings in tables. The **rowspan attribute** of the <th> or <td> tag sets a number of rows spanned by a cell. Although the chapter project uses only row spanning, you can also span columns. The **colspan attribute** of the <th> or <td> tag sets a number of columns spanned by a cell. Figure 4–10 on page HTML 168 shows examples of both column and row spanning. Notice that both the 5K and 10K headings span (or go across) two columns each. The heading Meet Dates spans (or goes across) four rows of information.

Figure 4–36 shows what the pizza.html Web page looks like at the start of the process. All of the table content is present, but there is no row or column spanning. You decide during the design phase that this table would benefit from row spanning, but that column spanning is unnecessary. You will enter the HTML code to complete the row spanning (Figure 4–37). In Figure 4–37, the heading Thin Crust Pizza is an example of row spanning. In this case, this heading spans three rows. In the same figure, the words Thick Crust Pizza also span three rows of information.

**Figure 4–36** Pizza Web page before enhancements.**Figure 4–37** Pizza Web page after enhancements.

The first step when deciding to span rows or columns is to sketch the table design on a piece of paper, as shown in Figure 4–38. Again, for this Web page, it was determined that column spanning was unnecessary. The table organizes pizza type by thin and thick crust and thus should have row spanning for those two main headings.

	<i>Cheese</i>	Medium - \$ 7.95	Large - \$ 9.95
Thin Crust Pizza	<i>Sausage</i>	Medium - \$ 8.95	Large - \$ 10.95
	<i>Veggie</i>	Medium - \$ 8.95	Large - \$ 10.95
	<i>Cheese</i>	Medium - \$ 9.95	Large - \$ 11.95
Thick Crust Pizza	<i>Sausage</i>	Medium - \$ 10.95	Large - \$ 12.95
	<i>Veggie</i>	Medium - \$ 10.95	Large - \$ 12.95

Figure 4–38

After defining the main sections of the table, you must determine how many rows each heading should span. For example, the first heading, Thin Crust Pizza, should span three rows. The second heading for Thick Crust Pizza should span three rows as well. In the following steps, you open the file pizza.html, link the external style sheet styles1.css, and add rowspan attributes to create table headings that span rows.

Plan Ahead

- **Determine if a caption is needed.** A caption can help clarify the table's purpose. For some tables, such as the table used to position images and the tables used to create menu bars, captions are not appropriate. Tables used to structure columns and rows of information, such as the pizza table, can benefit from having a caption to clarify or add information about the contents of the table. The caption tag must be inserted directly after the `<table>` tag.
- **Determine whether to use row and column spanning.** The purpose of the table determines whether you need to add row or column spanning. If the content is broken into logical segments of information, you may need to include row or column spanning in order to make the content clear. If you decide to add row or column spanning, it is best to sketch your ideas on paper first. This could help you understand more clearly what tags you need to use where.
- **Determine if different colors are needed for backgrounds.** You can help visitors more easily read a table full of information by varying the background colors effectively. If you use the same color background for the same level (or type) of information, it can help visually organize the information. Again, you may have to use a light font color if the background color is very dark.

To Open an HTML File

- 1 Click the Notepad++ button on the taskbar.
- 2 With the USB drive plugged into your computer, click File on the menu bar and then click Open on the File menu.
- 3 If necessary, navigate to the Chapter04\ChapterFiles folder on the USB drive.
- 4 Double-click pizza.html in the list of files to open the file in Notepad++.

To Link the External Style Sheet

The first thing you need to do in this Web page is link to the external style sheet that contains the styles that you want for this Web page.

- 1 Highlight the text <!--Insert link statement here --> on line 10.
- 2 Type `<link rel="stylesheet" type="text/css" href="styles1.css" />` to enter the link to the external style sheet. Do not press the ENTER key.

To Add a Table Caption

Captions are added to tables using the `<caption></caption>` tags to enclose the caption text. The formatting to make the caption italic and align it at the bottom of the table is included in the external style sheet (`styles1.css`) that is now linked to this page.

The following step shows how to add a caption below the pizza table.

1

- Highlight the text <!--Insert caption statement here --> on line 35.
- Type `<caption>Additional ingredients include mushrooms, olives, pepperoni - $2.00 per item</caption>` as the tag to add the italic caption below the table (Figure 4-39).



Experiment

- Add an inline style to the caption statement that aligns the caption-side to the top (top alignment for captions is the default). The inline style will override the external style sheet. Remember to take the inline style out once you are finished with the caption so that the bottom alignment default in the external style sheet takes effect again.

Q&A

Why would I use the caption tag? The caption further explains the main purpose of the table, or adds information that doesn't fit elsewhere in the table. Other uses are to identify units of measure or to give a number or title to the table.

```

10  <link rel="stylesheet" type="text/css" href="styles1.css" />
11
12 </head>
13
14 <body>
15
16 <div style="text-align: center">
17   
18 </div>
19
20 <table class="menu">
21   <tr>
22     <td class="menupipe"><a href="sabatinas.html">Home</a></td>
23     <td class="menupipe"><a href="appetizers.html">Appetizers</a></td>
24     <td class="menupipe"><a href="#">|</a></td>
25     <td class="menupipe"><a href="salads.html">Salads</a></td>
26     <td class="menupipe"><a href="#">|</a></td>
27     <td class="menupipe"><a href="pizza.html">Pizza</a></td>
28   </tr>
29 </table>
30
31 <br />
32 <table border="1" cellpadding="5" cellspacing="10">
33
34   <caption>Additional ingredients include mushrooms, olives, pepperoni - $2.00 per item</caption>
35
36   <tr>
37     <td>
38       <!--Insert first rowspan heading here -->
39       <td style="font-weight: bold">Cheese</td>
40       <td>Medium - $7.95</td>
41       <td>Large - $9.95</td>
42     </td>
43   </tr>
44 </table>
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91

```

Figure 4-39

To Create the Headings That Span Rows

The following steps illustrate how to enter HTML code to create two headings that each span three rows.

1

- Highlight <!--Insert first rowspan heading here --> on line 38.
- Press the TAB key twice and then type <th rowspan="3">Thin Crust and then press the ENTER key; the next line is automatically indented to the same level.
- Type
Pizza and then press the ENTER key.
- Type </th> as the HTML code but do not press the ENTER key (Figure 4–40).

```

<head>
  <link rel='stylesheet' type='text/css' href='styles1.css'>
</head>
<body>
  <div style='text-align: center;'>
    <img src='sabatinaslogo.jpg' width='735' height='195' alt='Sabatina's logo' />
  </div>
  <table class='menu'>
    <tr>
      <td class='menupipe'><a href='sabatinas.html'>Home</a></td>
      <td class='menupipe'><a href='appetizers.html'>Appetizers</a></td>
      <td class='menupipe'><a href='salads.html'>Salads</a></td>
      <td class='menupipe'><a href='pizza.html'>Pizza</a></td>
    </tr>
  </table>
  <br />
  <table border='1' cellpadding='5' cellspacing='10'>
    <caption>Additional ingredients include mushrooms, olives, pepperoni - $2.00 per item</caption>
    <tr>
      <th>
        <br />
        <th rowspan='3'>Thin Crust</th>
        <th>Pizza</th>
        <th>Cheese</th>
      </th>
    </tr>
  </table>
</body>
</html>

```

Figure 4–40

Q&A

What is the purpose of the
 tag in the steps above?

The
 tag moves the word Pizza to a second line so that the first column is not too much wider than the other columns in the table.

Q&A

Why are we not using the colspan attribute in this table?

The colspan attribute is used to add headings that span columns in a table. In this pizza table, column spanning is not necessary or appropriate. One column of information concerns the ingredients on the pizza, and the other two columns contain the pricing for medium and large pizzas. There is no column heading that would be appropriate to span those three columns.

2

- Highlight `<!--Insert second rowspan heading here -->` on line 59.
- Press the TAB key twice and then type `<th rowspan="3">Thick Crust` and then press the ENTER key.
- Type `
Pizza` and then press the ENTER key.
- Type `</th>` as the HTML code but do not press the ENTER key (Figure 4–41).

```

<tr>
  <th rowspan="3">Thin Crust
  <br />PIZZA
  </th>
  <td style="font-weight: bold">Cheese</td>
  <td>Medium - $9.95</td>
  <td>Large - $9.95</td>
</tr>

<tr>
  <td style="font-weight: bold">Sausage</td>
  <td>Medium - $8.95</td>
  <td>Large - $10.95</td>
</tr>

<tr>
  <td style="font-weight: bold">Veggie</td>
  <td>Medium - $8.95</td>
  <td>Large - $10.95</td>
</tr>

<tr>
  <th rowspan="3">Thick Crust
  <br />PIZZA
  </th>
  <td style="font-weight: bold">Cheese</td>
  <td>Medium - $9.95</td>
  <td>Large - $11.95</td>
</tr>

<tr>
  <td style="font-weight: bold">Sausage</td>
</tr>

```

Figure 4–41**Q&A**

Why isn't there an extra pair of `<tr> </tr>` tags between the rowspan title (Thin Crust Pizza) and the line of HTML code for data (Cheese)?

Thin Crust Pizza and Cheese are on the same row (row 1) that is being spanned. Row 2 contains the line with Sausage; row 3 is the line with Veggie. Therefore, the rowspan value is set to 3. The text that spans the rows (Thin Crust Pizza and Thick Crust Pizza) is always aligned with the first row of text.

Experiment

- Remove the `
` tag from the HTML code that you entered on lines 39 and 60. See how this changes the look of the table.
- Change the `
` tag on lines 39 and 60 to a `<p>` tag (don't forget the `</p>`). See what that does to the look of the table.

To Save, Validate, and Print the HTML File

- With the USB drive plugged into your computer, click the Save icon on the Notepad++ toolbar to save the `pizza.html` file.
- Validate the HTML file using the validator.w3.org Web page.
- Print the Notepad++ file.

To View and Print the Web Page

- 1 Click the Internet Explorer button on the taskbar.
- 2 Click the Pizza link on the menu table to display the Pizza Web page.
- 3 Print the Web page with rowspan attributes entered (Figure 4–42).



Figure 4–42

To Quit Notepad++ and a Browser

- 1 In Notepad++, click the File menu, then Close All.
- 2 Click the Close button on the Notepad++ title bar.
- 3 Click the Close button on the browser title bar. If necessary, click the Close all tabs button.

Chapter Summary

In this chapter, you learned how to create and link an external style sheet, about table elements, and the steps to plan, design, and code a table in HTML. You also learned to enhance a table with background color, cellspacing, cellpadding, a caption, and headers that span rows. The items listed below include all the new HTML skills you learned in this chapter.

1. Insert and Center an Image (HTML 173)
2. Create a Horizontal Menu Bar with Text Links (HTML 175)
3. Create an External Style Sheet (HTML 181)
4. Link to an External Style Sheet (HTML 186)
5. Copy and Paste HTML Code to a New File (HTML 190)
6. Create a Table with Borders and Insert Text (HTML 193)
7. Add Cellspacing and Cellpadding to a Table (HTML 199)
8. Add a Table Caption (HTML 203)
9. Create Headings that Span Rows (HTML 204)

Learn It Online

Test your knowledge of chapter content and key terms.

Instructions: To complete the Learn It Online exercises, start your browser, click the Address bar, and then enter the Web address scsite.com/html16e/learn. When the HTML Learn It Online page is displayed, click the link for the exercise you want to complete and read the instructions.

Chapter Reinforcement TF, MC, and SA

A series of true/false, multiple choice, and short answer questions that test your knowledge of the chapter content.

Flash Cards

An interactive learning environment where you identify chapter key terms associated with displayed definitions.

Practice Test

A series of multiple choice questions that test your knowledge of chapter content and key terms.

Who Wants To Be a Computer Genius?

An interactive game that challenges your knowledge of chapter content in the style of a television quiz show.

Wheel of Terms

An interactive game that challenges your knowledge of chapter key terms in the style of the television show, *Wheel of Fortune*.

Crossword Puzzle Challenge

A crossword puzzle that challenges your knowledge of key terms presented in the chapter.

Apply Your Knowledge

Reinforce the skills and apply the concepts you learned in this chapter.

Editing a Table on a Web Page

Instructions: Start Notepad++. Open the file apply4-1.html from the Chapter04\Apply folder of the Data Files for Students. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files.

The apply4-1.html file is a partially completed HTML file that you will use for this exercise. Figure 4–43 shows the Apply Your Knowledge Web page as it should display in a browser after the additional HTML tags and attributes are added.

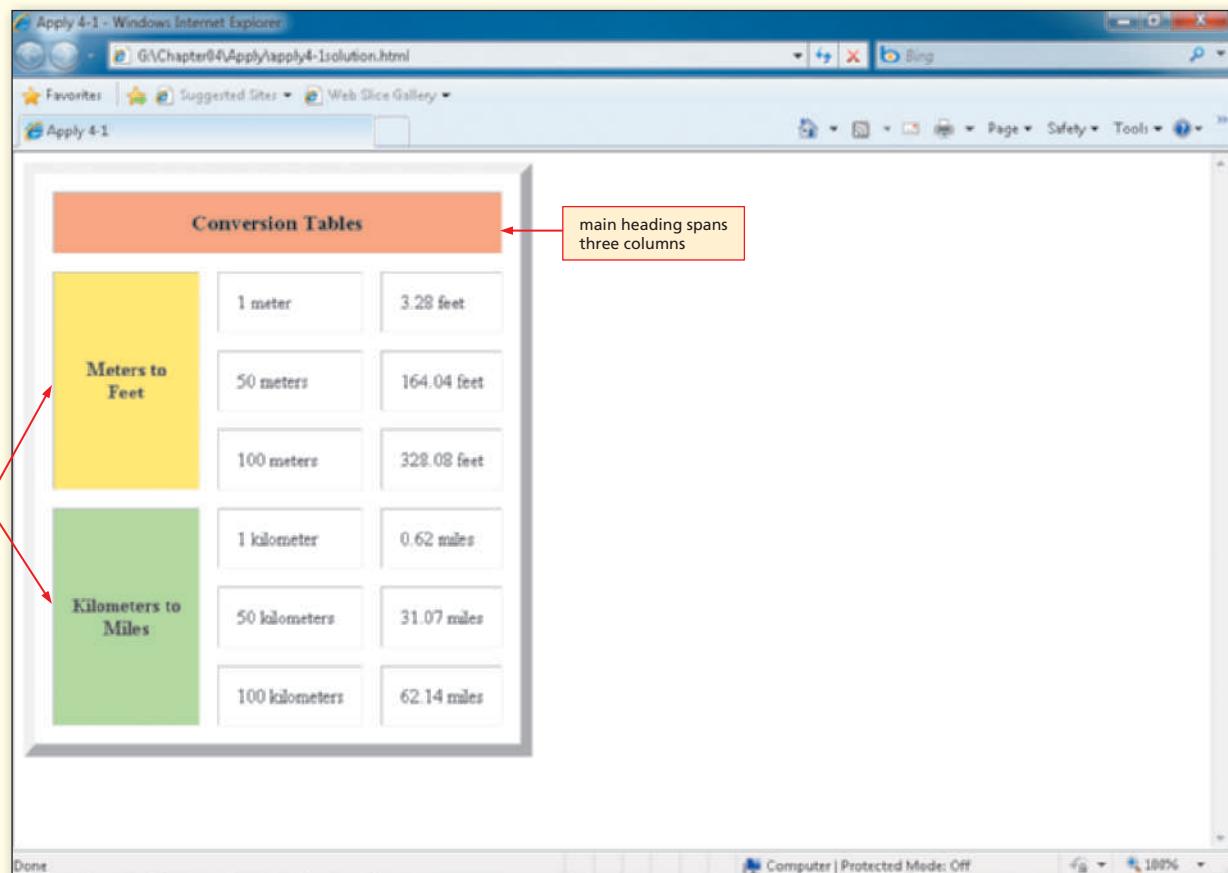


Figure 4–43

Perform the following tasks:

1. Enter the URL g:\Chapter04\Apply\apply4-1.html to view the Web page in your browser.
2. Examine the HTML file and its appearance as a Web page in the browser.
3. Add a border of 10, cellspacing of 15, and cellpadding of 15 to the table.
4. Add any HTML code necessary for additional features shown on the Web page in Figure 4–43. Your changes should include a colspan heading that spans three columns of information and two rowspan headings that span three rows each. The main heading is font-size large.
5. Colors used for the headings are colspan - #ff9473; first rowspan - #ffff6b; second rowspan - #a5de94 (experiment with the colors if you wish).
6. Save the revised file in the Chapter04\Apply folder using the file name apply4-1solution.html.
7. Validate the code using the W3C validator service.
8. Print the revised HTML file.
9. Enter the URL g:\Chapter04\Apply\apply4-1solution.html to view the Web page in your browser.
10. Print the Web page.
11. Submit the revised HTML file and Web page in the format specified by your instructor.

Extend Your Knowledge

Extend the skills you learned in this chapter and experiment with new skills.

Creating a Table with Rules

Instructions: Start Notepad++. Open the file extend4-1.html from the Chapter04\Extend folder of the Data Files for Students. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files. This sample HTML file contains all of the text for the Web page shown in Figure 4–44. You add the necessary tags to make this Web page display the table, as shown in Figure 4–44.

NURSE ON DUTY				
June 10 - June 16				
	Cheri Kimble	Elaine Clesi	Robert Skillman	Tyler Walters
1st floor North			X	
1st floor South		X		
2nd floor North	X			
2nd floor South				X
2nd floor Maternity				X

use rules attribute to show rules (borders) only between columns

border size is 12 and cellpadding is 15

Figure 4–44

Perform the following tasks:

1. Add HTML code to align the table on the left of the Web page. Align the text in the first column to the right of the data cell. (*Hint:* Review the text-align property.) Also give it a border of 12 with cellpadding of 15.
2. Insert the additional HTML code necessary to change the rules (see Appendix A) to only display columns.
3. Color the background #e0e0e0.
4. Add other table attributes not used in this chapter to further enhance the table. (*Hint:* See Appendix A.)
5. Save the revised document as extend4-1solution.html and validate the code using the W3C validator service.
6. Create an external style sheet that contains the following styles and save it as styleextend4-1.css.
7. Link styleextend4-1.css into the Web page extend4-1solution.html.

```
table      {text-align: center}
.twentyfive {width: 25%}
.fifteen   {width: 15%}
```

8. Re-save extend4-1solution.html. Print the revised HTML file and Web page and submit them in the format specified by your instructor.

Make It Right

Analyze a document and correct all errors and/or improve the design.

Correcting the Golf Course Tournament Schedule

Instructions: Start your browser. Open the file makeitright4-1.html from the Chapter04\MakeItRight folder of the Data Files for Students. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files. The Web page is a modified version of what you see in Figure 4–45. Make the necessary corrections to the Web page to make it look like Figure 4–45, using inline styles for all styles. The Web page should include the six columns of information with a main heading that spans all six columns. The second row contains the image golf.jpg in the first cell. The second row also has a line break between the person's first and last name. (*Hint:* Use the `
` tag.) Save the file as makeitright4-1solution.html and validate the code.

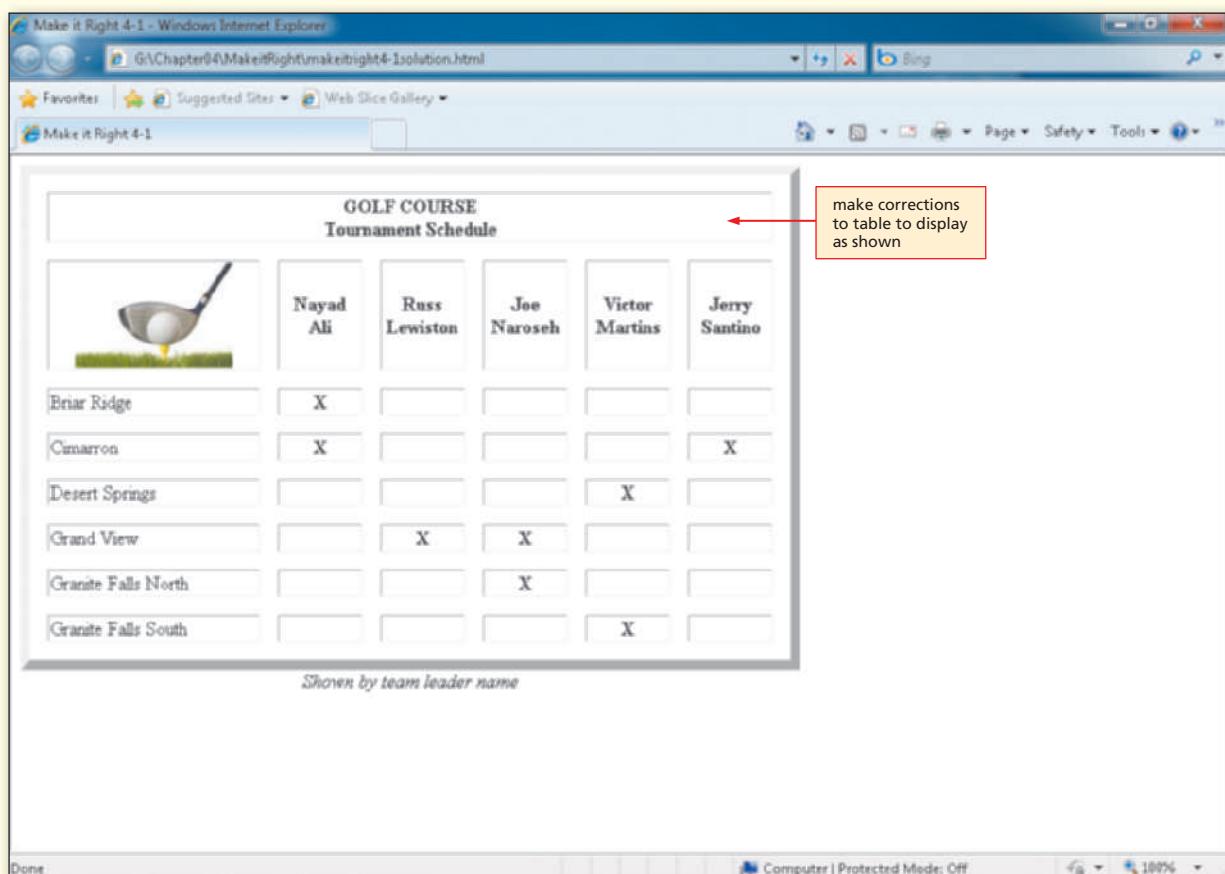


Figure 4–45

In the Lab

Lab 1: Creating a Table with Multiple Images

Problem: The owners of Beautiful Backyards (see Lab 3-2 at the end of Chapter 3) want to review the potential for the use of tables on their company home page and compare that with the Web page created in Lab 3-2. You have been asked to create a Web page that shows the two images and how tables can be used to display them and the associated text, as shown in Figure 4–46.

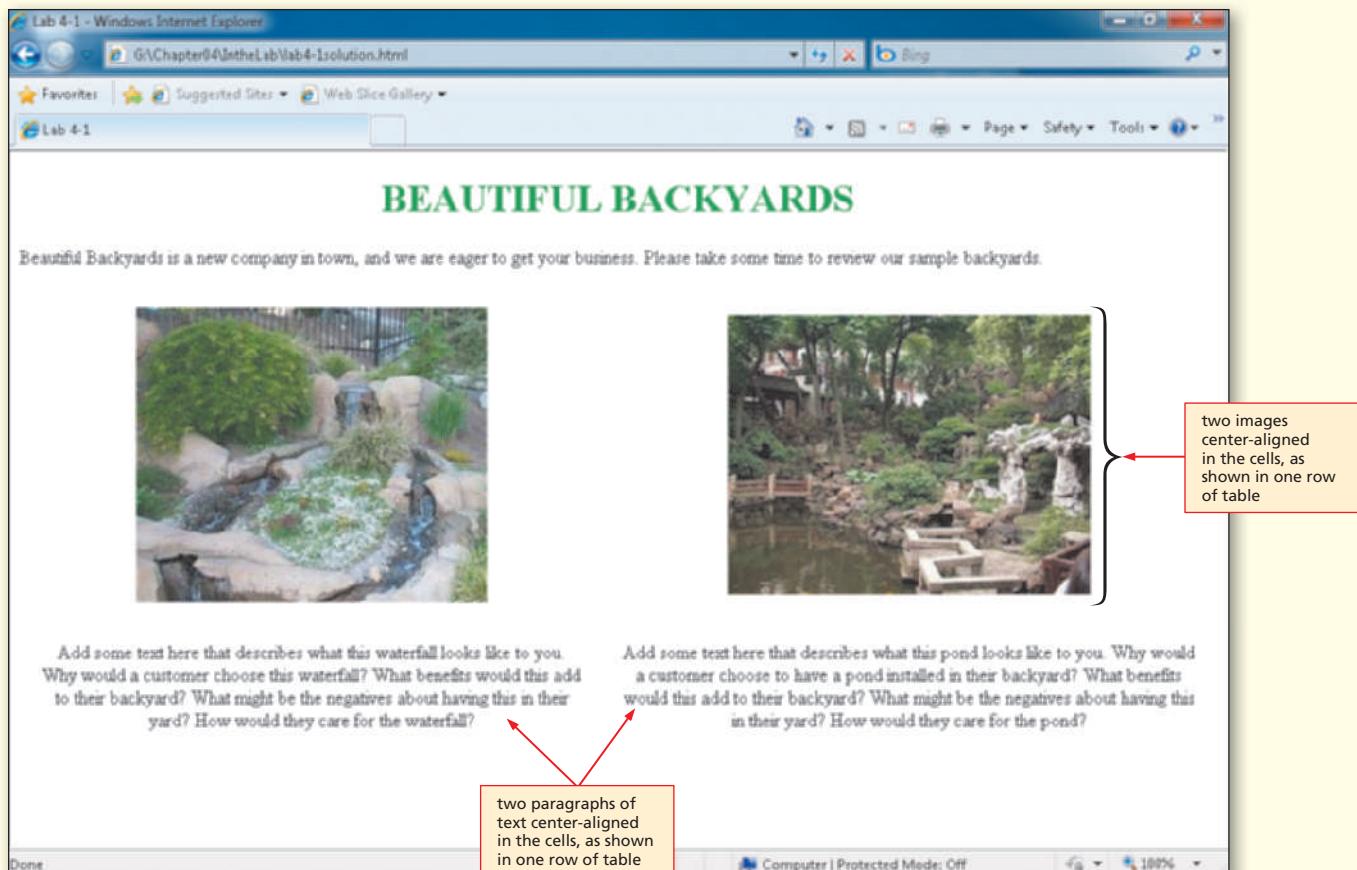


Figure 4–46

Instructions: Perform the following steps:

1. Start a new HTML file with the title Lab 4-1 in the main heading section.
2. Just as in Lab 3-2, the heading `<h1>` should be color #00934a.
3. Insert the text shown in the top lines of the Web page using a font size of large. (*Hint:* You can copy/paste the text from Lab 3-2.)
4. Add a centered borderless table with two columns and two rows and cellpadding of 15.
5. Insert the image largewaterfall.jpg in the first column of the first row. (*Hint:* You can use Microsoft Paint to determine the width and height of each image or right-click the image and select Properties.)
6. Add the second image pond.jpg to that same row in a second column.
7. Start a new row and add the text for each of the images.
8. Save the HTML file in the Chapter04\IntheLab folder using the file name lab4-1solution.html.
9. Validate the Web page using the W3C validator service.
10. Print the lab4-1solution.html file.
11. Open the lab4-1solution.html file in your browser to view it as a Web page.

Continued >

In the Lab *continued*

12. Print the Web page.
13. Submit the revised HTML file and Web page in the format specified by your instructor.

In the Lab

Lab 2: Creating Two Linked Pages

Problem: Your manager at Voytkovich Antiquities has asked you to create two Web pages, similar to the ones shown in Figures 4–47a and 4–47b. The first Web page is a home page that presents information about Voytkovich Antiquities, together with two links. The Prices link on the first page will be linked to a price list of items found at the antiquities store. The second link, called Links, should direct the Web page visitor to another Web page of your choosing that has to do with antiquities. You may select a museum or another similar site of your choosing. For this project, use a combination of inline style sheets, external (linked) style sheets, and formatting tag attributes in order to accomplish the tasks.

Instructions: Perform the following steps:

1. Start a new HTML file with the title Lab 4-2a in the main heading section.
2. Create a one-row, two-column borderless table with the image mask.jpg in the left-hand data cell and the words Voytkovich Antiquities (use the
 tag between those words) in an olive color and size xx-large in the right-hand data cell.
3. Create a second one-row, two-column borderless table. In the first column, include two text links named Prices (which links to lab4-2bsolution.html) and Links (which links to an antiquity Web site of your choosing). Include the text and an e-mail link, as shown in Figure 4–47a.
4. Create an external style sheet with the following styles. Save it as lab4-2styles.css. Link this external style sheet into the Web page.

```
body {font-family: Arial, Verdana, Garamond;  
      font-size: 11 pt}  
  
a {text-decoration: none;  
   color: black}  
  
table {width: 65%;  
       margin-left:auto;  
       margin-right:auto}  
  
.menu {text-align: left;  
        width: 20%}  
  
.content {width: 80%}
```

5. Save the HTML file using the file name lab4-2asolution.html in the Chapter04\IntheLab folder. Validate the file using the W3C validator service. Print the HTML file.
6. Start a new HTML file with the title Lab 4-2b in the main heading section. Link the external style sheet lab4-2styles.css into the file.
7. Create a five-row, two-column table with a five-pixel border, cellpadding of 15, and cellspacing of 5.
8. Span the first heading across both columns, as shown in Figure 4–47b, with olive text in size large.
9. Enter the headings, Item and Price, and additional information in the appropriate table cells, as shown in Figure 4–47b. Make sure to include a link (font size of small) back to the home page.
10. Save the HTML file in the Chapter04\IntheLab folder using the file name lab4-2bsolution.html. Validate the file using the W3C validator service. Print the HTML file.

11. Open the file lab4-2asolution.html in your browser and test the Prices link to verify it links to the lab4-2bsolution.html Web page.

12. Print both Web pages.

13. Submit the revised HTML file, .css file, and Web pages in the format specified by your instructor.

Voytkovich Antiquities

Voytkovich Antiquities is the leading wholesaler for Asian, Middle Eastern, and European antiquities. Items in our store include masks from a variety of cultures, statues, paintings, and other collectibles. Bring the past into your own homes with the beautiful art pieces offered in our store. Come visit us often to find new and unusual items not found elsewhere.

The links on the left take you to pricing information and our favorite Web sites. Please take a look at our prices and call or e-mail us for a price quote.

Please send any comments to voytkovich@sp.com.

Prices

Links

text links with background color in cell and no text-decoration

Voytkovich Antiquities

use olive color for font for table heading; span two columns

Back to home

link back to home is separate paragraph, not caption

(a)

(b)

Item	Price
Asian Mask	\$550.00
Egyptian statue	\$750.00
European painting	\$895.00

Figure 4–47

In the Lab

Lab 3: Creating Schedules

Problem: You want to create a Web page and an external style sheet that lists your piano practice and volunteer schedule, similar to the one shown in Figure 4–48. The Web page will use a table with images that span several rows and columns to organize the information.

Instructions: Perform the following steps:

1. Start a new HTML file with the title Lab 4-3 in the main heading section.
2. In the Web page, create a bordered table that displays in 90% of the browser in which only rows display, as shown in Figure 4–48. (*Hint:* See the rules attribute.)
3. Include the headings and data cells as shown, with valid information (i.e., real days and times as per a normal schedule of activities) in the data cells. The main headings should be center-aligned across three columns. The data other than the main headings should have an indent of 10 pixels.
4. Add two images, piano.jpg and volunteer.jpg, with all appropriate image attributes, each spanning five rows.
5. The external style sheet should contain the following styles and be saved as lab4-3styles.css:

```
body {font-family: Arial, Verdana, Garamond;  
      font-size: 11 pt}  
th.subtitle {text-align: left; text-indent: 10px}  
td {text-indent: 10px}
```

(*Note:* The <th> element aligns text left. Where would you use that class named subtitle?)

6. Link lab4-3styles.css to the HTML file, and save the HTML file as lab4-3solution.html.
7. Validate the HTML file using the W3C validator service.
8. Print the HTML and CSS files.
9. Print the Web page from your browser.
10. Submit the HTML file, .css file, and Web page in the format specified by your instructor.

PIANO PRACTICE SCHEDULE

Day of Week	Time
Day 1	Time 1
Day 2	Time 2
Day 3	Time 3
Day 4	Time 4

KID CAMP SCHEDULE

Day of Week	Time
Day 1	Time 1
Day 2	Time 2
Day 3	Time 3
Day 4	Time 4

Done Computer | Protected Mode: Off 100%

each image spans five rows

each title spans three columns

lab4-3styles.css external style sheet sets styles for Web page

Figure 4-48



Cases and Places

Apply your creative thinking and problem-solving skills to design and implement a solution.

- EASIER •• MORE DIFFICULT

• 1: Adding to the Sabatina's Pizza Web Site

In the Sabatina's Pizza project, you created an external style sheet that contained many styles for different Web page elements. Analyze what styles might be good to change or add, and determine a design plan for the changes. Make changes to the style sheet and save the changes with another file name (remember to give the file name a .css extension). Revise the link statement in one of the Sabatina's Pizza Web pages to use the new external style sheet name. Have other students evaluate the page that uses the new external style sheet, comparing it with the same page using the original external style sheet. If you like your changes and decide you want to make these changes throughout all pages, modify the other Sabatina's Web pages to include the link to the new .css file.

• 2: Finding Tables on the Web

The Dean of your school wants to update the Web pages for the school's Web site. She has asked your help in doing this and wants to see a proposal. You think that tables would provide the perfect format for displaying the various academic programs available in your school, potential class schedules, and a calendar of events. Browse the Web to find examples of tables used for information such as what is needed on your school's Web site. Print those pages so that you have concrete examples to show the Dean. Prepare a document that explains to the Dean how you would use such tables for your school's particular needs. Try using a storyboard (a graphical organizer that contains a series of illustrations or images displayed in sequence and that visually depicts your ideas). Sketch a Web page design (see Figure 4–9 on page HTML nn and Figure 4–40 on page HTML nn) that incorporates tables for your purpose.

•• 3: Researching Style Sheet Classes

Your manager at WebSource wants you to prepare a brief presentation on the use of classes in style sheets, as described in this chapter. He asks you to find at least two Web sites that describe the use of classes in style sheets and then review how the techniques discussed in the Web sites compare to the style sheet methods described and used in this chapter. The presentation also should discuss how the use of classes can help make Web development more effective — both in developing a single Web page and in developing an entire Web site.

•• 4 Creating a Time Schedule

Make It Personal

Your computer club wants you to create a table that lists meeting, open lab, and lab class times for the computer labs. Sketch a basic table format to use for this purpose and ask a few friends (or classmates) what they think. Once you have determined a good design for the Web page, begin to code the table needed. As you begin to build the Web page, you start thinking about other table attributes that could make the Web pages look even better. Create a Web page with a basic five-row, two-column table with a one-pixel border. Review the additional table attributes listed in Appendix A, including the rules attribute. Find information on those attributes on other Web sites, including the W3C Web site (www.w3.org). Modify the basic table on your Web page to incorporate at least four of these attributes.

•• 5 Creating a Gift Shop Web Site

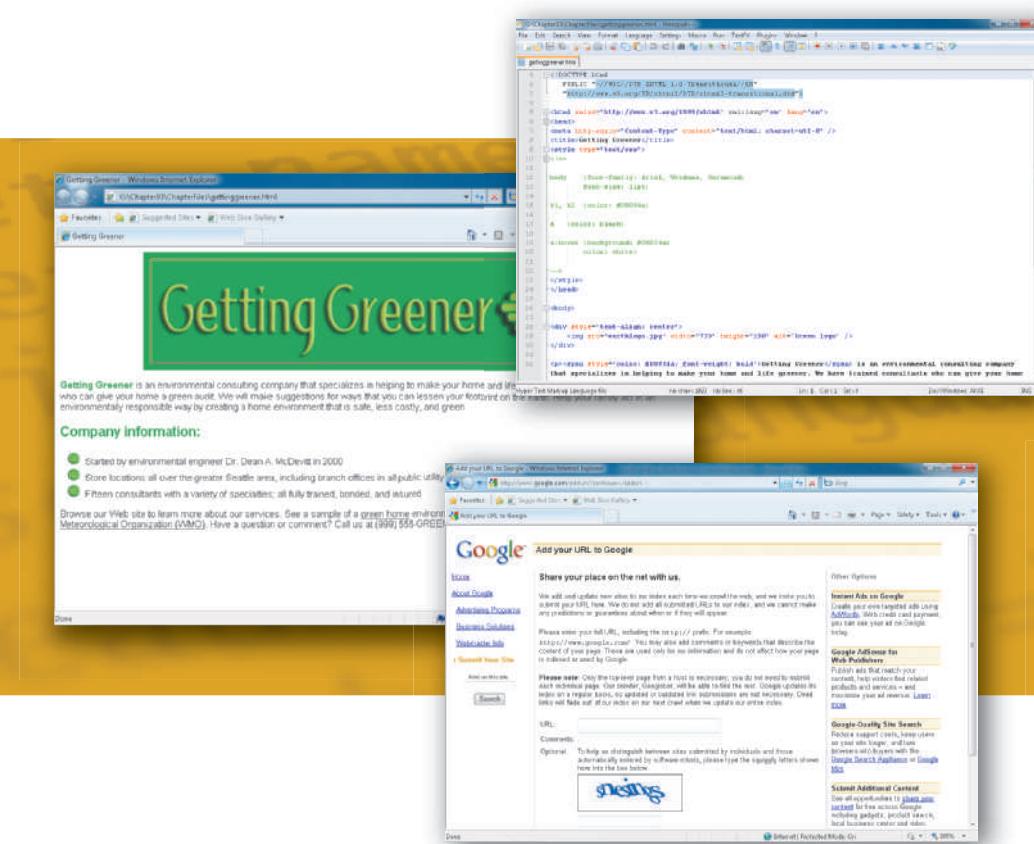
Working Together

Your design team at Triple-D Design has been asked to create a proposal for an existing customer to explain the value of using Cascading Style Sheets. Select a Web site with which you are familiar. Verify that the Web site does not utilize any of the three types of style sheets. Develop a graphic of the Web site hierarchy. Determine how the three types of style sheets could be used in this Web site and develop an outline explaining how they could enhance pages or sections of the site, add style consistency, or make the site easier to maintain. Write a proposal to the owners of the Web site that describes the features you could add with style sheets and the benefits of doing so, relative to the formatting techniques currently used in the Web site. As an example, you might want to address the number of times that a particular `` tag is used in the site and contrast that with the ease of using one external style sheet and a link statement per page. Use other ideas as discussed in the chapter project to emphasize the other benefits of style sheets. Write the proposal in the form of a bid, giving time estimates and costs associated with the development effort. Include your hierarchy chart and style sheet outline as appendices to the proposal.

This page intentionally left blank

Special Feature 1

Attracting Visitors to Your Web Site



Objectives

You will have mastered the material in this special feature when you can:

- Add keywords and descriptions to your Web pages
- Find appropriate Web site servers
- Determine the availability of a domain name
- Discuss Web page publishing options
- Develop an advertising plan to get the word out about your Web site

Introduction

In Chapter 3, you developed the Getting Greener Web site, which consisted of two Web pages. In this special feature, you learn how to fine-tune that Web site to make sure that it will attract visitors. A Web site is a passive marketing tool; it serves no purpose if no one knows that it is there. It is not enough to just develop a Web site. You also have to make modifications to the Web pages to ensure that they will attract visitors.

Project — Attracting Visitors

Web sites have become an important means of worldwide communication. Businesses utilize Web sites to communicate with their customers and vendors. Teachers create Web sites to communicate with other teachers and their students, and private users create Web sites to share aspects of their personal life with family, friends, and others.

In Chapter 3, you created the Getting Greener home page, `gettinggreener.html`, as shown in Figure 1. The project in this feature shows you how to utilize `<meta />` tags to add keywords and descriptions to this page to help Web site visitors looking for such topics find your Web page.

Overview

As you read through this feature, you will learn how to add keywords and descriptions to the meta tags that you previously developed (Figure 1a) to the Web page, as shown in Figure 1b. You also learn how to find a hosting site, determine a domain name, publish the Web pages, and determine an advertising plan by performing these general tasks:

- Decide what meta names (keywords and descriptions) you should use.
- Insert the keywords and descriptions into the meta tags.
- Identify available domain names.
- Determine an appropriate hosting situation for your Web site.
- Establish an advertising plan.

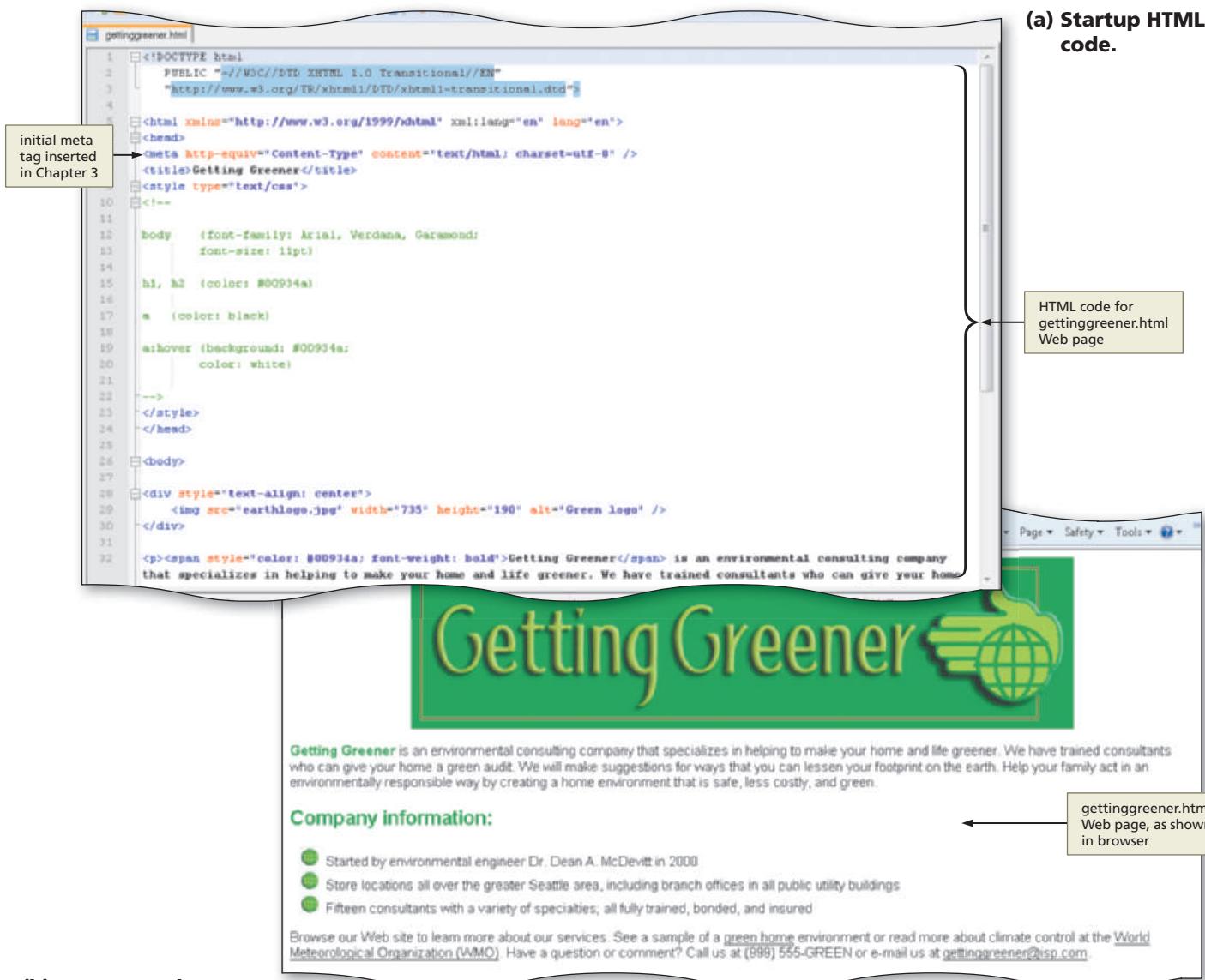


Figure 1

General Project Guidelines

In the Getting Greener home page in Chapter 3, you added a title, Getting Greener, that included the keyword “greener” in it. That word “greener” identifies the main subject matter for the Web site, but while your topic is clear, more must be done to your Web site. Once your Web page is complete, you need to publish it and attract visitors. There are several ways to attract visitors, which you will accomplish in the following project. In preparation for this project, you should follow these general guidelines:

Plan Ahead

- 1. Identify the meta names you should use.** There are different meta names that you can use within the `<meta />` tag, including keywords and description. In this step, you determine whether to use keywords or description or to use both keywords and description in different meta tags.
- 2. Determine the keywords and descriptions that reflect the purpose of the Web site.** Before you add keywords and description to your meta tags, you need to determine the words and description that apply to the Web site. When Web users are searching for informative sites, they type those keywords or description into the search engine. You should put yourself in the role of a user looking for information relative to the theme of your Web site. This helps you come up with great ideas for the keywords that will work. These keywords/description are in addition to the relevant phrases already included in the Web page title or body content.

(continued)

Plan Ahead

(continued)

3. **Decide the available domain names.** A Web page address or URL is an important part of advertising your Web page. You can register your domain name, which will make it easier for Web users to find your site. You need to decide on a few possibilities and then determine if those domain names are already registered. If the name you choose is not registered, you can purchase and register it.
4. **Assess your Web hosting alternatives.** Many Web developers have access to their own corporate Web servers. If you do, then you do not have to consider other Web hosting options. If this is not an option, then you need to find a Web server on which to host your Web pages.
5. **Establish an advertising plan.** You have many choices for publicizing your Web pages. Most companies include the URL in all corporate correspondence, including letterheads, advertisements, and products. Sharing links with a related Web site also helps get visitors. You need to determine a plan that will incorporate the best techniques to effectively publicize your Web pages.

Adding Keywords

You have already created the HTML file that is used in this special feature. You use the file gettinggreener.html that you created in Chapter 3. The page includes a number of keywords: You added the word “greener” to the title on the Web page; you also used the phrases “environmental consulting company”, “green audit”, and “green home” in the Web page content. Any of these phrases might be used by visitors searching for companies that deal with environmental or “green” issues. You can explicitly identify the keywords that you want the search engine to find by adding additional keywords and phrases to your <meta /> tag.

As with other projects, you use Notepad++ to enhance this file by adding keywords and descriptions. To include additional information in your Web page, you will follow these general steps:

1. Open the gettinggreener.html file in Notepad++.
2. Add the keywords and description to new <meta /> tags.
3. Save and validate the file.

To Open the File

For this project, you will add keywords and a description to the gettinggreener.html Web page already created. The following steps show you how to add keywords.

- 1 Start Notepad++.
- 2 Open the gettinggreener.html file in the Chapter03\ChapterFiles folder that you stored on the G:\ drive (Figure 2).
- 3 If necessary, enable Word wrap in Notepad++.

Q&A What if I did not create the gettinggreener.html file from Chapter 3?
Your instructor should have a copy of the gettinggreener.html file.

Other Ways

1. Right-click the file name gettinggreener.html in Windows Explorer, select Edit with Notepad++

```

<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Getting Greener</title>
<style type="text/css">
<!--
body {font-family: Arial, Verdana, Garamond; font-size: 1pt}
h1, h2 {color: #00934a}
a {color: black}
a:hover {background: #00934a; color: white}
-->
</style>
</head>
<body>
<div style="text-align: center">
<img src='earthlogo.jpg' width="735" height="190" alt="Green logo" />
</div>
<p><span style="color: #00934a; font-weight: bold">Getting Greener</span> is an environmental consulting company that specializes in helping to make your home and life greener. We have trained consultants who can give your home a green makeover!</p>
</body>

```

Figure 2

Meta Names

There are several meta names that you can use in the `<meta />` tag, as shown in Table 1. You already included one meta tag in the initial HTML code that you inserted into every Web page created thus far. The

```
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
```

line has been included in all Web pages throughout the book. As mentioned earlier, this statement declares the character-encoding as UTF-8. The Unicode Transformation Format (UTF) is a compressed format that allows computers to display and manipulate text. When the browser encounters this meta tag, it will display the Web page properly, based on the particular UTF-8 encoding embedded in the tag. UTF-8 is the preferred encoding standard for Web pages, e-mail, and other applications.

Some of the more frequently used meta names are listed in Table 1. In this project, you use two of those meta names: `description` and `keywords`. The `keywords` are used by some search engines to find your Web pages. Other search engines use the `keywords` included in the content of your Web pages. The best plan includes putting `keywords` in both places. The `description`, on the other hand, is what some search engines add next to your Web page URL to describe the content of the Web page. Visitors often look at that description to determine whether they want to click that particular link (or URL) from the list of URLs that the search engine displays. The format that is used for each type of meta name is:

```
<meta name="keywords" content="green ideas, eco friendly" />
<meta name="description" content="Seattle area home or business environmental consulting.">
```

where `name` identifies the type of meta element that you use, and `content` identifies the specific phrases or words that are used.



SEO

SEO is an acronym for Search Engine Optimization or Search Engine Optimizer. SEO is the process of improving the amount of traffic that you get on your Web site. Optimizing a Web site involves editing the content and HTML in the Web page to increase its relevance to specific keywords and to remove barriers to the page indexing functions of search engines. To find more information, search for SEO on the Internet.

Table 1 Meta Names and Their Functions

Meta Name	Function
author	Supplies the name of the document author
description	Provides a description of the document
keywords	Provides a list of keywords that describe the document

To Add Keywords

The following step illustrates how to add keywords to the gettinggreener.html Web page.

1

- Click after the > at the end of line 7 and press the ENTER key to position the insertion point on line 8.
- Type `<meta name="keywords" content="green ideas, eco friendly"/>` and then press the ENTER key (Figure 3).

A screenshot of a code editor window titled "gettinggreener.html". The code is an XHTML document with the following structure:

```

<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="keywords" content="green ideas, eco friendly" />
    <title>Getting Greener</title>
    <style type="text/css">
        ...
    </style>
    <!--
        body {font-family: Arial, font-size: 1pt}
        h1, h2 {color: #00934a}
        a {color: black}
    -->

```

Annotations in the screenshot highlight specific parts of the code:

- A red box labeled "second meta tag added on line 8" points to the line containing `<meta name="keywords" content="green ideas, eco friendly" />`.
- A yellow box labeled "insertion point on line 9" shows the cursor position just before the closing quote of the previous tag.
- A red box labeled "keyword phrases 'green ideas' and 'eco friendly' added in meta tag" highlights the content attribute of the meta tag.

Figure 3

To Add a Description

The following step illustrates how to add a meta tag description to your Web page.

1

- If necessary, position the insertion point on line 9.
- Type `<meta name="description" content="Seattle area home or business environmental consulting."/>` but do not press the ENTER key (Figure 4).

Q&A

Is there a difference between keywords and description?

Yes. Search engines use keywords to find your Web pages, while they display descriptions next to the respective Web page URLs.

A screenshot of a code editor window titled "gettinggreener.html". The code is an XHTML document with the following structure:

```

<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="keywords" content="green ideas, eco friendly" />
    <meta name="description" content="Seattle area home or business environmental consulting." />
    <title>Getting Greener</title>
    <style type="text/css">
        ...
    </style>
    <!--
        body {font-family: Arial, Verdana, Garamond; font-size: 1pt}
        h1, h2 {color: #00934a}
        a {color: black}
    -->

```

Annotations in the screenshot highlight specific parts of the code:

- A red box labeled "third meta tag added on line 9" points to the line containing `<meta name="description" content="Seattle area home or business environmental consulting." />`.
- A red box labeled "description added" points to the content attribute of the meta tag.

Figure 4**Q&A**

If I have relevant keywords in the content of my Web page, why should I add other words to the `<meta />` tags?

It always helps to have keywords identified in both places for those search engines that choose one method over the other.

To Save, Validate, and Print a Document

You are finished entering the meta tags. As with all Web development projects, you now need to save, validate, and print the file.

- 1** In Notepad++, click the Save icon on the toolbar to save the file with the new meta tags.
- 2** Validate the file using the w3.org validation service.
- 3** Once the file is successfully validated, print the file (Figure 5).

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<meta name="keywords" content="green ideas, eco friendly" />
<meta name="description" content="Seattle area home or business environmental
consulting." />
<title>Getting Greener</title>
<style type="text/css">
<!--
body {font-family: Arial, Verdana, Garamond;
font-size: 11pt}

hi, h2 {color: #00934a}

a {color: black}

a:hover {background: #00934a;
color: white}

-->
</style>
</head>

<body>

<div style="text-align: center">

</div>

<p><span style="color: #00934a; font-weight: bold">Getting Greener</span> is an
environmental consulting company that specializes in helping to make your home and life
greener. We have trained consultants who can give your home a green audit. We will make
suggestions for ways that you can lessen your footprint on the earth. Help your family
act in an environmentally responsible way by creating a home environment that is safe,
less costly, and green.</p>

<h2>Company information:</h2>

<ul style="list-style-image: url(earthbullet.jpg)">
<li>Started by environmental engineer Dr. Dean A. McDevitt in 2000</li>
<li>Store locations all over the greater Seattle area, including branch offices in all
public utility buildings</li>
<li>Fifteen consultants with a variety of specialties; all fully trained, bonded, and
insured</li>
</ul>

<p>Browse our Web site to learn more about our services. See a sample of a <a href=
"greenhome.html">green home</a> environment or read more about climate control at the <a
```

Figure 5

Determining a Domain Name

You may use the domain name of the server on which you publish your Web pages together with a path to your specific pages as an option for your Web page address or URL. A **domain name** is the server name portion of the URL. You are also able to register your own domain name on the Internet for about \$10 per year. That sometimes makes it easier for visitors to find your Web pages. In the case of the gettinggreener.html file, you could register a domain name that is available for your Getting Greener Web site.

BTW

InterNIC

InterNIC contains trusted public information regarding Internet domain name registration services. The InterNIC Web site has a FAQs section, information about domain name registrars, as well as links for domain name disputes.

To determine if the domain name you are considering is available, you can start your search at InterNIC. InterNIC is a registered service mark of the U.S. Department of Commerce. The InterNIC Web site (www.internic.net/alpha.html) is operated by the Internet Corporation for Assigned Names and Numbers (ICANN) to provide information to the public regarding Internet domain name registration services. ICANN is responsible for managing and coordinating the Domain Name System (DNS) to ensure that every Internet address is unique, and that all users of the Internet can find all valid addresses.

Check Domain Name Availability

In order to check to see if a domain name is available, complete the following steps:

1

- Open Internet Explorer.
- Type `http://www.internic.net/alpha.html` into the address bar, as shown in Figure 6.
- Review the FAQ section of the Web site to better understand the domain naming process.
- Click the Whois link to see what domain names have previously been registered for gettinggreener. You can try .com, .net, or .org to see the results.

links to variety of functions within InterNIC Web site

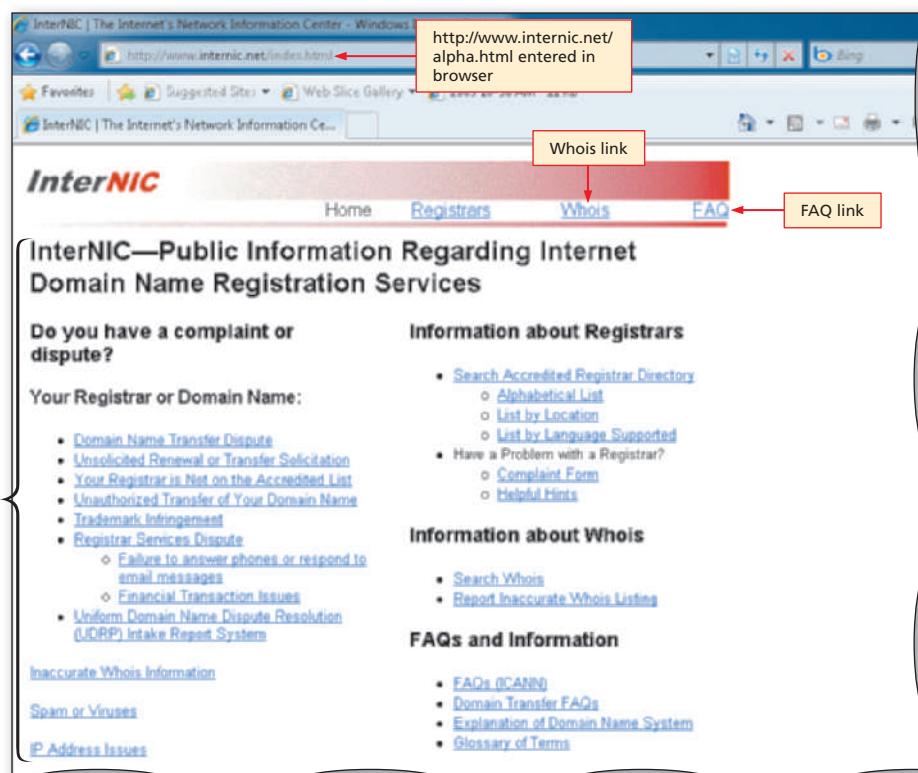


Figure 6

Finding a Web Hosting Site

The next step in the Web development process is to publish your Web pages so that visitors can see them. In order to publish your Web site, you need hosting services. There are many options available for Web hosting. You need access to a Web server onto which you can upload all of the Web pages in your Web site.

One option is to use the ISP that you use to connect to the Internet. ISPs sometimes provide space for their clients to host a Web site. If you registered your own domain name, you can even have your ISP set up a virtual domain, or shared Web hosting, on their server with your new domain name. Your ISP's server may be set up to allocate hosting services and bandwidth to more than one Web site by using a virtual domain. In other words, although you register the domain name gettinggreener.com, your ISP would host it on its own Web server. Virtual Web hosting is a much cheaper option because you do not have to pay for a dedicated server to host just your Web site. You can check with your ISP for details. A second option is to use a company that charges for Web site hosting. There are thousands of companies that provide Web hosting services. Most charge a monthly fee, but some offer free Web hosting in exchange for advertising on your Web site. A final option is for you to set up your own Web server. You would have to know enough about the technology to set it up and keep it running.

If you choose to utilize your own ISP or a Web hosting service, you need to consider a few things. These include:

- What is the total cost? Compare monthly or annual costs; the highest cost may not always provide the best service.
- How much space is available to you? You need to assess your current needs (i.e., file sizes, sizes of graphics) and also your future needs (i.e., how much more information you will create).
- How fast is the connection speed? The speed of the connection to the Internet is important to efficiently serve your visitors.
- How much total bandwidth transfer is available? The number and size of Web pages in your Web site together with the number and size of graphical images is important to consider.
- Do they provide technical support? You may occasionally need help, especially in the beginning.
- Do they offer tracking services? Many hosting companies allow you to see how visitors utilize your Web site by viewing a tracking log.

After you have selected a Web hosting service, you need to transfer your files to that server.

Publishing Your Web Site

Once you have determined a Web hosting strategy, the next step is to publish your Web pages so that visitors can see them. **Publishing** your Web site means transferring your files to the Web server. There are many options available for file transfers. You could use a File Transfer Protocol (FTP) program such as WS-FTP for Windows (Figure 7) or Fetch for Mac (for more information about FTP programs, see Appendix E). In addition, many Web page editors also provide publishing functionality. Once your Web pages are published, the last step is to advertise their location to attract visitors.

BTW

Web Site Hosting

There are a variety of Web site hosting options available today. To search for Web hosting services, use different search engines to find different alternatives. Make sure to use the checklist shown on this page to assess the hosting services and fees.

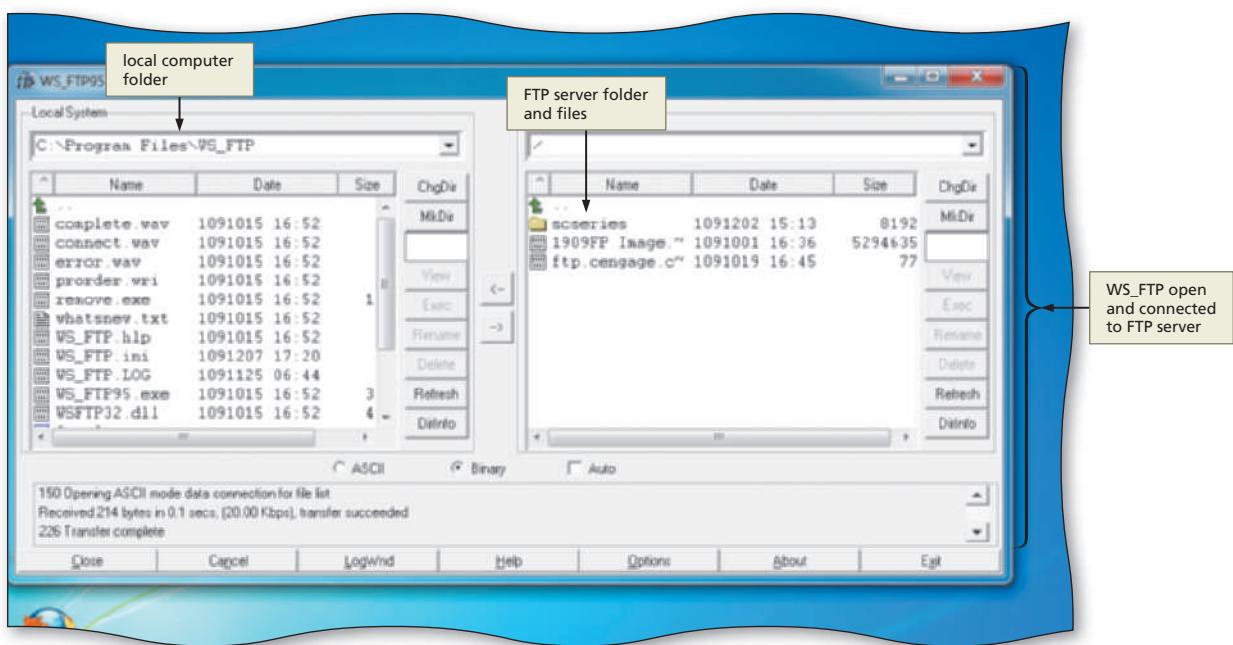


Figure 7

Advertising Your Web Site

Now that your Web pages are published, you need to get the word out to potential Web site visitors. You need to determine a comprehensive advertising plan. It serves no purpose for you to publish a Web site if no one visits it. There are several ways to start an advertising campaign:

- Put your URL on your business cards, company brochures, stationery, and e-mail signature.
- Advertise your URL in newsletters and print articles.
- Tell people verbally about your Web site.
- Find and get listed on targeted directories and search engines specific to your industry.
- Buy banner ads.
- Negotiate reciprocal links in which you agree to link to a Web site if they agree to link to your Web site.
- Utilize newsgroups specific to your industry.

BTW

Search Engines

Both Google and Yahoo! contain information about optimizing your Web sites. Review their Webmaster guidelines for great ideas on registering with their search engines. Google even provides a Google 101 section that explains how Google indexes and serves the Web.

Registering with Search Engines

You are finished entering the meta tags, publishing, and advertising your Web site. The next step is to register your Web site with the two most popular search engines, Google and Yahoo! It is also a good idea to register your site with search engines that specialize in subject matter related to your Web site.

To Register Your Web Site with Search Engines

The next step shows you how to register your Web pages with the Google and Yahoo! search engines.

1

- In Internet Explorer, type `http://www.google.com/addurl.html` in the address bar and press the ENTER key (Figure 8).
- Follow the directions to add your URL.

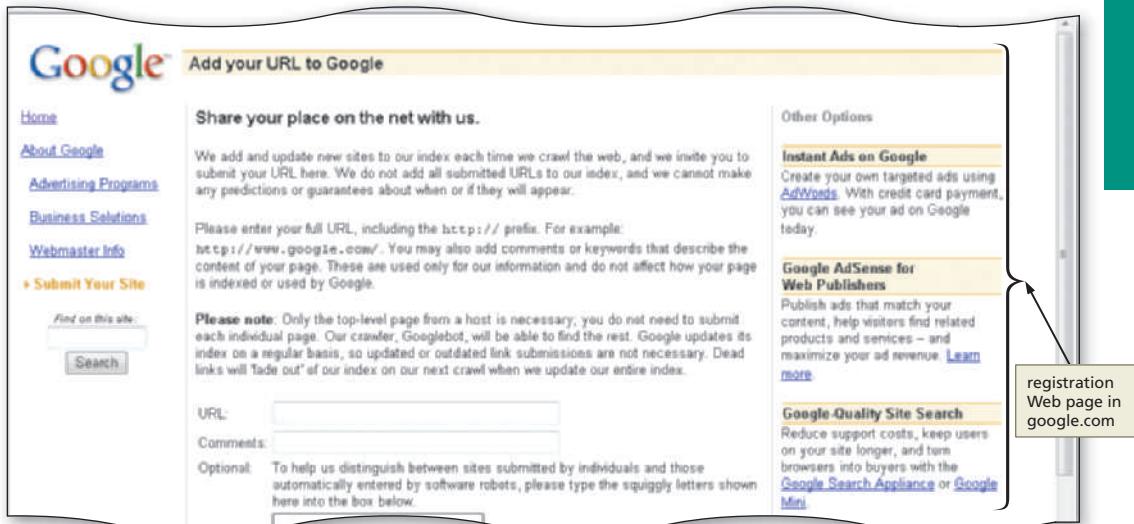


Figure 8

2

- For the Yahoo! registration, type `http://search.yahoo.com/info/submit.html` in the Internet Explorer address bar and press the ENTER key (Figure 9).
- Follow the directions to add your URL.

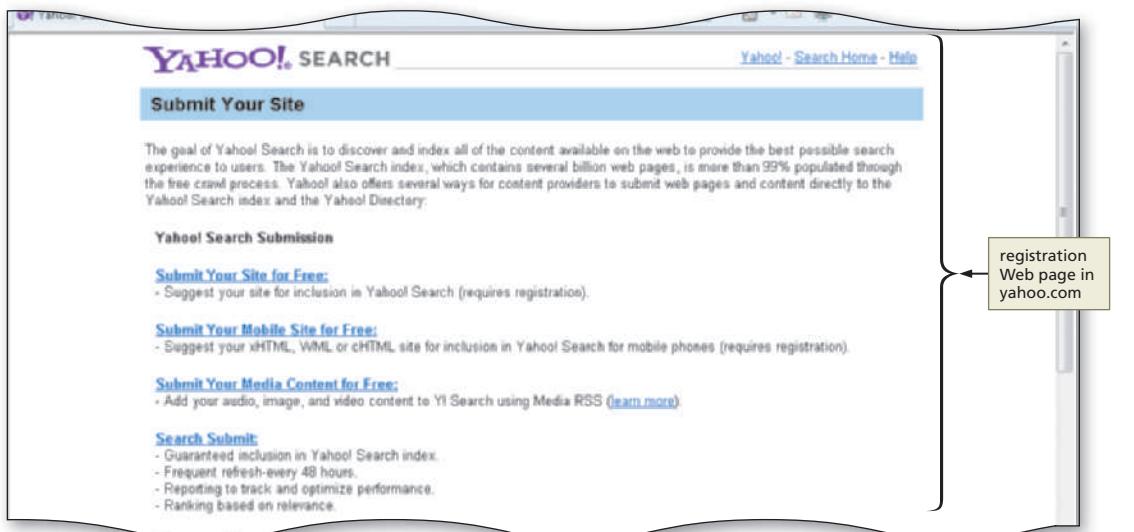


Figure 9

Feature Summary

In this feature, you have learned how to insert keywords and description meta tags into your Web page. You also learned about domain names, how to search for a Web host, publish your Web page, advertise your Web site, and register your site with search engines. The items listed below include all the new skills you have learned in this feature.

- Add Keywords (HTML 226)
- Add a Description (HTML 226)
- Check Domain Name Availability (HTML 228)
- Register Your Web Site with Search Engines (HTML 230)

In the Lab

Design and/or create a document using the guidelines, concepts, and skills presented in this chapter. Labs are listed in order of increasing difficulty.

Lab 1: Creating a Publishing and Advertising Plan

Problem: Your assignment is to apply the ideas and suggestions listed in this special feature to write a comprehensive publishing and advertising plan. This plan should all be noted in a Word file that can be submitted to your instructor.

Instructions:

1. Determine a domain name that is available to use for your Getting Greener Web site.
2. Research and identify several possible Web hosting options. Answer all six questions covered in this special feature section for each of your hosting options.
3. Research and identify the specifics about registering your Web site with both Google and Yahoo!.
4. Write an advertising plan that addresses specific ways that you can get the word out about your Web site. Specific plans should include:
 - a. Locating targeted directories and search engines specific to the industry reflected in the Web site
 - b. Investigating the pros and cons of buying banner ads
 - c. Determining Web sites to which you could have possible reciprocal links
 - d. Finding newsgroups specific to the industry reflected in the Web site
5. Save the file with the name Lab SF1-1 Advertising. Submit the file in the format specified by your instructor.

In the Lab

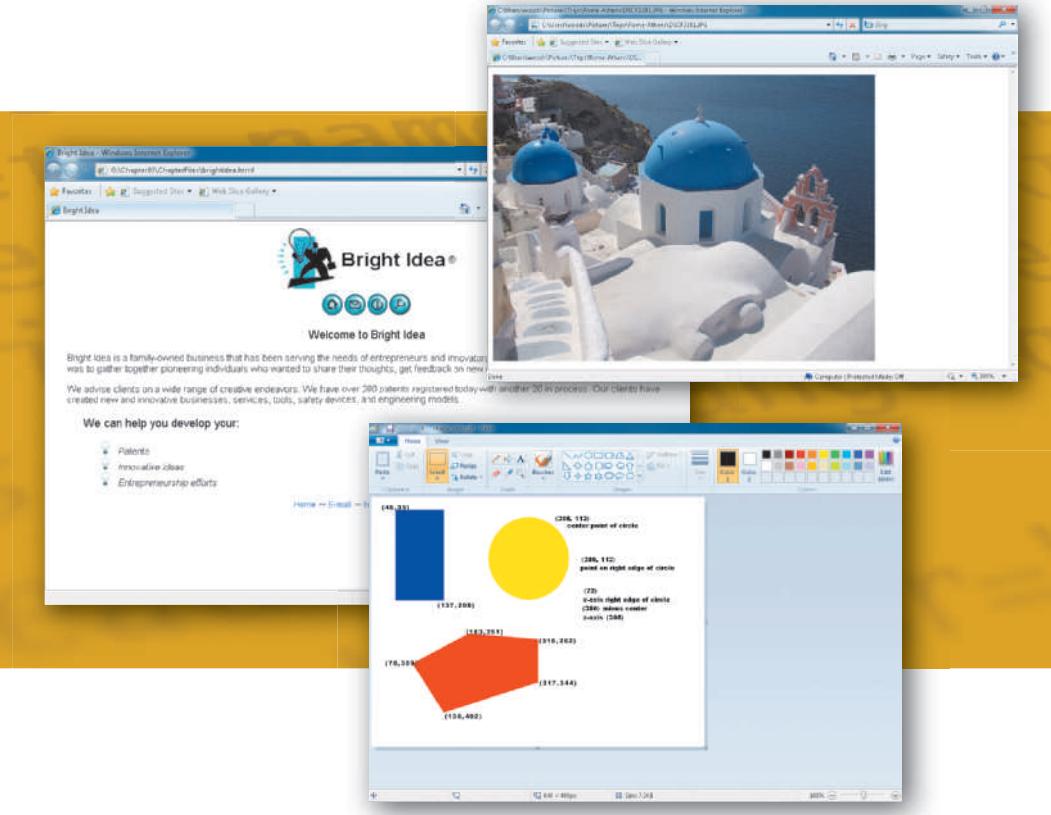
Lab 2: Attracting Visitors to Another Web Site

Problem: In this assignment, you are to assess another Web site to complete the same basic steps as taken in this feature to improve the site's visibility on the Internet.

Instructions:

1. Select a Web site that is of interest to you and open the site's home page.
2. Review the meta tags (if any) that are used on the home page. (*Hint:* View the page source to review those tags.) Determine how you can utilize additional meta tags for this Web page. What keywords and descriptions would be good to use? What keywords and descriptions do other related Web sites use?
3. Write an advertising plan that addresses specific ways that you can get the word out about this Web site. Specific plans should include:
 - a. Locating targeted directories and search engines specific to the industry reflected in the Web site
 - b. Determining Web sites to which you could have possible reciprocal links
 - c. Finding newsgroups specific to the industry reflected in the Web site
4. Save the file with the name Lab SF1-2 Advertising. Submit the file in the format specified by your instructor.

5 Creating an Image Map



Objectives

You will have mastered the material in this chapter when you can:

- Define terms relating to image mapping
- List the differences between server-side and client-side image maps
- Name the two components of an image map and describe the steps to implement an image map
- Distinguish between appropriate and inappropriate images for mapping
- Sketch hotspots on an image
- Describe how the x- and y-coordinates relate to vertical and horizontal alignment
- Open an image in Paint and use Paint to locate the image map coordinates
- Create a home page
- Create a table, insert an image into a table, and use the `usemap` attribute to define an image map
- Insert special characters into a Web page
- Use the `<map>` `</map>` tags to start and end a map
- Use the `<area>` tag to indicate the shape, coordinates, and URL for a mapped area
- Create an external style sheet for styles used across the Web site

5 | Creating an Image Map

Introduction

Many of the Web pages in Chapters 2 through 4 used the `` tag to add images. In Chapter 3, an image also was used as a link back to the home page, by using the `<a> ` tags to define the image as the clickable element for the link. When an image is used as a link, as in Chapter 3, the entire image becomes the clickable element, or hotspot. With an image map, the entire image does not have to be clickable. Instead, one or more specific areas serve as hotspots. An image map is a special type of inline image in which you define one or more areas as hotspots. For example, each hotspot in an image map can link to another part of the same Web page or to a different Web page. Using an image map in this way gives Web page developers significant flexibility, as well as creative ways to include navigation options. Instead of using only text links, a Web page can include an image map that highlights key sections of a Web site and allows a user to navigate to that section by clicking the appropriate area of the image map.

Project — Bright Idea®

Chapter 5 illustrates how to create an image map with links to other Web pages within the Bright Idea Web site. The Bright Idea Web site includes the home page and three additional Web pages, each linked to the home page using an image map and text links, and an e-mail link, as shown in Figure 5–1. In Chapter 5, you will create the home page of the Bright Idea Web site (Figure 5–1a) and an external style sheet that is used with all Web pages in the site. On this home page, you include a link to the `brightidea@isp.com` e-mail address that opens an e-mail program, as shown in Figure 5–1b. The Web pages shown in Figures 5–1c and 5–1d are included in the Data Files for Students. (See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files.) HTML tags are used to create the image map that supports the four clickable areas in the image. One of the key features of the Web is its support for graphics, so Web visitors expect to view many images on the Web pages that they visit. Images make Web pages more exciting and interesting to view and, in the case of image maps, provide a creative navigational tool.

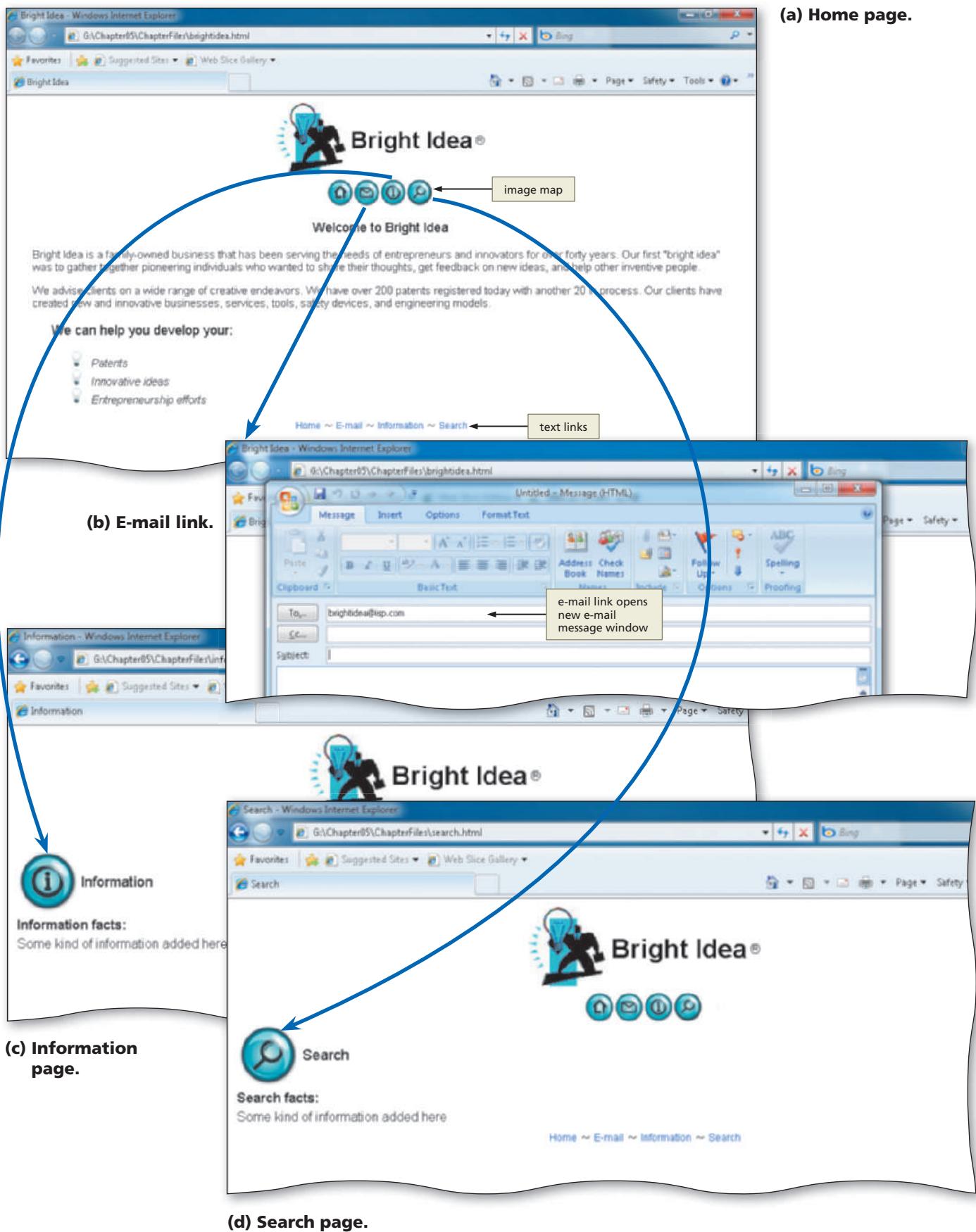


Figure 5-1

Overview

As you read this chapter, you will learn how to create the Web pages shown in Figure 5–1 on the previous page by performing these general tasks:

- Enter HTML code into the Notepad++ window.
- Save the file as an HTML file.
- View an image in Microsoft Paint to see image map coordinates.
- Enter basic HTML tags and add text to the file.
- Insert an image to be used as an image map.
- Create an image map by mapping hotspots on the image.
- Create links to the other Web pages and to the home page with a horizontal menu bar.
- Add special characters to the home page.
- Create an external style sheet and insert the link into the home page.
- Save and print the HTML and CSS code.
- Validate, view, and print the Web pages.

Plan Ahead

General Project Guidelines

As you create Web pages, such as the project shown in Figure 5–1 on the previous page, you should follow these general guidelines:

1. **Plan the Web site.** As always with a multiple-page Web site, you should plan the site before you begin to write your HTML code. Refer to Table 1–4 on page HTML 15 for information on the planning phase of the Web Development Life Cycle.
2. **Analyze the need.** In the analysis phase of the Web Development Life Cycle, you should analyze what content to include on the Web page. The Web development project in Chapter 5 is different from the one completed in other chapters because it contains an image map. Part of the analysis phase then includes determining what image to use and where to put links within the image map.
3. **Choose the image.** You need to select an image that has distinguishable areas that can be used as links. Not all images are conducive to image mapping, as described in the chapter.
4. **Determine what areas of the image map to use as links.** Once an appropriate image is selected, you need to determine how to divide up the image map for links. You want to make sure that your hotspot (link) areas do not spill over into each other.
5. **Establish what other links are necessary.** In addition to links between the home page and secondary Web pages, you need an e-mail link on this Web site. It is a general standard for Web developers to provide an e-mail link on the home page of a Web site for visitor comments or questions. Additionally, you need to provide links to all other Web pages on the Web site (information.html and search.html).
6. **Create the Web page, image map, and links.** Once the analysis and design are complete, you create the Web pages. Good Web development standard practices should be followed, such as utilizing the initial HTML tags as shown in previous chapters, providing text links for all hotspots in the image map, and always identifying alt text for images.
7. **Test all Web pages within the Web site.** It is important to test your pages to assure that they follow XHTML standards. In this book, you use the World Wide Web Consortium (W3C) validator to validate your Web pages. Additionally, you should check all content for accuracy. Finally, all links (image map hotspots, text links, and page-to-page links within the same Web site) should be tested.

When necessary, details concerning the above guidelines are presented at appropriate points in the chapter. The chapter also will identify the actions performed and decisions made regarding these guidelines during the creation of the Web pages shown in Figure 5–1.

Introduction to Image Maps

In this chapter, you use an image map to create four clickable areas within a single menu bar image: a link to the home page, an e-mail link, a link to the Information page, and a link to the Search page. All four of the clickable areas have a circular shape. Figure 5–2 shows the four circular clickable areas, each of which encloses a specific area. A Web page visitor clicking one of the circular-shaped clickable areas will link to an e-mail window or to one of the associated Web pages (Home, Information, or Search).

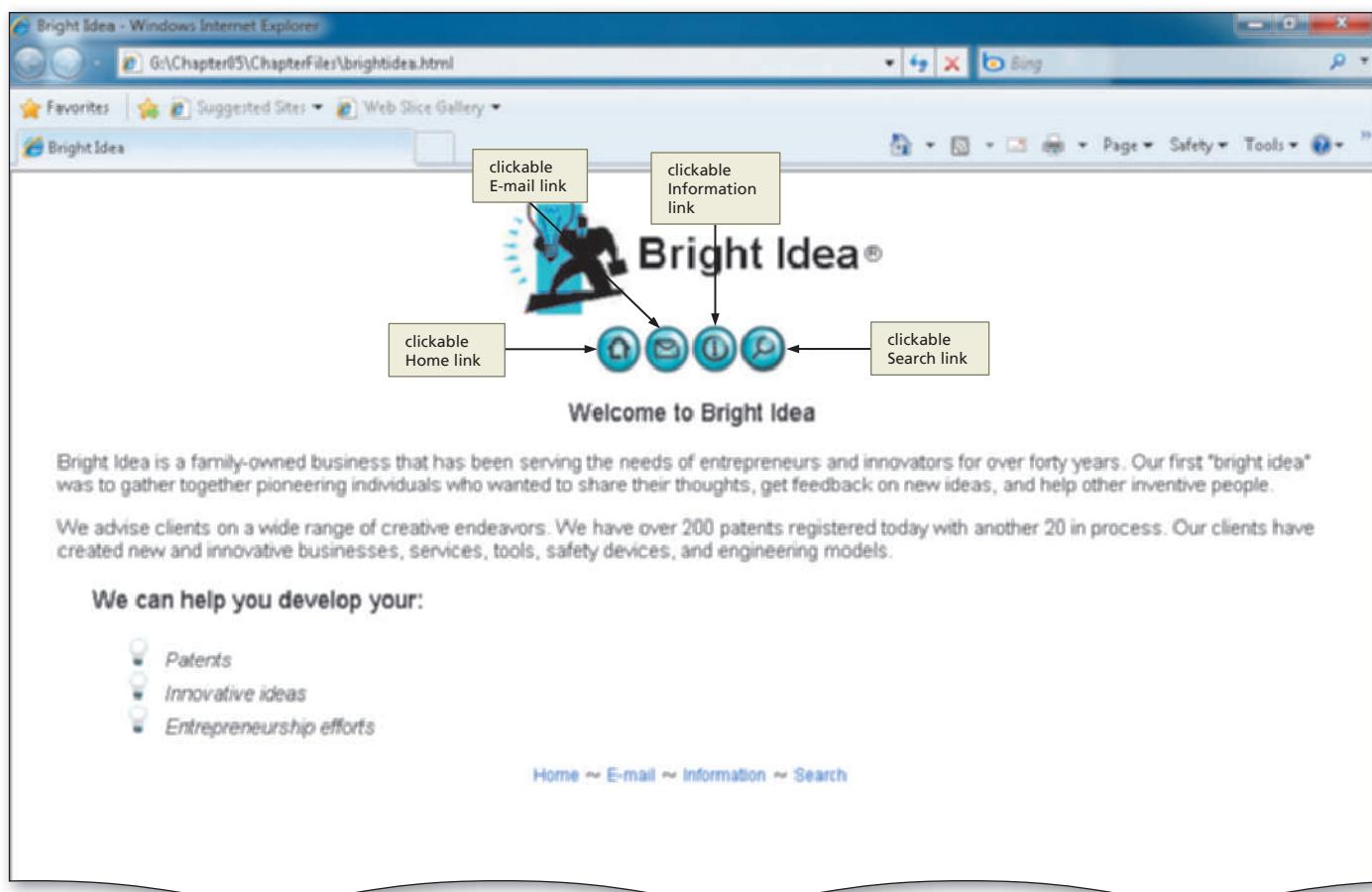


Figure 5–2

Using Image Maps

One of the risks in using image maps to provide navigational elements is that if the image does not load, a user will not have the ability to navigate to other linked Web pages. Another potential issue is that using a large image for an image map can increase the amount of time required for pages to download over lower-speed connections. To avoid such performance issues, some people turn off the viewing of images when they browse Web pages, electing to display only text in their browsers. These users, and users of text-based browsers, also will not be able to navigate a Web page that relies on an image map. For these reasons, a Web page that uses an image map for navigation also should include text links to the URLs and reflect these in the image map, as shown in Figure 5–3a on the next page. Using text links in conjunction with the image map ensures that if the image does not download or a Web page visitor has images turned off, as shown in Figure 5–3b, a user still can navigate to other Web pages using the text links.

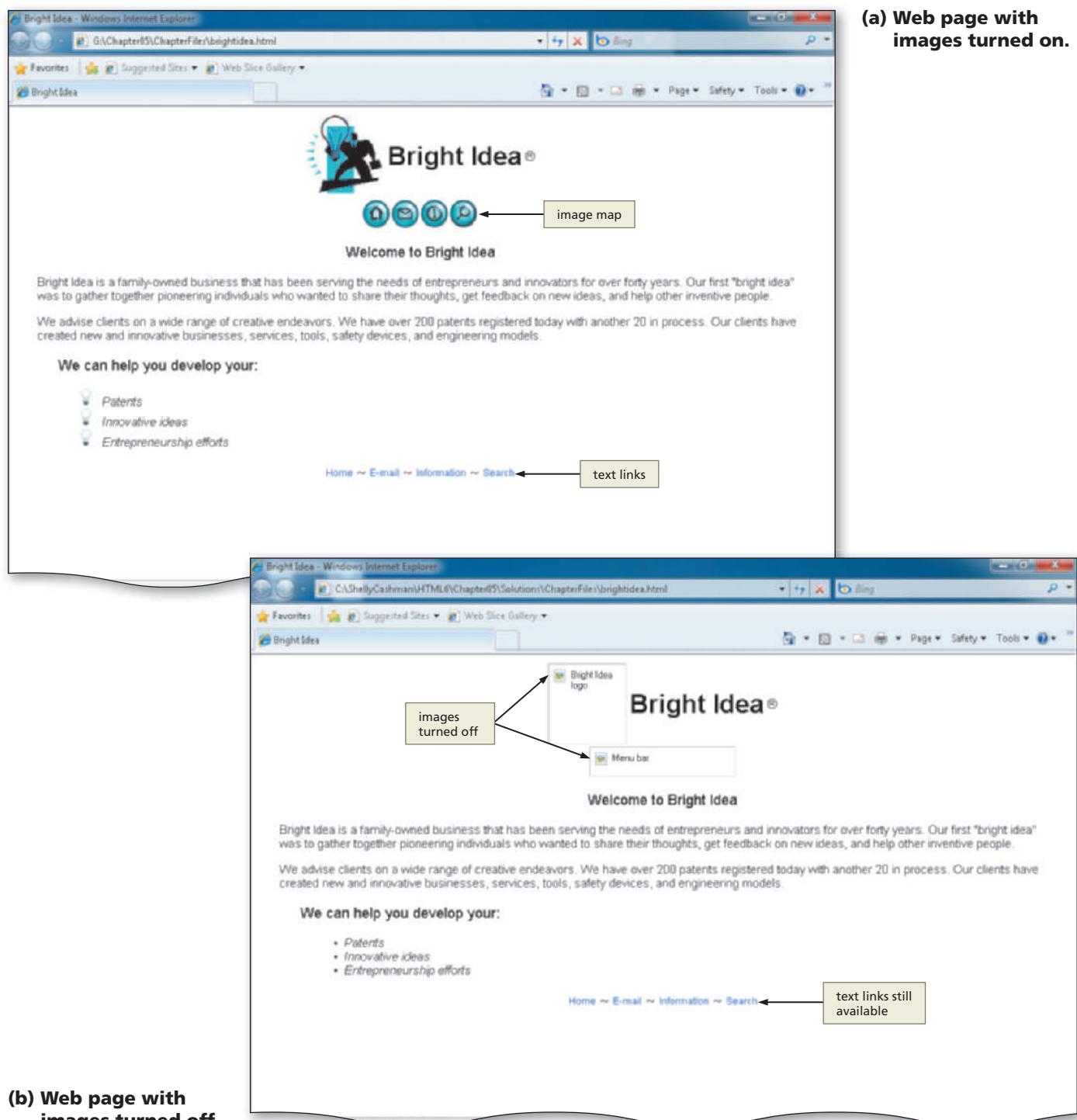


Figure 5-3

Image maps can enhance the functionality and appeal of Web pages in many ways. For example, an image map can be used as an **image map button bar**, which is a menu bar that uses graphical images, as shown in Figure 5-4 and in this chapter's project. This makes the menu bar a more attractive feature of the Web page.

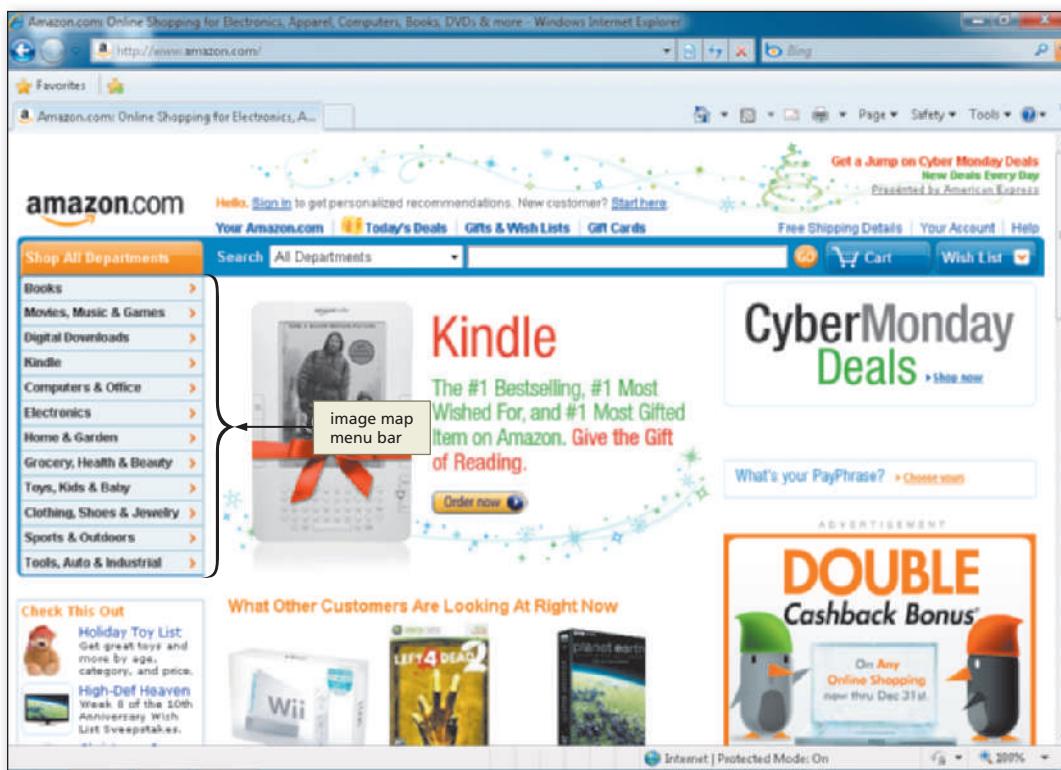


Figure 5–4 Image map menu bar.

Image maps are also utilized to divide a geographical map into hotspots, as shown in Figure 5–5. A Web page visitor can click a geographical area on the map and be linked to additional information about that location.

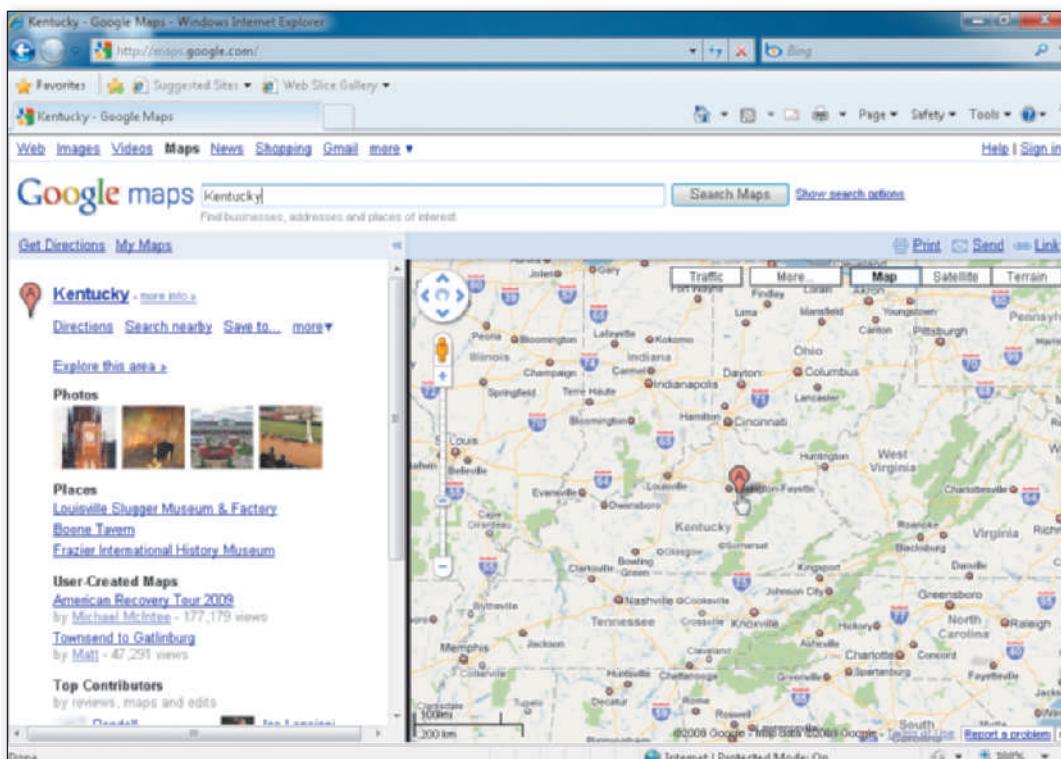


Figure 5–5 Image map on map Web page.

A theater may use an image map to show the seating chart of the facility (Figure 5–6). You could develop an application that links from the ticket purchase Web page to the seating chart image map and allow customers to select their seats. This map shows how the various tables are arranged, which helps customers understand the general layout of the theater.

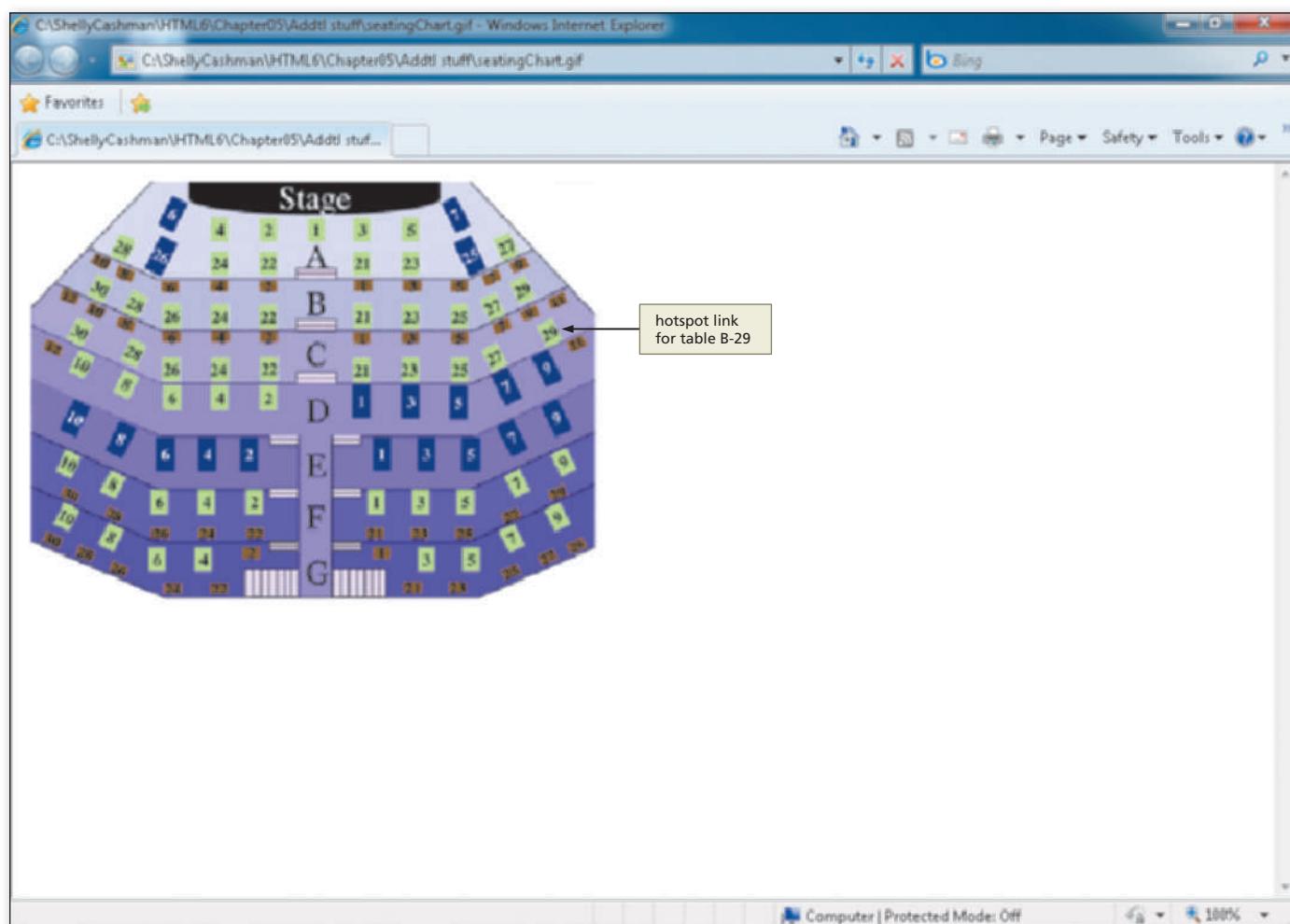


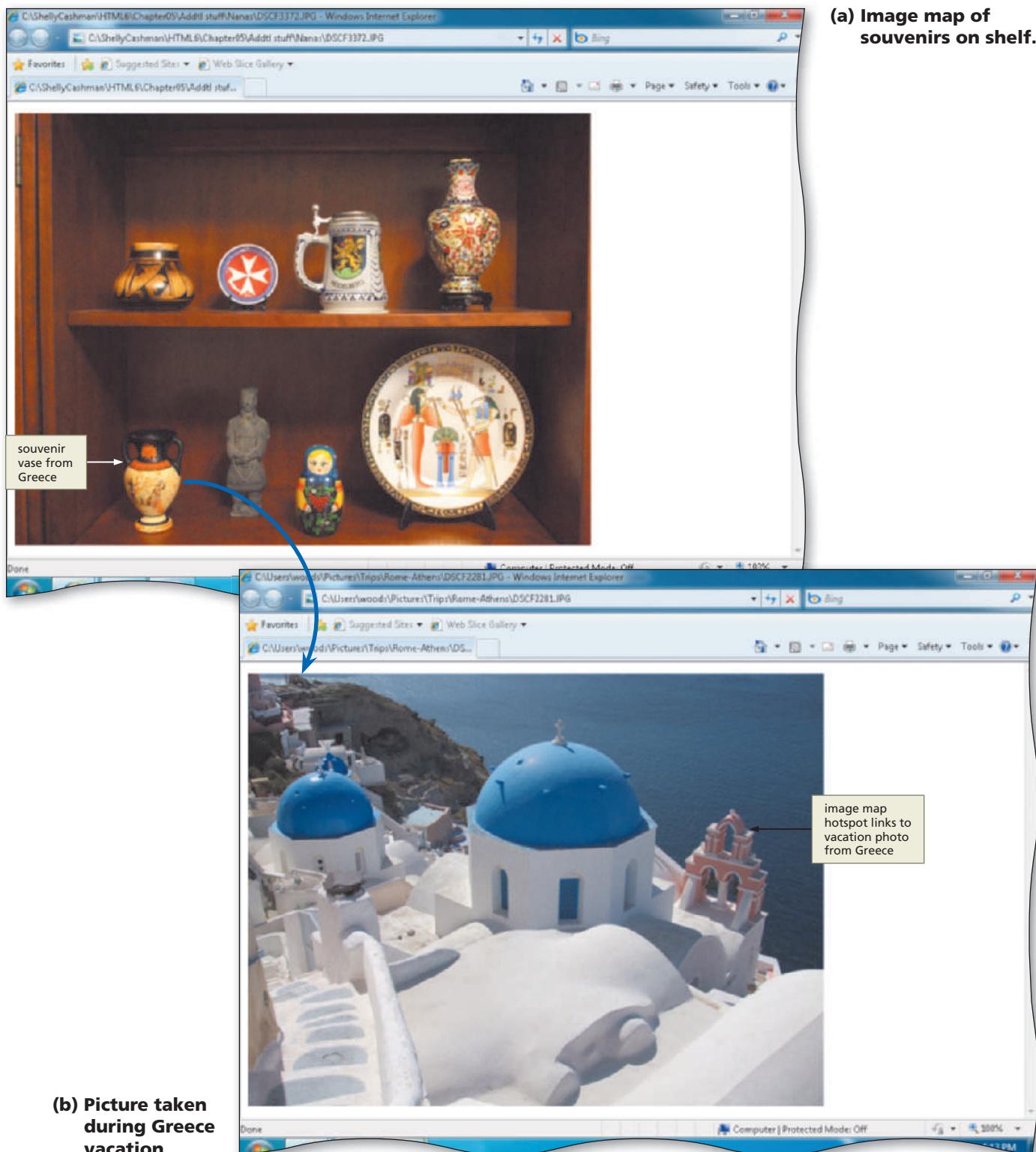
Figure 5–6 Image map of theater seating chart.

BTW

Image Maps

Image maps are used frequently for Web site navigation. Many online HTML sources address the purposes of image maps and give suggestions for their use. Review the w3schools section of the World Wide Web Consortium site (www.w3schools.com) for comprehensive image map information.

Image maps can be used for many applications. For instance, if you want to create a Web site that depicts your vacation travels, you could take a digital picture of your travel souvenirs (Figure 5–7a) and use that as an image map, linking personal souvenirs to photographs (Figure 5–7b) that you took in a particular country that you visited.

**Figure 5–7**

There are real estate applications in which an image map of the house floor plan is used (Figure 5–8). This map allows a potential customer to click a hotspot of a particular room in the house in order to see pictures or get additional information about that room. Using an image map like this gives people the opportunity to view for sale houses online.

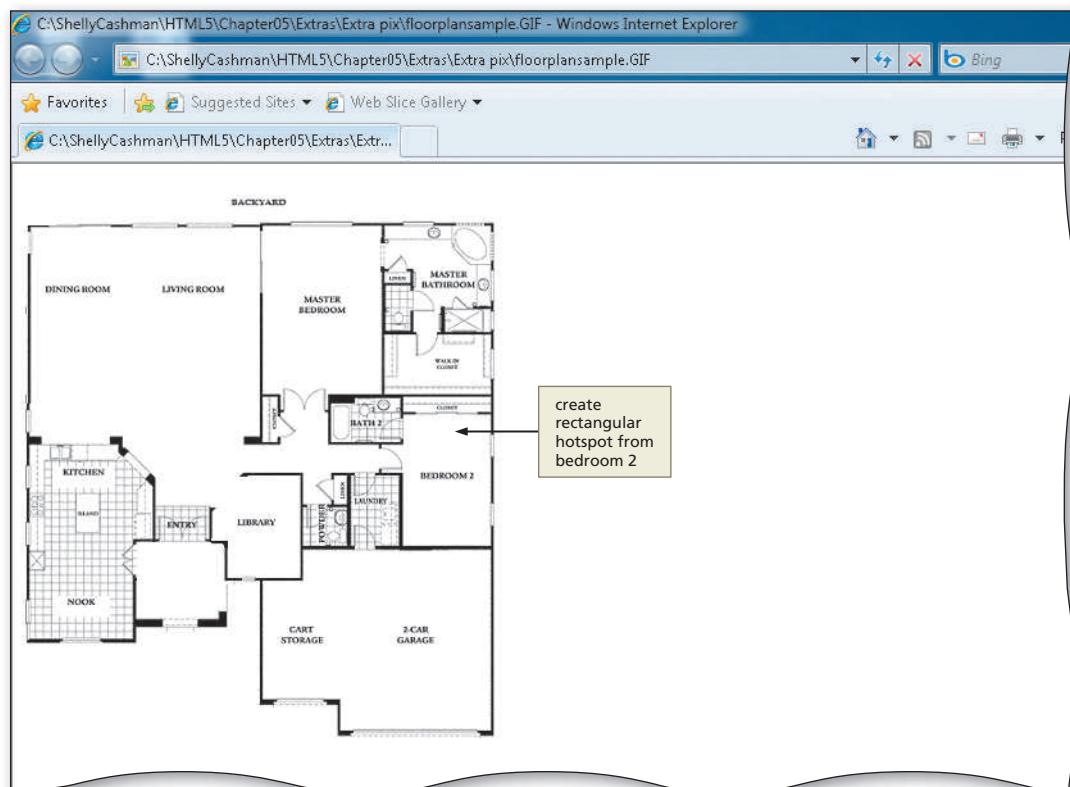


Figure 5–8 Floor plan can be used as image map.

BTW

Server-Side vs. Client-Side Image Maps

Web sites exist that provide information about server-side versus client-side image maps. To see an example of how image maps can be used for Web pages and which type is more efficient, use a search engine to search for the terms “server-side image maps” and “client-side image maps”.

BTW

Server-Side Image Maps

When a hotspot on an image map is clicked, a special image map program that is stored on the Web server is run. In addition, the browser also sends the x- and y-coordinates to the Web server for the position of the link on the image map. Most, if not all, browsers support server-side image maps.

Server-Side vs. Client-Side Image Maps

Two types of image maps exist: server-side and client-side. In a **server-side image map**, the image is displayed by the client (browser) and implemented by a program that runs on the Web server. When a Web page visitor clicks a link on a server-side image map, the browser sends the x- and y-coordinates of the mouse click to the Web server, which interprets them and then links the visitor to the correct Web page based on those coordinates. Thus, with a server-side image map, the Web server does all the work.

With a **client-side image map**, the browser does all the work. Most Web developers prefer to work with client-side image mapping, which does not have to send the x- and y-coordinates of the mouse click to the Web server to be interpreted. Instead, the coordinates are included in the HTML file along with the URL to which to link. When a visitor to a Web page clicks within a client-side image map, the browser processes the data without interaction with the Web server.

One advantage of server-side image mapping is that most, if not all, browsers support server-side image maps, while some older browsers do not support client-side image maps. Server-side image maps have disadvantages, however. They require additional software in order to run on a Web server. This requires that the server administrator maintain and update the server software on a regular basis. Also, an image map available on a particular Web site’s server must be registered to the server before it can

be used. Although this process is simple, it must be done. Further, all changes to that registered image map must be coordinated on the Web server, which does not allow for quick updates. Client-side image maps help reduce the load on the Web server, generally download faster, and provide faster response when a user clicks a link. In this chapter's project, you will create a client-side image map with four links on the home page of the Bright Idea Web site. You will learn more about client- and server-side image maps later in the chapter.

Understand the image map process.

Before inserting the graphical and color elements on a Web page, you should plan how you want to format them. By effectively utilizing graphics and color, you can call attention to important topics on the Web page without overpowering it. Creating a client-side image map for a Web page is a four-step process:

1. **Select an image to use as an image map.** Not all images are appropriate for good image mapping. Those images without distinct boundaries are not easy to map. Besides causing difficulty to the Web developer to find the points to plot, nondistinct areas make it difficult for visitors to see where one link might end and another begins. When choosing an image to map, choose wisely.
2. **Sketch in the hotspots on the image.** It is sometimes good to print a copy of the image and draw the hotspot areas on top of the paper image. You can then take that hard copy and review it while working with the image in the image editing software. When sketching (either on paper or in the software), determine what shapes (i.e., circle, rectangle, or polygon) make sense for the specific area that you want to link. Based on this determination, start the next step of plotting those areas on the image.
3. **Map the image coordinates for each hotspot.** This chapter explains what x- and y-coordinates you need to provide for every linkable area. One thing to consider is making sure that the linkable areas do not run over one another. This overrun ends up confusing your Web site visitors because they think they will link to one area, and the coordinates take them somewhere else.
4. **Create the HTML code for the image map.** Writing HTML code for an image map is different from anything that you have done thus far in the book. When you create an image map, you first insert the image itself and then identify the name of the map that you use later in the HTML code. Further down in the code, you actually use that name and identify the map areas that form the boundaries around the hotspot.

Plan Ahead

Creating an Image Map

An image map consists of two components: an image and a map definition that defines the hotspots and the URLs to which they link.

Selecting Images

Not all images are appropriate candidates for image mapping. An appropriate image, and a good choice for an image map, is one that has obvious visual sections. The United States map image shown in Figure 5–9a on the next page, for example, has distinct, easy-to-see sections, which serve as ideal hotspots. A user easily could select an individual area on the map to link to more information about each region. The image in Figure 5–9b, however, would not be a good choice because the boundaries of the states are indistinct.

BTW

Images

Not all images are appropriate for image mapping. An appropriate image has obvious visual sections that can be easily divided into clickable areas. An inappropriate image does not have obvious visual sections and therefore is not a good choice as an image map.

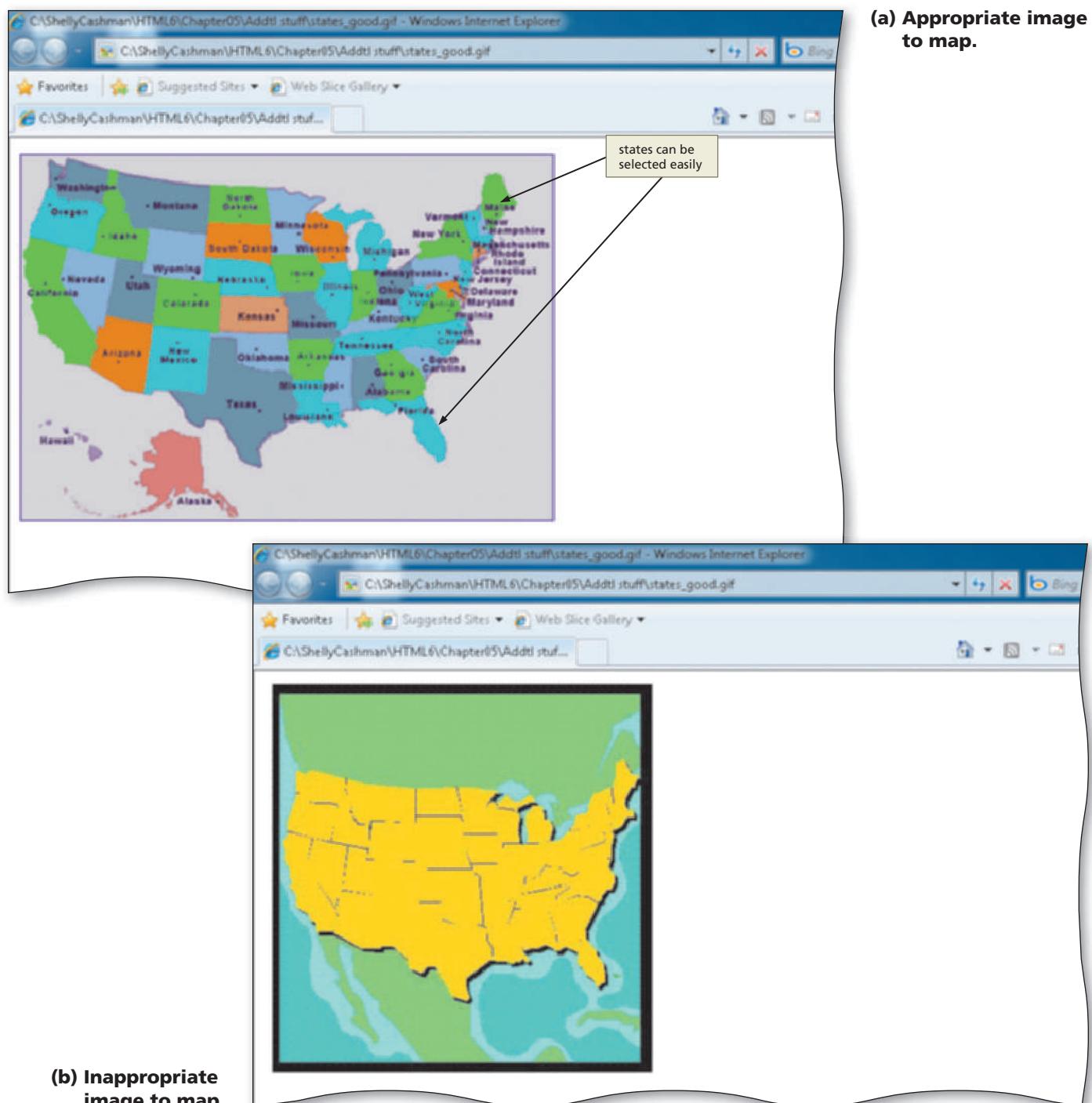


Figure 5–9

Sketching the Borders of Hotspots

After an appropriate image is selected for the image map, the next step is to sketch the hotspots (clickable areas) within the image. Figure 5–10 shows an example of an image map with the borders of the hotspots sketched on the image. A map of Europe is used, with two countries (Spain and Sweden) defined as hotspots. The image map thus will include a hotspot for two countries, each of which can link to a different Web page.

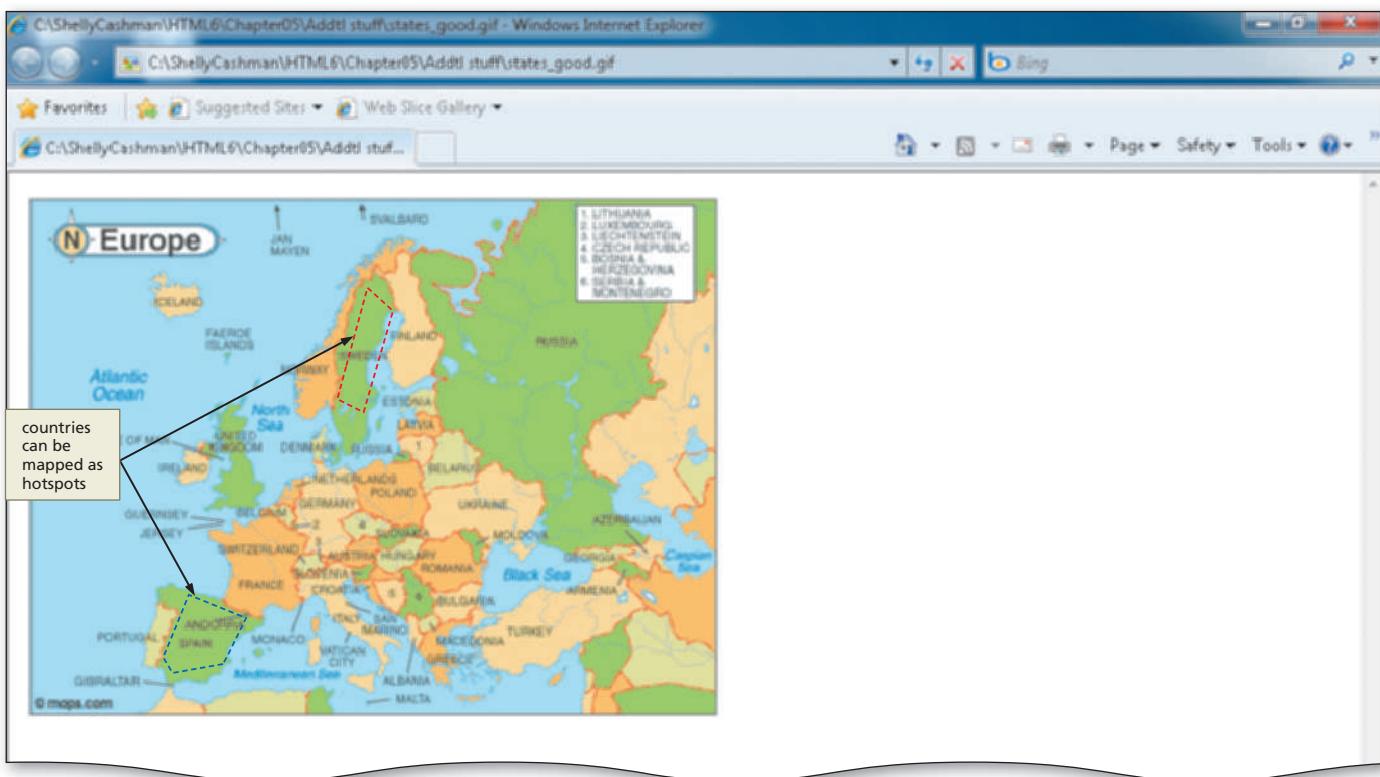


Figure 5–10 Sketched areas for image map hotspots.

Figure 5–11 shows an enlarged version of the Bright Idea menu bar image. It is made larger in this screenshot so that it is easier to show how you can sketch hotspots on this particular image. This menu bar image, menubar.jpg, is used as the image map in this chapter, and is shown here with the hotspots sketched in. The image is included in the Data Files for Students. Four circular shapes (labeled as Home, E-mail, Information, and Search) are defined as hotspots, which will link to an e-mail window or the Web pages that contain information about each option. The process of mapping the image coordinates for each hotspot is based on this initial sketch. The four icons used in this image menu bar are easily recognizable. The first link, Home, uses an icon that shows an outline of a house. This is commonly used for home page links on the Web. The second icon, an envelope, is also commonly used to depict an e-mail link. The third icon is a lowercase “i” which is often used for an Information link. The fourth icon is a magnifying glass, a symbol often used to depict a Search link.

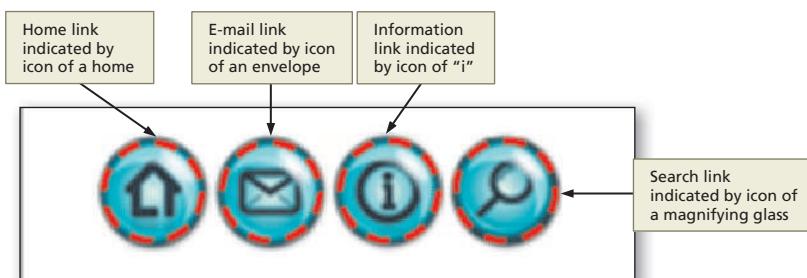


Figure 5–11 Sketched circular areas to be used as hotspots.

Mapping Image Coordinates

After you have determined how to divide the image into hotspot areas, you must determine the x- and y-coordinates for each of the sections. The x- and y-coordinates are based on a position relative to the x- and y-axes. The **x-axis** runs horizontally along the base of the image, while the **y-axis** runs vertically along the left of the image. The top-left corner of an image thus is the coordinate point (0,0), as shown in Figure 5–12. The first number of a **coordinate pair** is the x-coordinate, and the second number is the y-coordinate.

Figure 5–12 shows the starting (0,0) x- and y-coordinates in a Paint window that contains the image menubar.jpg. The y-coordinate numbers increase as you move the mouse pointer down the image, and the x-coordinate numbers increase as you move the mouse pointer to the right on the image. As you move the mouse pointer, the coordinates of its position as it relates to the image are displayed on the status bar.

You can use a simple or a sophisticated image editing or paint program to determine the x- and y-coordinates of various image points. In this project, the Paint program is used to find the x- and y-coordinates that you will use in the map definition that divides a single image into several areas.

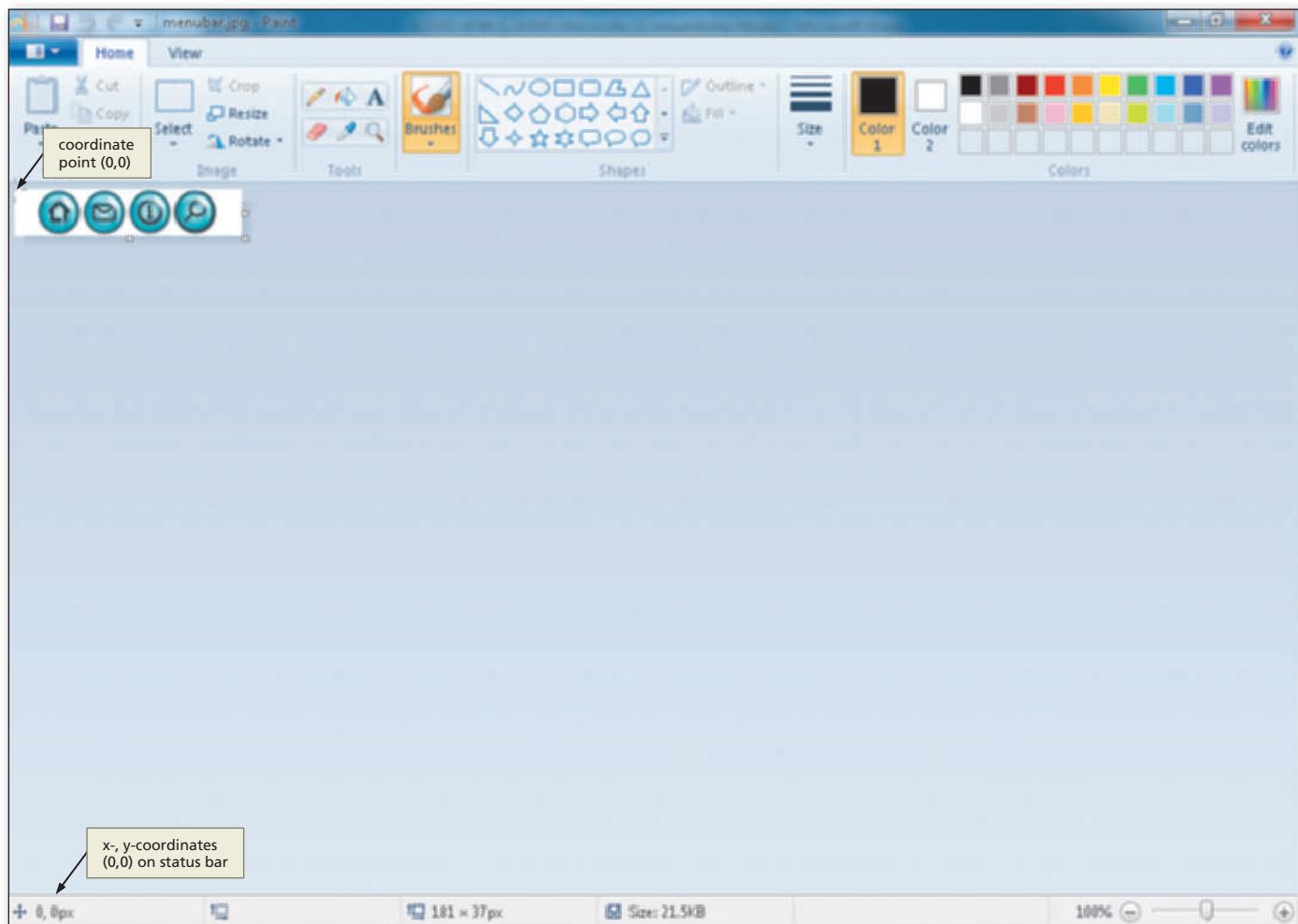


Figure 5–12

Map areas can use one of three shapes: rectangle, circle, or polygon. These shapes are shown in Figure 5–13. To define a map area of an image, you must determine the x- and y-coordinates for that shape and then insert the coordinates for the various map shapes in the HTML code.

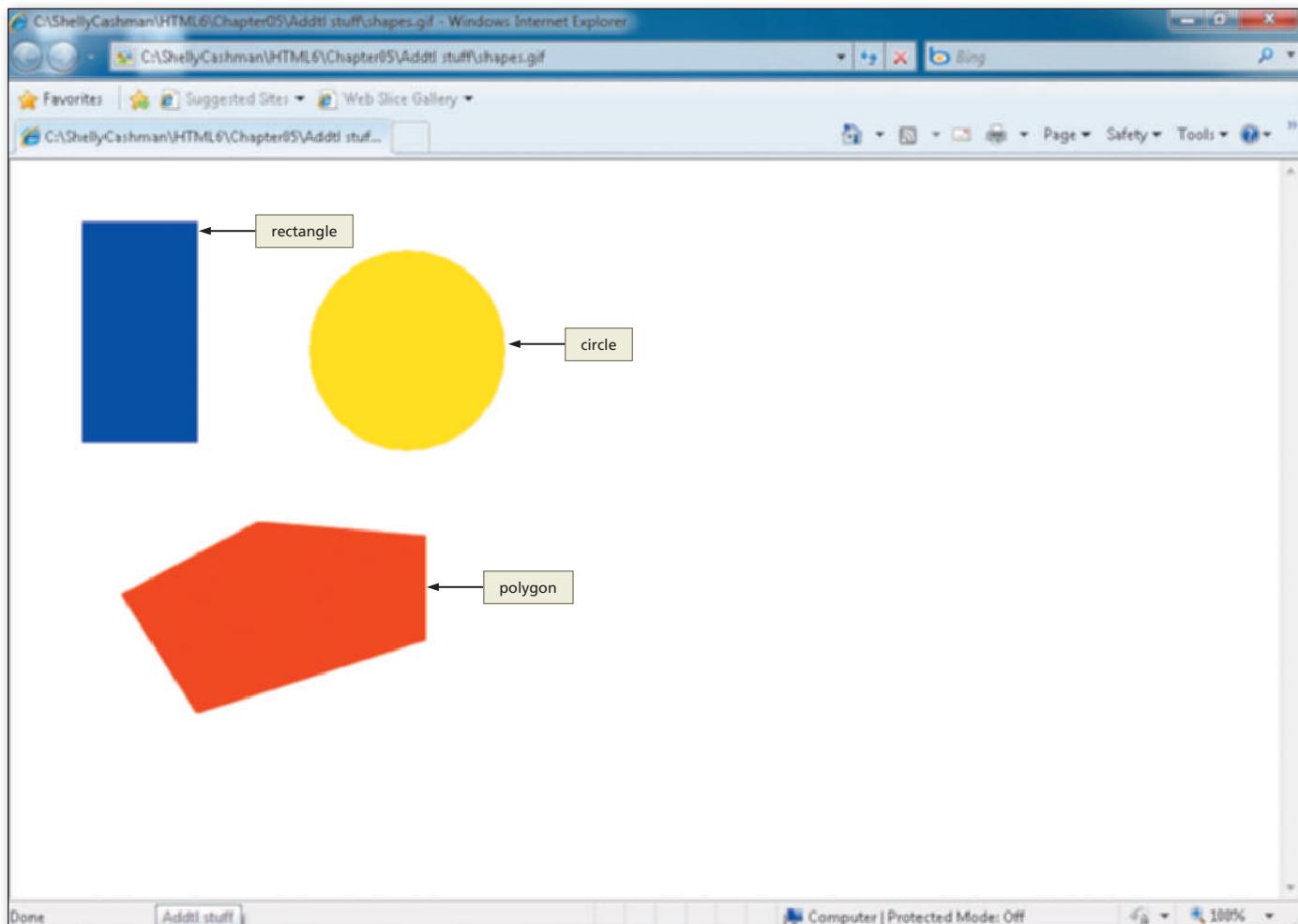


Figure 5-13

For a rectangular map area, you use the coordinates of the top-left and the bottom-right corners. For example, as shown in Figure 5–14a on the next page, the rectangle's x- and y-coordinates are (46,35) for the top-left corner and (137,208) for the bottom-right corner. You use “rect” as the value for the shape attribute for rectangles. For a circular map area, you use the center point and the radius as the coordinates. The x- and y-coordinates of the center point of the circle in Figure 5–14a are (308,113). If the mouse pointer is moved along the y-axis (113) to the border of the circle, the x-axis is 380. The radius can be calculated by subtracting the x-axis value of the center point (308) from the x-axis value of the circle's right border (380), which gives a radius of 72 (380 – 308). For circles, you use “circle” as the value for the shape attribute. For a polygonal map area, you must use the coordinates for each corner of the shape. For example, in Figure 5–14a, the polygon has five corners with the coordinates (78,309), (183,251), (316,262), (317,344), and (136,402). For polygonal shapes, you use “poly” as the value for the shape attribute. Figure 5–14b shows how you would use those x- and y-coordinates in the map statements needed to define these three shapes as clickable areas.

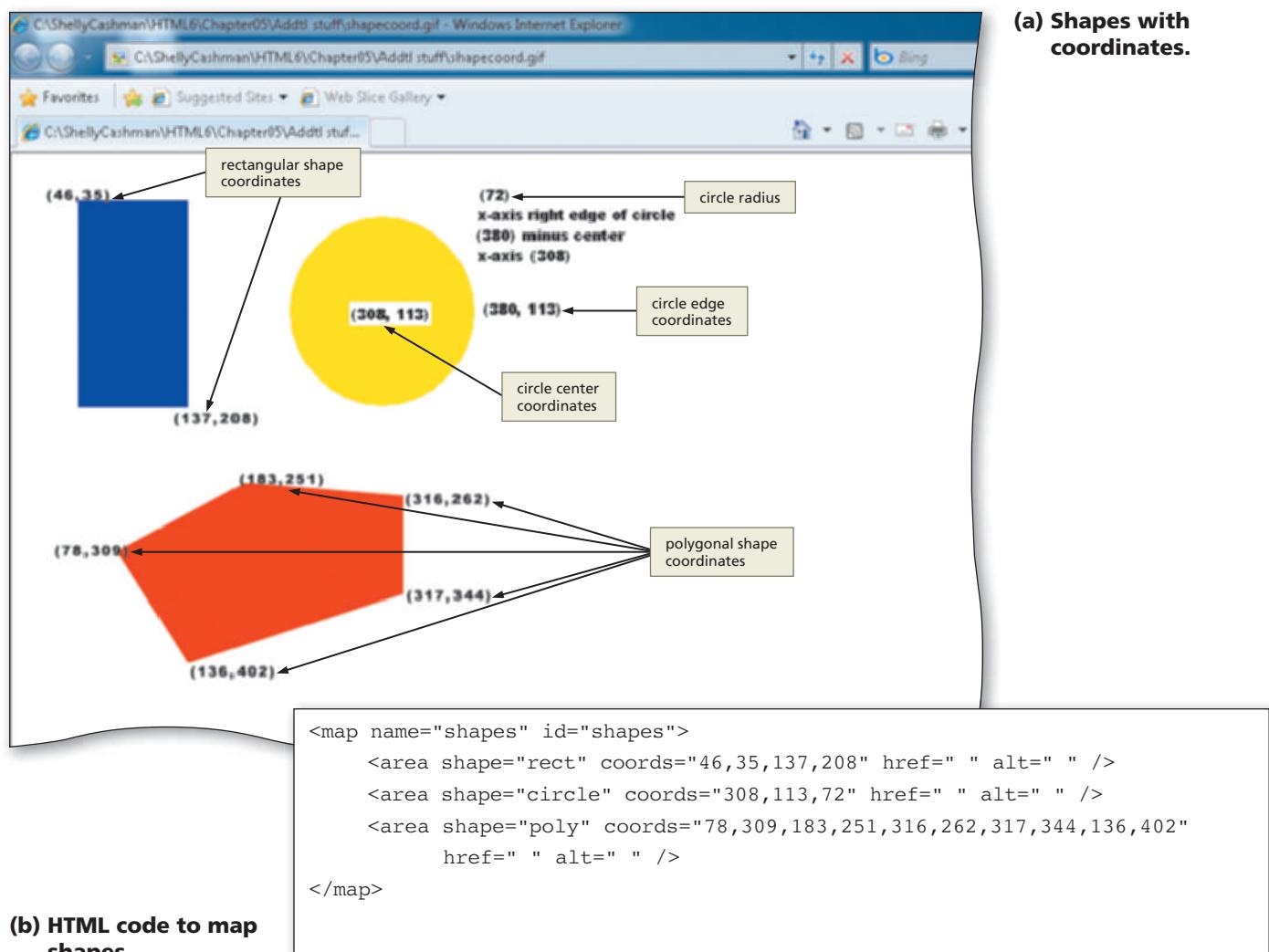


Figure 5-14

In the Bright Idea image (menubar.jpg), the image map will use four circular shapes for the four hotspots, as sketched in Figure 5-11 on page HTML 245. Clickable areas are mapped in circular shapes enclosing the following areas: Home, E-mail, Information, and Search.

Coding the Map

The final step in creating an image map is writing the HTML code for the map. To create a client-side image map, the tags `<map>` `</map>` and `<area>` are used. The map start tag (`<map>`) and map end tag (`</map>`) create the client-side image map. The `<area>` tag defines the specific areas of the map and the links and anchors for those areas. The x- and y-coordinates for each map area are inserted into the `<area>` tag with the **coords** attribute, within quotation marks and separated by commas. The HTML tags, attributes, and values needed to code the map are discussed later in this chapter.

Working with the image.

In order to determine the x- and y-coordinates for image map points, you need to open the image in the chosen software tool.

- **Select a software tool.** Computers running the Windows operating system already have an image editing tool available, Paint. This chapter shows you how to work with your image within Paint. For other suggested editing software products, see Table 5–2 on page HTML 254.
- **Edit the image.** It is sometimes necessary to alter the image before using it on the Web page (you may want to resize it or reposition it). Paint also gives you the image dimensions (i.e., width and height) you need for the tag.
- **Make other changes to the image.** In Paint, you can make other changes to the image such as flipping the image horizontally or vertically, or altering the colors of the image. Other graphic editing software options provide a variety of tools to alter an image slightly or significantly.

Plan Ahead

Using Paint to Locate X- and Y-Coordinates

As you have learned, you can use a simple or a sophisticated image editing or paint program to determine the x- and y-coordinates of various points on an image. In this chapter, the Paint program is used to find the x- and y-coordinates used in the map definition that divides a single image into several areas.

To Start Paint

The following steps illustrate how to start Paint.

1

- Click the Start button on the taskbar.
- Point to All Programs on the Start menu, click Accessories on the All Programs submenu, and then point to Paint on the Accessories submenu (Figure 5–15).
- Click Paint.

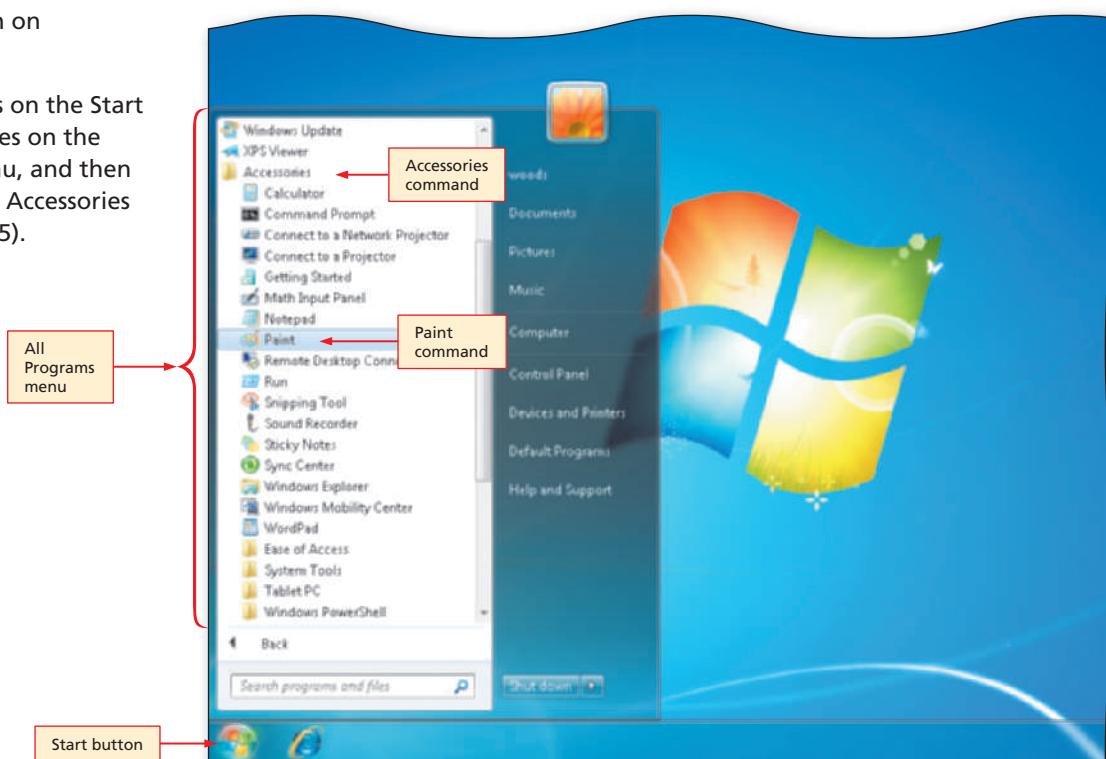


Figure 5–15

2

- If necessary, click the Maximize button on the right side of the title bar to maximize the window (Figure 5–16).

Q&A

Do all computers running Windows include Paint?

Yes, Paint should be included with all Windows operating systems.

Q&A

How can I find out more about using Paint?

The Paint Help utility is quite good. You can search for information using its Search option or Index. Paint Help gives step-by-step instructions for many tasks.

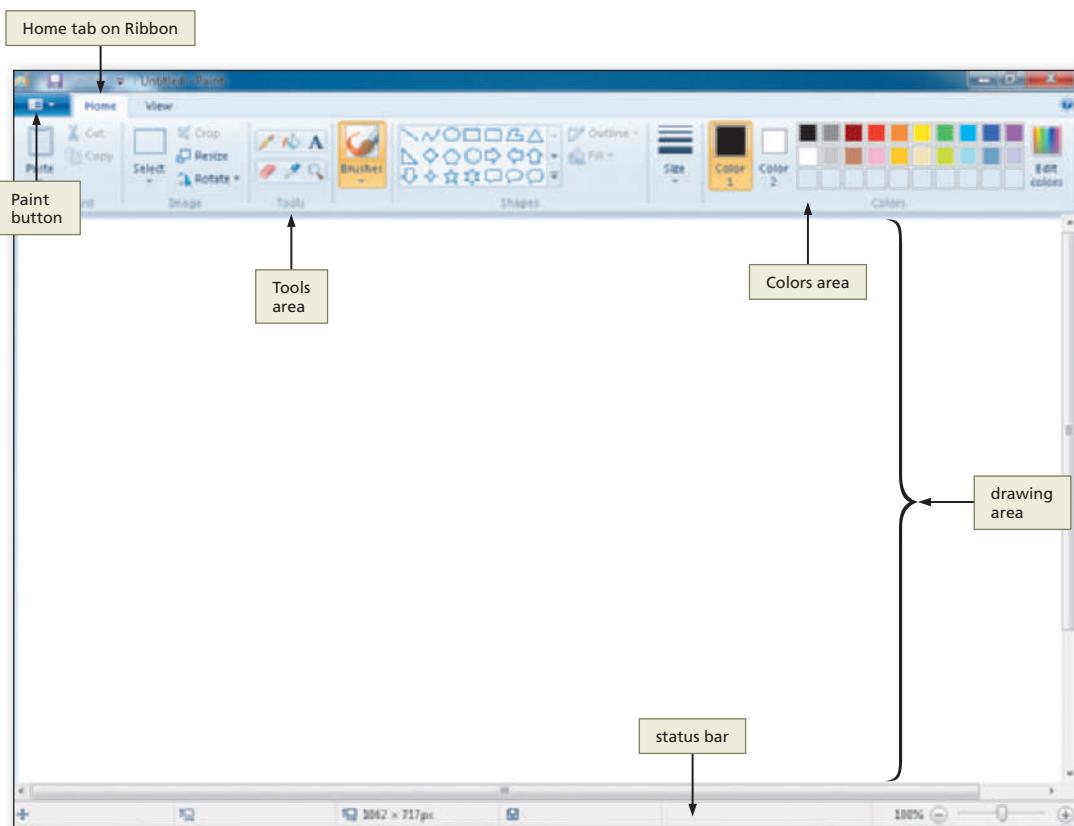


Figure 5-16

BTW**Paint Help**

The Help feature of Paint can answer your questions about the use of this popular tool. Paint can be used to identify the x- and y-coordinates in an image used as an image map. It also can be used to create images that are used as image maps.

The Paint Window

The Paint window contains several elements similar to the document windows in other applications. The main elements of the Paint window are the drawing area, the Home tab on the Ribbon, the Paint button, the Tools area, the Colors area, and the status bar, as shown in Figure 5–16.

Drawing area — The **drawing area** is where the image is displayed.

Home tab — The **Home tab** on the Ribbon holds tools and commands most frequently used to create and edit images.

Paint button — The **Paint button** to the left of the Home tab contains a drop-down arrow to access functions such as Open, Save, and Print that were on the File menu in earlier versions of Windows.

Tools area — The **Tools area** on the Home tab displays tools that are used to edit or draw an image. In this project, the Pencil tool in the Tools area is used to find the x- and y-coordinates of the menu bar image.

Colors area — The **Colors area** on the Home tab displays a palette of colors that can be used to set the colors of the foreground, the background, or other elements in a drawing.

Status bar — The **status bar** displays the coordinates of the center of the mouse pointer at its current position on the image.

To Open an Image File in Paint

The image file (menubar.jpg) used for the image map in this chapter is stored in the Data Files for Students. This image is a menu bar that includes circular icons (from left to right) that are used for the following navigation functions: Home, E-mail, Information, and Search. The following step illustrates how to open an image file in Paint.

1

- With a USB drive plugged into your computer, click the Paint button arrow and then click Open.
- If Computer is not displayed in the left-hand navigation window, scroll until Computer is displayed.
- Click Computer to display a list of available drives.
- If necessary, scroll until UDISK 2.0 (G:) or your drive appears in the list of available drives.
- Double-click the drive, then double-click the Chapter05 folder, and then double-click the ChapterFiles folder in the list of available folders.
- Click the menubar.jpg image, and then click the Open button in the Open dialog box to display the image that will be used for image mapping in this chapter, as shown in Figure 5–17.

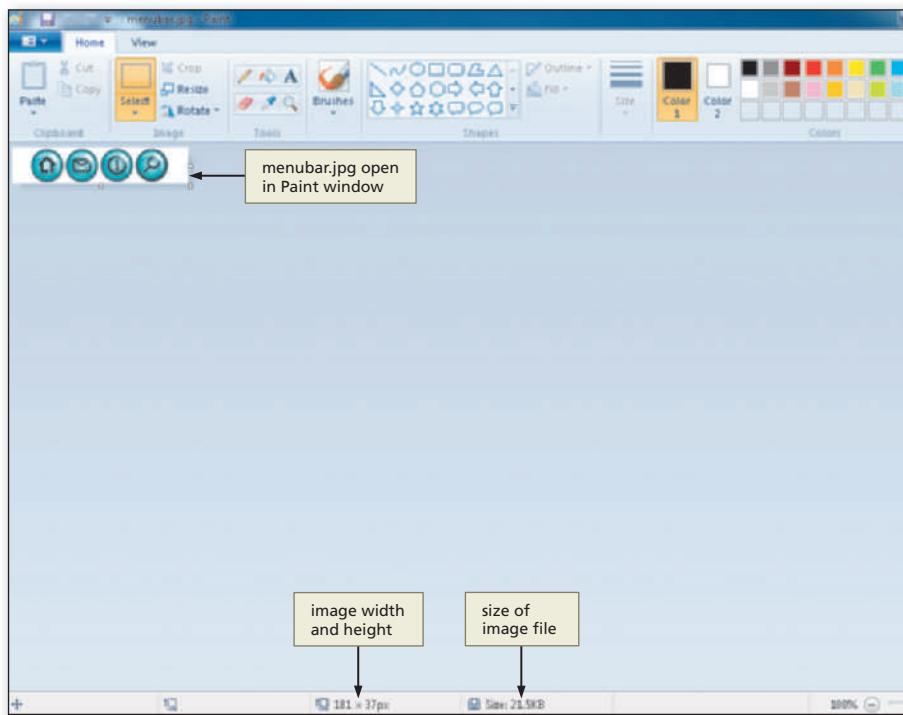


Figure 5–17

Locating X- and Y-Coordinates of an Image

The next step is to locate the x- and y-coordinates of the areas that should be mapped on the image. As shown in Figure 5–18, the image map should include four clickable circular areas that will link to other Web pages. For each of the four linkable map areas, the center x- and y-coordinate pair must be determined first. You then locate a point on the circumference of the circle to determine the circle radius.

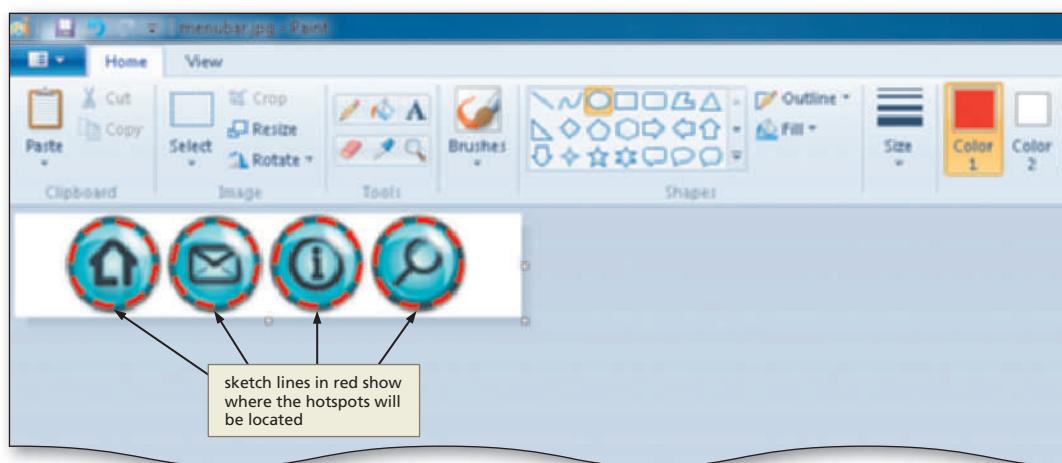


Figure 5–18

As you have learned, the x- and y-coordinates begin with (0,0) in the top-left corner of the image, as shown in Figure 5–12 on page HTML 246. As stated previously, moving the mouse pointer to the right (horizontally) increases the x-coordinate, and moving the mouse pointer down (vertically) increases the y-coordinate. Because the four clickable areas in the menu bar image sketched in red on the menubar.jpg image are circular, the map definition must include the x- and y-coordinate of the center of each circular shape. You then find a point on the edge of the circle and determine the radius of the circle. Those three numbers (x, y of center + radius) define the hotspot of the circular shape.

Table 5–1 shows the x- and y-coordinates for the four circular-shaped map areas. The first number is the x-coordinate, and the second number is the y-coordinate. For example, the Home circle consists of one pair of x- and y-coordinates in the center of the Home circle together with the number that depicts the circle radius. The first x-coordinate is 35 and the first y-coordinate is 18. You then find a point on the circumference of the circle, which gives you the next x-coordinate in the Home map shape. That x-coordinate point on the edge of the circle is 50, which makes the radius of that circle 15 (i.e., 50 – 35). For each circle, you find the x- and y-coordinates of the center of the circle. You then locate a point on the circumference of the circle, which will give you the radius. Notice that the radius (15) of each circular shape is the same (Table 5–1) because the four circular shapes are all the same size. These x- and y-coordinates are used in the <area> tag to create the map definition for the image map.

Table 5–1 Circle Center X- and Y-Coordinates and Circle Radius

Location	Coordinates and Radius
Home	35,18,15
E-mail	71,18,15
Information	107,18,15
Search	144,18,15

To Locate X- and Y-Coordinates of an Image

The following steps illustrate how to locate the x- and y-coordinates of the boundary points of each clickable circular area by moving the mouse pointer to the various points to see the x- and y-coordinates of those points. Although you do not need to record the coordinates for this project, you generally would do that. In this case though, you will compare the coordinates with those shown in Table 5–1, which lists the exact coordinates used in the <area> tags for this project.

1

- If necessary, click the Pencil button in the Tools area (Figure 5–19).

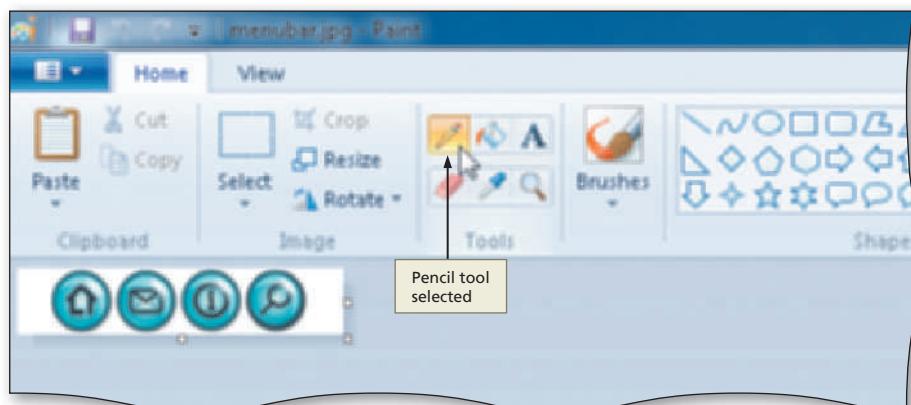


Figure 5–19

2

- Move the mouse pointer to the center of the Home icon circle and note the x- and y-coordinates for that point, as indicated in the status bar. Move the mouse until the coordinates read (35,18) (Figure 5–20a). (Do not click the mouse button.)

- Move the mouse pointer along the x-axis of the Home icon. The coordinates should read (50,18) (your coordinates may differ slightly), as indicated on the status bar (Figure 5–20b). (Do not click the mouse button.) Calculate the radius of the circle by subtracting the initial x-coordinate found in Figure 5–20a (35) from the x-coordinate found in Figure 5–20b (50). The radius is therefore 15. The three coordinates to be used for this circular hotspot are 35,18,15, as shown in Table 5–1.

- Move the mouse pointer to the center and edge of the three other circular shapes (E-mail, Information, and Search) hotspots on the menu bar image by following the x- and y-coordinates in Table 5–1 on page HTML 252.

- After you have finished, click the Close button on the right side of the title bar. If prompted, do not save any changes to the file.

Q&A

I am not sure of the purpose of this exercise because the coordinates are already given to us for the project. Why am I doing these steps using Paint?

For the purpose of the project, the coordinates are given. The normal image mapping process, however, consists of: finding an appropriate image, sketching out where you think the boundaries will be, and finding the coordinates on your own using a software tool that shows that information. The purpose of this exercise is to become familiar with using Paint to find the coordinates.

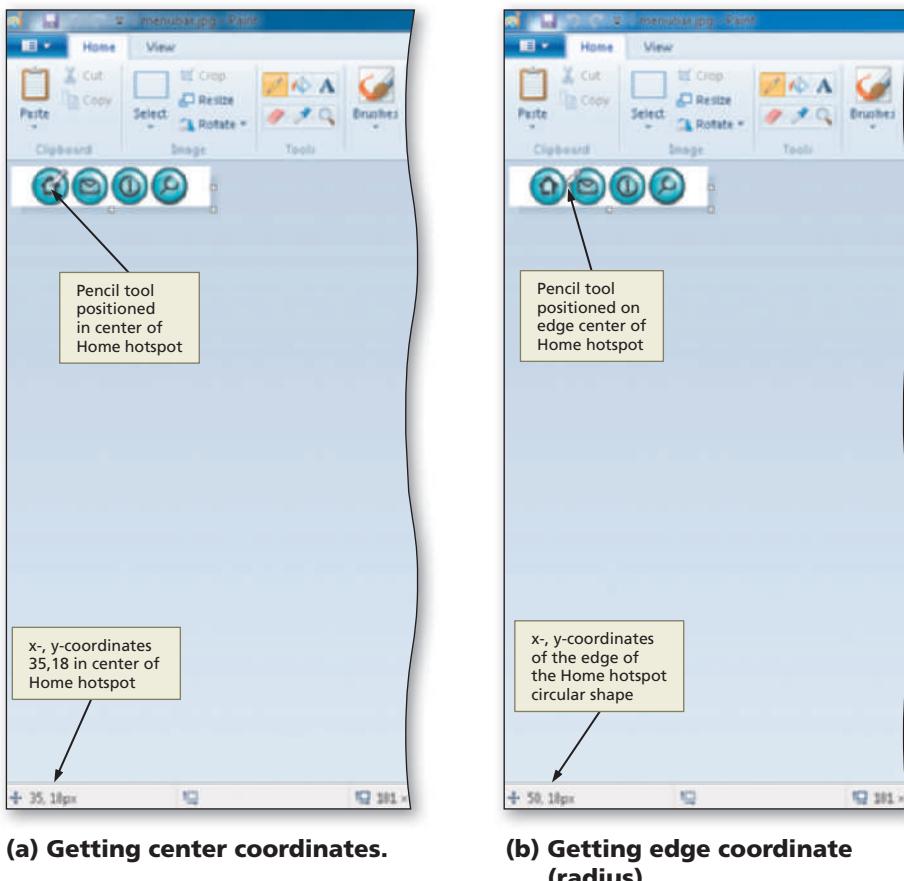
Q&A

I notice that in addition to the Pencil tool, I can use the Free-Form Select and the Select tools to show the x- and y-coordinates on the status bar. Is it okay to use them?

It is fine to use any of the three tools for this purpose. You are only trying to see the x- and y-coordinates for the hotspot areas.

Experiment

- Play with the Image and Colors areas on the Home tab. They give you many options to alter the image. Make sure to save the file with a different file name so that the image file included in the Data Files for Students folder remains untouched.

**Figure 5–20**

BTW

Graphics Software

Many graphics tools are available for advanced image creation and maintenance. One of the more popular tools is Adobe Photoshop. Photoshop allows you to create images that can be used as image maps or crop sections out of existing images.

Other Software Tools

Although Paint allows you to identify the coordinates for a map area manually, there are dedicated image map software tools that can simplify this process (see Table 5–2). These tools allow you to click the image to define the clickable areas of the image map and then automatically generate the x- and y-coordinates and HTML code needed for the image map. If possible, download one of the software tools listed in Table 5–2 and use that software to map the clickable areas in the menubar.jpg image. As further practice, open the file shapcoor.gif found in the Chapter05\ChapterFiles folder of the Data Files for Students in Paint (Figure 5–21) and use your mouse pointer to identify the coordinates to map the clickable areas in the shapcoor.gif image. You also could experiment with using one or more of the tools in Table 5–2 to map clickable areas in the image.

Table 5–2 Image Map Software Tools

Tool	Platform
Mapedit	Windows, UNIX, Mac OS
CoffeeCup Image Mapper	Windows
Imaptool	Linux/X-Window

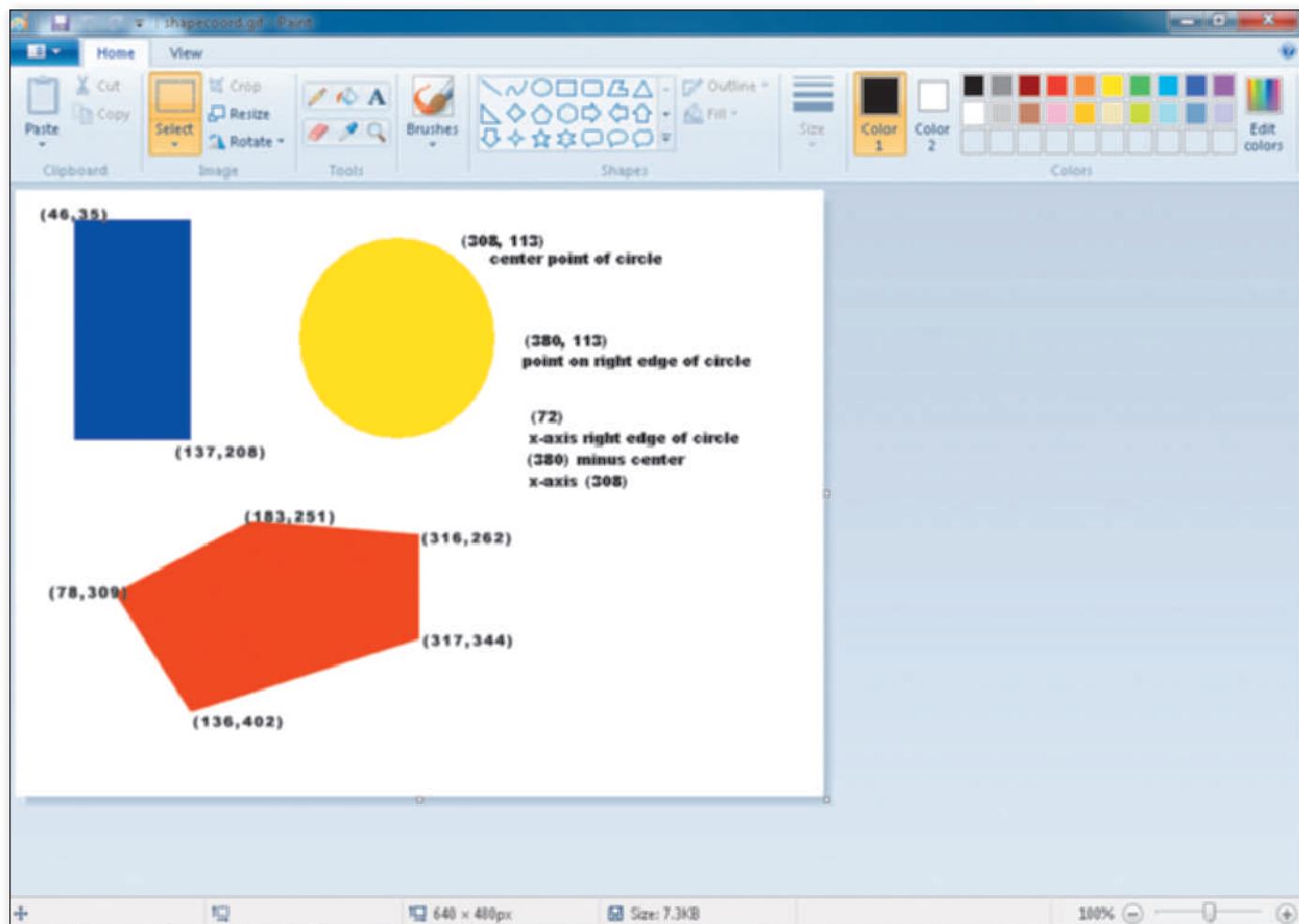


Figure 5–21

Starting the home page.

Just as with the other projects in previous chapters, you need to review good Web development standards before you start a new Web page.

- **Use the HTML structure tags required.** You will validate your Web pages for this project, so make sure that you use the HTML tags needed to make the page XHTML compliant. This includes using the `<meta>` tag and a DOCTYPE statement.
- **Copy what you can.** In earlier chapters, you copied HTML code from one completed page to another to make it easier. You should do the same in this project. Once a Web page is validated, you know that the initial HTML tags are correct. It makes sense then to copy/paste those lines of code to the next Web page file. If you are utilizing the same menu bar throughout a Web site, it also makes sense to copy that code from one Web page to another.

Plan Ahead

Creating the Home Page

Before the image map can be added to the home page of the Bright Idea Web site, the home page must be created. The home page includes a borderless table, a logo image, and paragraphs of text. At the bottom of the home page, a table of text links is inserted. The text links allow the Web page visitor to navigate to the home page, an e-mail link, the information.html Web page, and the search.html Web page.

To Start Notepad++

- 1 Click the Start button on the Windows taskbar to display the Start menu.
- 2 Click All Programs at the bottom of the left pane on the Start menu to display the All Programs list.
- 3 Click Notepad++ in the All Programs list.
- 4 Click Notepad++ in the list to display the Notepad++ window.
- 5 If the Notepad++ window is not maximized, click the Maximize button on the Notepad++ title bar to maximize it.
- 6 Click View on the menu bar.
- 7 If the Word wrap command does not have a check mark next to it, click Word wrap.

To Enter Initial HTML Tags to Define the Web Page Structure

To create the home page, you will start Notepad++ and enter the initial HTML tags to define the overall structure of the Web page, as shown in Table 5–3 on the next page. Notice that one additional line of code is added to this section of code. Line 10 is the link statement that links the external style sheet named styles5.css to the Web page. You will create the external style sheet later in this project.

Table 5–3 HTML Code to Define Web Page Structure

Line	HTML Tag and Text
1	<!DOCTYPE html
2	PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3	"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4	
5	<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
6	<head>
7	<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
8	<title>Bright Idea</title>
9	
10	<link rel="stylesheet" type="text/css" href="styles5.css" />
11	
12	</head>
13	
14	<body>
15	
16	
17	</body>
18	</html>

- 1 Enter the HTML code shown in Table 5–3. Press ENTER at the end of each line. If you make an error as you are typing, use the BACKSPACE key to delete all the characters back to and including the incorrect characters, and then continue typing.
- 2 Position the insertion point on the blank line 16 (Figure 5–22).
- 3 Compare what you typed to Figure 5–22. If you notice errors, use your mouse pointer or ARROW keys to move the insertion point to the right of each error and use the BACKSPACE key to correct the error.

```

new 1 - Notepad++
File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?
new 1
1 <!DOCTYPE html
2 PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5 <html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">
6   <head>
7     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
8     <title>Bright Idea</title>
9
10    <link rel="stylesheet" type="text/css" href="styles5.css" />
11
12   </head>
13
14   <body>
15
16
17   </body>
18 </html>
19

```

Figure 5–22

To Save an HTML File

With the initial HTML code for the Bright Idea home page entered, you should save the file. Saving the file frequently ensures you won't lose your work. Saving a file in Notepad++ also adds color to code that can help you identify different elements more easily.

- 1** With a USB flash drive connected to one of the computer's USB ports, click File on the Notepad++ menu bar and then click Save.
- 2** Type `brightidea.html` in the File name text box (do not press ENTER).
- 3** Click Computer in the left-hand navigation pane to display a list of available drives.
- 4** If necessary, scroll until UDISK 2.0 (G:) or the name of your storage device is displayed in the list of available drives.
- 5** Open the Chapter05\ChapterFiles folder.
- 6** Click the Save button in the Save As dialog box to save the file on the USB flash drive in the Chapter05\ChapterFiles folder with the name `brightidea.html`.

Creating a Table

The next task in developing the home page is to create a centered, borderless table with one row and three columns, as shown in Figure 5–23. The first data cell contains the image `brightidealogos.jpg`. The second data cell contains an `<h1>` heading with the company name Bright Idea. The third cell contains the special character that is the registered trademark symbol. You learn more about inserting special characters later in the chapter.

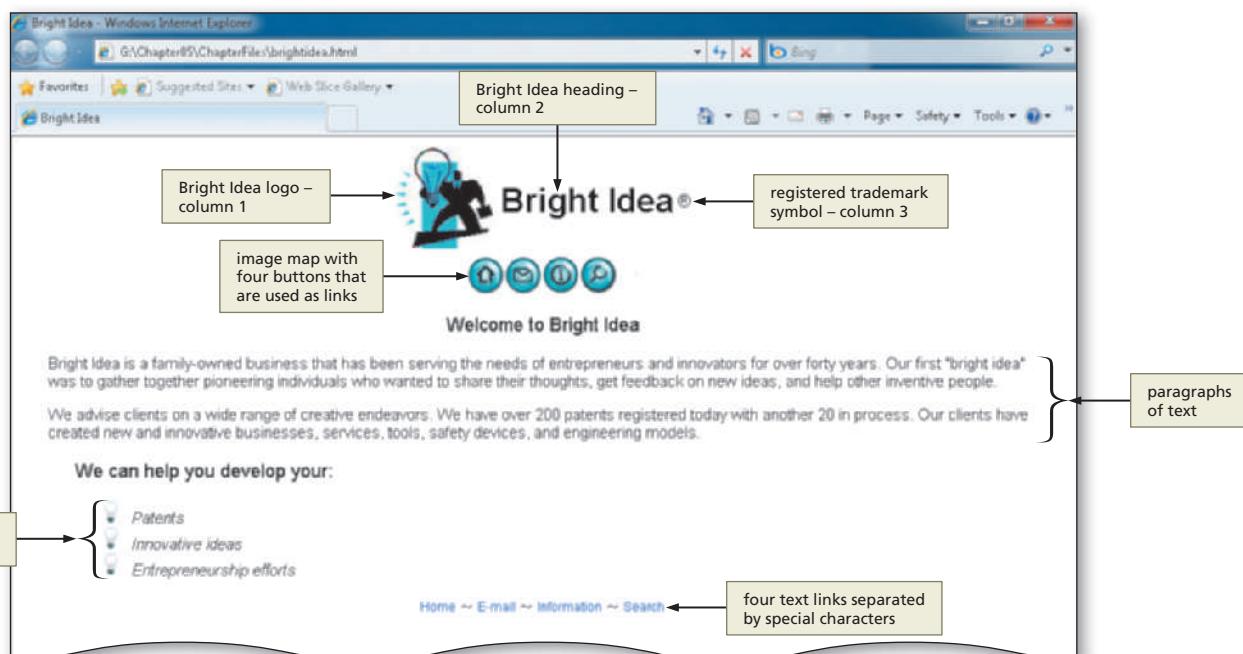


Figure 5–23

The three cells of the table are created using `<td>` tags that create table data cells. As you learned in Chapter 4, the `<td>` tag aligns the contents of a cell in the center of the cell vertically and to the left horizontally, by default. As shown in Figure 5–23, the table should use a vertical alignment so the contents of all cells are aligned with the top of

the cell. The HTML code thus should use a `<tr>` tag with the `valign="top"` attribute to create a table row that uses vertical alignment. Using this tag eliminates the need to set each table data cell to use vertical alignment.

To Create a Table

Table 5–4 shows the code to insert a table to hold the company logo.

Table 5–4 HTML Code to Insert Company Logo	
Line	HTML Tag and Text
16	<code><table style="margin-left: auto; margin-right: auto"></code>
17	<code> <tr></code>
18	<code> <td></code>
19	<code> </code>
20	<code> </td></code>
21	<code> <td></code>
22	<code> <h1>Bright Idea</h1></code>
23	<code> </td></code>
24	<code></table></code>

The following step creates a borderless table, with a table row that uses vertical alignment.

- With the insertion point on line 16, enter the HTML code, aligning as shown in Table 5–4, and press the `ENTER` key twice to position the insertion point on line 26 (Figure 5–24).

- Q & A** What happens when you set the `margin-left` and `margin-right` properties to `auto`?
Setting these properties to `auto` is how you center a table.

```

<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Bright Idea</title>
<link rel="stylesheet" type="text/css" href="styles5.css" />
</head>
<body>
<table style="margin-left:auto; margin-right:auto">
<tr>
<td>

</td>
<td>
<h1>Bright Idea</h1>
</td>
</tr>
</table>
</body>
</html>

```

Figure 5–24

Inserting Special Characters

The next step in creating the home page is to add the registered trademark symbol, as shown in Figure 5–23 on page HTML 257. The registered trademark symbol (®) is one of the Special Characters found in Appendix F. You use the ® entity to display the registered trademark symbol. These predefined entity characters enhance the content on the Web pages that you create. A registered trademark symbol is a distinctive indicator used by an individual, a business organization, or another type of legal entity to identify that the products or services with which the trademark appears originate from a unique source. This symbol indicates that the trademark has been federally registered. This symbol differs from the ™ (trademark) symbol in that the ™ symbol is one in which the individual or business either has a pending federal trademark application or they are simply claiming the rights to the mark. The entity for the ™ symbol is ™, as found in Appendix F.

You insert the ® entity in the third data cell of the table above. If you try to insert this symbol within the <h1> heading, the size of the symbol is the same as the heading size, which is too large (Figure 5–25a). If you try to insert the ® symbol just after the </h1> tag, it moves the symbol to a new line (Figure 5–25b). The best way is to insert this symbol in its own data cell in the table.

BTW

Using Special Characters

Some characters are reserved in HTML. If you want the browser to actually display these characters, you must insert the character entities in the HTML source. For example, you cannot use the greater than or less than signs within your text because the browser could mistake them for markup. You therefore use the special characters > (greater than) and < (less than) to display those characters.

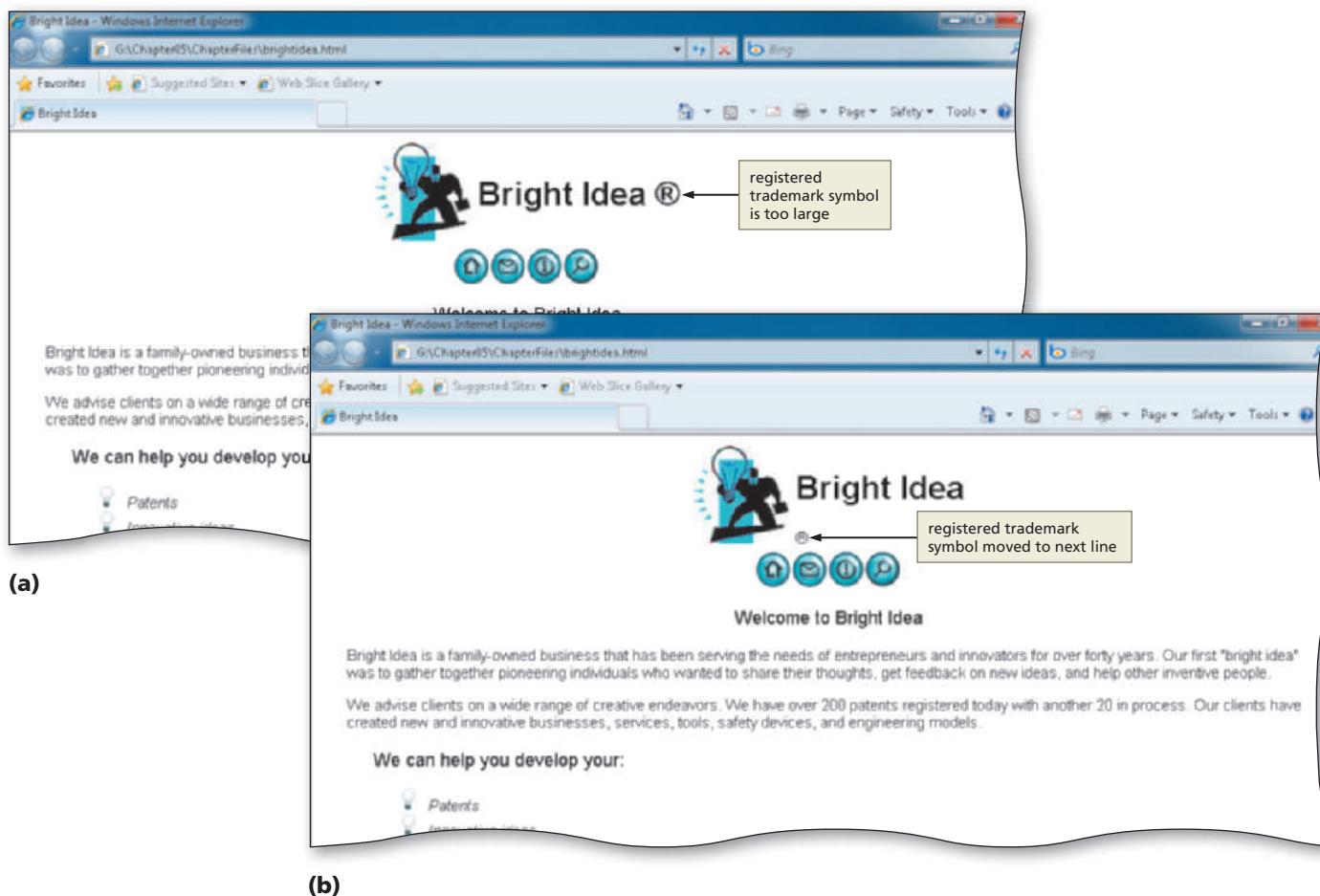


Figure 5–25

Also, in the first paragraph of text on the home page, you want to add quotation marks to your text by adding the " special character entity (see Appendix F). There are some browsers that may display the quotation marks if you insert the quotation marks themselves. In your Web development though, you need to account for browsers that would not display them unless you use the " symbol.

To Insert a Special Character

The following step inserts a registered trademark symbol.

1

- With the insertion point on line 26, press the TAB key twice and type <td>® and then press the ENTER key.
- Type </td> and then press the ENTER key.
- Press SHIFT-TAB once to move back a tab and type </tr> and then press the ENTER key.
- Press SHIFT-TAB to move back to the left margin and type </table> and press the ENTER key twice to position the cursor on line 31, as shown in Figure 5–26.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3 "http://www.w3.org/1999/xhtml">
4 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
5 <head>
6 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
7 <title>Bright Idea</title>
8
9 <link rel="stylesheet" type="text/css" href="style5.css" />
10
11
12 </head>
13
14 <body>
15
16 <table style="margin-left:auto; margin-right:auto">
17   <tr>
18     <td>
19       
20     </td>
21
22     <td>
23       <h1>Bright Idea</h1>
24     </td>
25
26     <td>&reg;</td>
27   </tr>
28 </table>
29
30 </body>
31 </html>

```

Figure 5–26

BTW

Image Map Tutorials

Many great resources are available on the Web that discuss image maps. For more information about tutorials, search for the term “image map tutorials” with any good search engine.

Inserting an Image to Use as an Image Map

The next step in creating the home page is to add the image, described in Table 5–5, which is used as the image map. The image, menubar.jpg, is stored in the Data Files for Students. This image consists of four circular icons that can be used as hotspots. Each icon has a picture of navigation function: Home, E-mail, Information, and Search. They are commonly used standards to identify navigation.

Table 5–5 shows the attributes associated with the `` tag. The `usemap` attribute, a client-side image map, is what you use in this project. A **client-side image map** is an image map that is run by the browser (the client) rather than by a CGI script on the Web server. You provide all the information required to run the map, the image, and the hotspot coordinates in your HTML document. The map’s functions are provided on the client’s side, rather than on the server’s side. When a visitor clicks a hotspot on the image map, the browser opens the corresponding URL.

A **server-side image map** uses CGI scripts to make the map work. The `ismap` attribute tells the browser to send the coordinates of the user mouse click directly to an associated map file on the server. If you look in Appendix C, Accessibility Standards and the Web, you will see that there is a reason to use client-side image maps instead of server-side image maps from a usability standpoint. The guideline in §1194.22(f) states, “Client-side image maps shall be provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape.” Also, the WAI Guidelines address this issue in guideline 9.1 where it says, “Provide client-side image maps instead of server-side image maps except where the regions cannot be defined with an available geometric shape.”

Table 5–5 Tag Attributes Used to Create Image Maps

Tag	Attribute	Function
	usemap ismap	<ul style="list-style-type: none"> Indicates the URL of a client-side image map Indicates a server-side image map

The Bright Idea home page will use a client-side image map. The HTML code to add the image thus will use attributes of the tag — src, width, height, and usemap — as follows:

```

```

where the src attribute identifies the image, and the width and height attributes define the image size.

The usemap attribute indicates to the browser which client-side image map will be used for that image. The client-side image map is placed within the <map> tag and defines the x- and y-coordinates of the areas on the image being used for the image map. When adding the image to use as an image map, the value of the usemap attribute — in this case, usemap="#menubar" — indicates that the browser should use the image map named menubar as its image map source.

BTW**Image Width
and Height**

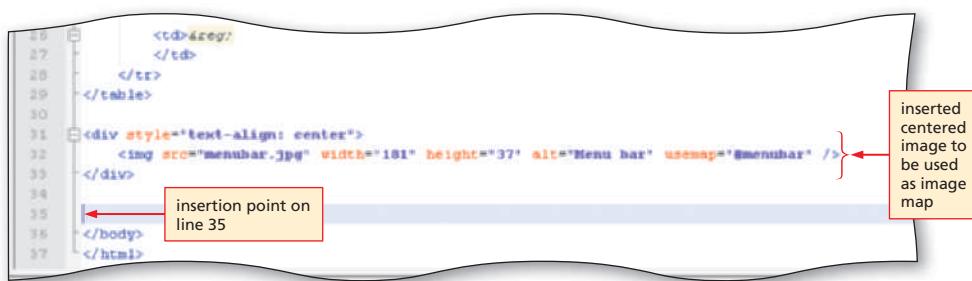
As you have learned in earlier projects, specifying the width and height attributes helps improve page loading time. By identifying these attributes in the HTML code, the browser does not have to determine the width and height of the image.

To Insert an Image to Use as an Image Map

The following step shows how to insert an image to use for the image map.

1

- If necessary, click line 31.
- Type <div style="text-align: center"> and then press the ENTER key.
- Press the TAB key once and type and then press the ENTER key.
- Press SHIFT+TAB to move back to the left margin, type </div>, and then press the ENTER key twice (Figure 5–27).

**Figure 5–27****Q&A**

I do not understand the purpose of the usemap attribute. Can you explain it?

The usemap attribute is what identifies the image with the map that will be inserted at the end of this Web page. The value (i.e., #menubar) in the usemap attribute tells the browser that this is an image map, and that it needs to look at the <map> tag name and id with that name (menubar) for the mapping.

Q&A

If I want to speed up the download of a large image, can I change the dimensions of the image using the width and height attributes to make it smaller?

Although you can do this, you should not. Making a change to an image with these attributes still forces the browser to download the entire image and then display it as you indicate in the width and height attributes. If you want to speed up the download by making the image smaller, you should use Paint (or some other image editing software) to change the dimensions and then save the image. In Paint, use the Resize button in the Image group on the Home tab.

To Add a Header and Paragraphs of Text

Now you add the header and paragraphs of text for the home page. An h2 heading would be too large, so you use an inline style to set the text to large. You use the "special character entity to add quotation marks to your text. The HTML code for this text is shown in Table 5–6.

Table 5–6 HTML Code for a Header and Paragraphs

Line	HTML Tag and Text
35	<p style="font-size: large; text-align: center">Welcome to Bright Idea</p>
36	
37	<p>Bright Idea is a family-owned business that has been serving the needs of entrepreneurs and innovators for over forty years. Our first "bright idea" was to gather together pioneering individuals who wanted to share their thoughts, get feedback on new ideas, and help other inventive people.</p>
38	
39	<p>We advise clients on a wide range of creative endeavors. We have over 200 patents registered today with another 20 in process. Our clients have created new and innovative businesses, services, tools, safety devices, and engineering models.</p>

The following steps show how to enter the tags for the heading and paragraphs of text.

- 1 If necessary, click line 35.
- 2 Enter the HTML code shown in Table 5–6 and then press the ENTER key twice (Figure 5–28).



Experiment

- Take out the " in front of and after the words "bright idea" in the first paragraph and put in regular quotation marks and save the file. Open the file in your browser to see if the quotation marks display. Depending on your browser, this may or may not show quotation marks. That is why using the " entity is better.

```

26 <td>
27 </td>
28 </tr>
29 </table>
30 <div id="header">
31 <h1>Bright Idea</h1>
32 
33 <map name="menubar">
34 <area alt="Home" href="#" title="Home" type="button" />
35 <p style="font-size: large; text-align: center">Welcome to Bright Idea</p>
36
37 <p>Bright Idea is a family-owned business that has been serving the needs of entrepreneurs and innovators for over forty years. Our first "bright idea" was to gather together pioneering individuals who wanted to share their thoughts, get feedback on new ideas, and help other inventive people.</p>
38
39 <p>We advise clients on a wide range of creative endeavors. We have over 200 patents registered today with another 20 in process. Our clients have created new and innovative businesses, services, tools, safety devices, and engineering models.</p>
40
41 </div>
42 </body>
43 </html>
44

```

Figure 5–28

To Add a Title and an Unordered List

The home page also contains a section title and an unordered (bullet) list that identifies the company's special talents. Table 5–7 shows the code for this. You use an image (bulletbulb.png) for the bullets in the list. The font-size of the title is set to the same size (large) as the previous title for consistency. You also insert margin-left styles to bring the section title and the unordered list in to the right. You use more margin space on the unordered list to account for the image bullet.

Table 5–7 HTML Code for Inserting an Unordered List

Line	HTML Tag and Text
41	<p style="font-size: large">We can help you develop your:</p>
42	
43	<div style="margin-left: 50pt">
44	<ul style="list-style-image: url(bulletbulb.png); font-size: 11pt; font-style: italic">
45	Patents
46	Innovative ideas
47	Entrepreneurship efforts
48	
49	</div>

The following steps show how to enter the tags for the unordered list.

- 1 If necessary, click line 41.
- 2 Enter the HTML code shown in Table 5–7 and then press the ENTER key twice (Figure 5–29).

```

1<div style="text-align: center">
2    
3</div>
4
5<p style="font-size: large; text-align: center">Welcome to Bright Idea</p>
6
7<p>Bright Idea is a family-owned business that has been serving the needs of entrepreneurs and innovators for over forty years. Our first "bright idea" was to gather together pioneering individuals who wanted to share their thoughts, get feedback on new ideas, and help other inventive people.</p>
8
9<p>We advise<br/>20 in process<br/>engineering m<br/>to range of creative endeavors. We have over 200 patents registered today with another<br/>we created new and innovative businesses, services, tools, safety devices, and<br/>engineering<br/>inventive<br/>people</p>
10
11<p style="font-size: large"><span style="margin-left: 20pt">We can help you develop your:</span></p>
12
13<div style="margin-left: 50pt">
14    <ul style="list-style-image: url(bulletbulb.png); font-size: 11pt; font-style: italic">
15        <li>Patents</li>
16        <li>Innovative ideas</li>
17        <li>Entrepreneurship efforts</li>
18    </ul>
19</div>
20
21</body>
22</html>

```

Figure 5–29

BTW

Text Links

It is very important to use text links on all Web pages in addition to using an image map for navigation. Some people turn graphics off while browsing the Web. If you did not have text links, those people could not access your other Web pages. In addition, you want to give your Web site visitors access to all Web pages in the Web site. Text links should provide that access.

BTW

Accessibility

Developers should utilize suggestions from accessibility guidelines when developing Web pages. It is important that the Web page content is understandable and navigable. The W3C gives some good suggestions for Web site accessibility issues that should be reviewed. For more information about accessibility, visit the W3C Web site, or refer to Appendix C, Accessibility Standards and the Web.

To Create a Horizontal Menu Bar with Text Links

The next step is to create a horizontal menu bar of text links at the bottom of the page that mirror the image map links. As previously discussed, it is important that a Web page include text links to all URLs in the image map, in the event the image does not download, a user is using a text reader of some sort, or a user's browser is set to not display images. Notice that you use another special character entity, the tilde (~), between the menu options, using the character entity ∼

Table 5–8 shows the HTML code used to create the horizontal menu bar. As shown in lines 54 through 57, the HTML code adds the menu bar to a data cell in the table.

Table 5–8 HTML Code for Creating a Horizontal Menu Bar

Line	HTML Tag and Text
51	<table style="margin-left: auto; margin-right: auto">
52	<tr>
53	<td>
54	Home ∼
55	E-mail ∼
56	Information ∼
57	Search
58	</td>
59	</tr>
60	</table>

The following steps show how to create the text links at the bottom of the home page.

1 If necessary, click line 51.

2 Enter the HTML code shown in Table 5–8 and then press the ENTER key twice (Figure 5–30).

Q&A

I notice that we use a horizontal menu bar for many projects in the book. Are there other ways to display a menu?

Many different ways exist to display your menu. The horizontal menu bar is used because it makes sense aesthetically in these projects. A great idea is to review other menu bar options on the Internet and view the HTML source. You can get a lot of ideas by looking at the Web pages and source code from other Web developers. Remember that the whole point of the menu bar is to provide easy navigation for your Web site visitors.

```

<div>
<div>
<table style="margin-left: auto; margin-right: auto">
<tr>
<td>
<a href="brightidea.html">Home</a> ~
<a href="mailto:brightidea@isp.com">E-mail</a> ~
<a href="information.html">Information</a> ~
<a href="search.html">Search</a>
</td>
</tr>
</table>
</div>
</div>

```

Figure 5–30

Creating an image map.

This is the final step in the four-step process of image mapping. The HTML code is very specific about what is required for image mapping. It only takes one coordinate that is not correct or one shape that is wrong for the image map not to work as intended.

- **Use the <map> tag.** The <map> tag identifies the name and ID for the image map. It is important that the name is spelled correctly, and that the same name is used in the usemap attribute in the tag.
- **Use the <area> tag.** The <area> tag also is very important in image mapping. You identify the area shape and the x- and y-coordinates in this tag. Again, if even one number is typed incorrectly, it can make the image map nearly unusable. Image mapping software (described on page HTML 254) makes this a moot point because it inserts the coordinates for you into the HTML code.

Plan Ahead

Coding the Image Map Using HTML Tags and Attributes

Thus far, the chapter has addressed three of the four steps in creating an image map: the menubar.jpg image to use as an image map has been selected and added to the home page; the hotspots have been sketched on the menubar.jpg image; and Paint was used to locate the x- and y-coordinates for each map area on the image that serves as a hotspot. With these steps completed, the final step is to code the image map using HTML. Table 5–9 shows the two HTML tags used to create an image map, along with several key attributes of each.

Table 5–9 Tags and Tag Attributes Used to Create Image Maps

Tag	Attribute	Function
<map> </map>		<ul style="list-style-type: none"> • Creates a client-side image map
	name	<ul style="list-style-type: none"> • Defines the map's name
<area>		<ul style="list-style-type: none"> • Defines clickable areas within a <map> element, as well as links and anchors
	shape	<ul style="list-style-type: none"> • Indicates the shape of the map area; possible values are rect, poly, and circle
	coords	<ul style="list-style-type: none"> • Indicates the x- and y-coordinates of the points bounding the map area
	href	<ul style="list-style-type: none"> • Indicates the link (URL) used for a map area
	alt	<ul style="list-style-type: none"> • Indicates the alternate text for the image

The start <map> tag and end </map> tag define the section of code that includes the client-side image map. The <area> tag is used to define the clickable areas on the image map. An example of the <area> tag is:

```
<area shape="circle" coords="107,18,15" href="information.html"
      alt="Info" />brightidea.html
```

where the **shape** attribute with the **circle** value defines the clickable map area as a circle. Other possible values for the shape attribute are poly (polygon) and rect (rectangle). The alt attribute defines alternate text for the image. The **coords** attribute indicates the pairs of x- and y-coordinates of the center of the circle that serve as the starting point of the linkable area. In a circle, you next have to determine the radius of the circle. You do that by selecting a point on the circumference (edge) of the circle by moving along the x-axis to a point on the edge of the circle. You then subtract that x-axis coordinate from the x-axis coordinate in the center of the circle. Finally, the href attribute designates the URL of the link. In this example, a Web page visitor clicking anywhere within the circle bordered by the center x,y (107,18) and anywhere along the edge of the circle with the circle radius of 15 will link to the Web page information.html.

To insert the `<area>` tag for the circle, rectangle, and polygon shapes, such as those shown in Figure 5–14 on page HTML 248, the HTML code would be as follows:

```
<area shape="circle" coords="308,113,72" href="circle.html">
<area shape="rect" coords="46,35,137,208" href="rect.html">
<area shape="poly" coords="78,309,183,251,316,262,317,344,
136,402" href="poly.html">
```

To Create an Image Map

For the image map on the Bright Idea home page, four clickable areas are created: Home, E-mail, Information, and Search. All four clickable areas are circular in shape. Table 5–10 shows the HTML code used to create the image map for the menubar.jpg image on the home page. Line 62 defines the name of the image map as menubar, which is the name referenced in the `usemap` attribute of the `` tag that added the menubar.jpg image. Lines 63 through 66 define the four circular map areas for the image map, based on the x- and y-coordinates listed in Table 5–1 on page HTML 252. Each circular map area links to one of the three other Web pages on the Web site or to the e-mail link.

Table 5–10 HTML Code for Creating an Image Map

Line	HTML Tag and Text
62	<code><map name="menubar" id="menubar"></code>
63	<code><area shape="circle" coords="35,18,15" href="brightidea.html" alt="Home" /></code>
64	<code><area shape="circle" coords="71,18,15" href="mailto:brightidea@isp.com" alt="Email" /></code>
65	<code><area shape="circle" coords="107,18,15" href="information.html" alt="Info" /></code>
66	<code><area shape="circle" coords="144,18,15" href="search.html" alt="Search" /></code>
67	<code></map></code>

The following step illustrates how to enter the HTML code to create the image map for the menubar.jpg image.

1

- If necessary, click line 62.
- Enter the HTML code shown in Table 5–10 and press the ENTER key once (Figure 5–31).

Q&A

For this project, I am using all circular shapes. Could I have used other shapes for these four clickable areas?

For these menu buttons, there really is no other shape that could be used. You may think to use a polygon shape for the Home link, for instance, but that would make the clickable area too small.

Q&A

Could I have used other x- and y-coordinates for this image map?

Yes. This is a very subjective part of image mapping. You need to select the points in the boundaries that make sense to you. Just make sure that the points also will make sense to your Web page visitors. Also, take care not to overlap the points or you will end up with false results.

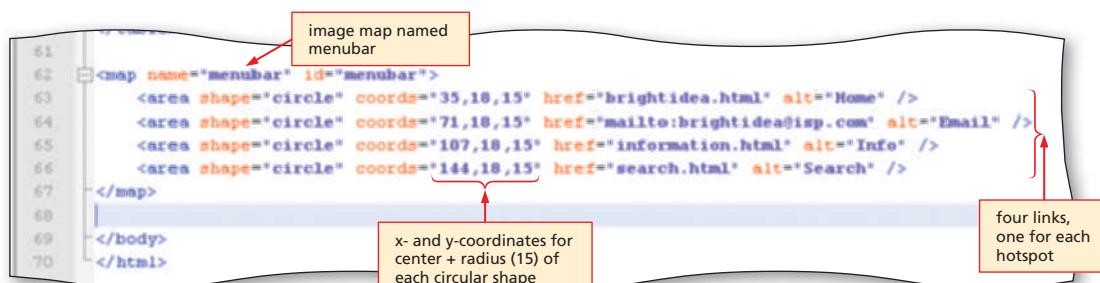


Figure 5–31

To Save the HTML File

With the HTML code for the Bright Idea home page complete, you should re-save the file.

- 1** Click the Save icon on the Notepad++ tool bar to save the most recent version of brightidea.html on the same storage device and in the same folder as the last time you saved it.

To Validate a Web Page

- 1** Open Internet Explorer and navigate to the Web site validator.w3.org.
- 2** Click the Validate by File Upload tab.
- 3** Click the Browse button.
- 4** Locate the brightidea.html file on your storage device and click the file name.
- 5** Click the Open button in the Choose File to Upload dialog box and the file name will be inserted into the File box.
- 6** Click the Check button.

Viewing the Web Page and Testing Links

After you save the HTML file for the Bright Idea home page, it should be viewed in a browser to confirm the Web page appears as desired. It also is important to test the four links on the Bright idea home page to verify they function as expected.

To View a Web Page

- 1** In Internet Explorer, click the Address bar to select the URL on the Address bar.
- 2** Type `g:\Chapter05\ChapterFiles\brightidea.html` on the Address bar of your browser and press **ENTER** to display the Web page (Figure 5–32 on the next page).

Q&A

Why do the Information and Search links work already?

These links work because the files information.html and search.html are stored in the Chapter05\ChapterFiles folder of the Data Files for Students.

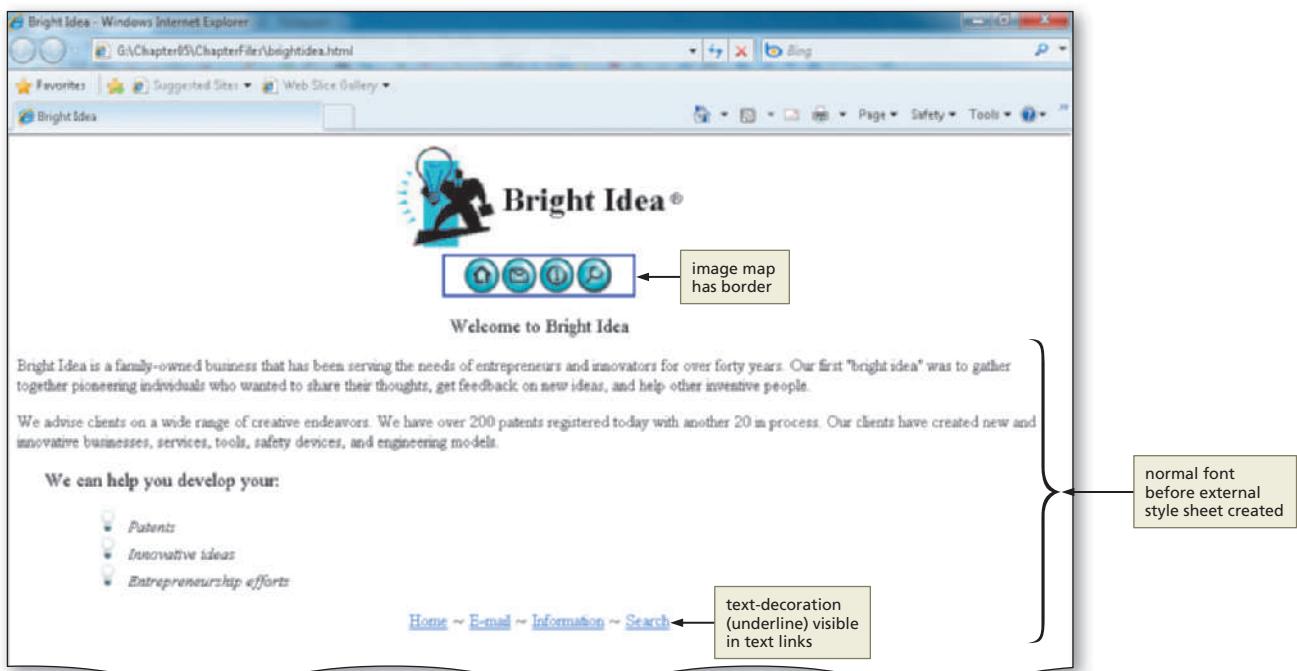


Figure 5-32

BTW**Testing**

Especially with image maps, it is important to test the Web page thoroughly in the browser. If one incorrect number is typed as an x- or y-coordinate, the entire image map can be wrong as a result. Make sure that the clickable area is exactly where you want it to be by testing your Web pages.

To Test Links on a Web Page

- 1** With the home page displayed in the browser, point to the e-mail link, brightidea@isp.com and click the link to open the default e-mail program with the address brightidea@isp.com in the To: text box.
- 2** Click the Close button in the New Message window. If a dialog box asks if you want to save changes, click No.
- 3** With the USB flash drive in drive G, click the Information link from the home page just created. Click back to the Home page from there, using either the image map or the text link. Next, click the other link to test the additional Web page provided in the Data Files for Students ([search.html](#)). Test the links back to the Home page from those Web pages.

To Print an HTML File

After your HTML code has passed validation, it is a good idea to make a hard copy printout of it.

- 1** Click the Notepad++ button on the taskbar to activate the Notepad++ window.
- 2** Click File on the menu bar and then click the Print command, and then click the Print button to print a hard copy of the HTML code (Figure 5-33).

```

<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Bright Idea</title>

<link rel="stylesheet" type="text/css" href="styles5.css" />

</head>

<body>

<table style="margin-left:auto; margin-right:auto">
<tr>
<td>

</td>

<td>
<h1>Bright Idea</h1>
</td>

<td>&reg;
</td>
</tr>
</table>

<div style="text-align: center">

</div>

<p style="font-size: large; text-align: center">Welcome to Bright Idea</p>

<p>Bright Idea is a family-owned business that has been serving the needs of entrepreneurs and innovators for over forty years. Our first "bright idea" was to gather together pioneering individuals who wanted to share their thoughts, get feedback on new ideas, and help other inventive people.</p>

<p>We advise clients on a wide range of creative endeavors. We have over 200 patents registered today with another 20 in process. Our clients have created new and innovative businesses, see your:</span></p>

<div style="margin-left: 20px; border-left: 1px solid black; padding-left: 10px; width: 300px; height: 150px; overflow: auto; vertical-align: top">
<ul style="list-style-type: none; padding-left: 0; margin: 0; font-size: small; color: #ccc; font-weight: bold; margin-bottom: 10px">
<li>Home</li>
<li>E-mail</li>
<li>Information</li>
<li>Search</li>
</ul>
<div style="border-top: 1px solid black; padding-top: 5px; margin-top: 5px; font-size: small; color: #ccc; font-weight: bold; margin-bottom: 10px">
<table style="margin-left: auto; margin-right: auto">
<tr>
<td>
<a href="brightidea.html">Home</a> &sim;
<a href="mailto:brightidea@isp.com">E-mail</a> &sim;
<a href="information.html">Information</a> &sim;
<a href="search.html">Search</a>
</td>
</tr>
</table>
</div>
<map name="menubar" id="menubar" style="text-decoration: none">
<area shape="circle" coords="35,18,15" href="brightidea.html" alt="Home" />
<area shape="circle" coords="71,18,15" href="mailto:brightidea@isp.com" alt="Email" />
<area shape="circle" coords="107,18,15" href="information.html" alt="Info" />
<area shape="circle" coords="144,18,15" href="search.html" alt="Search" />
</map>
</div>
</body>
</html>

```

Figure 5-33

Plan Ahead

Planning an external style sheet.

The home page is complete, but some style details are lacking. It is important to maintain a consistent look across a Web site. By creating an external style sheet, you can maintain that consistent look with great style.

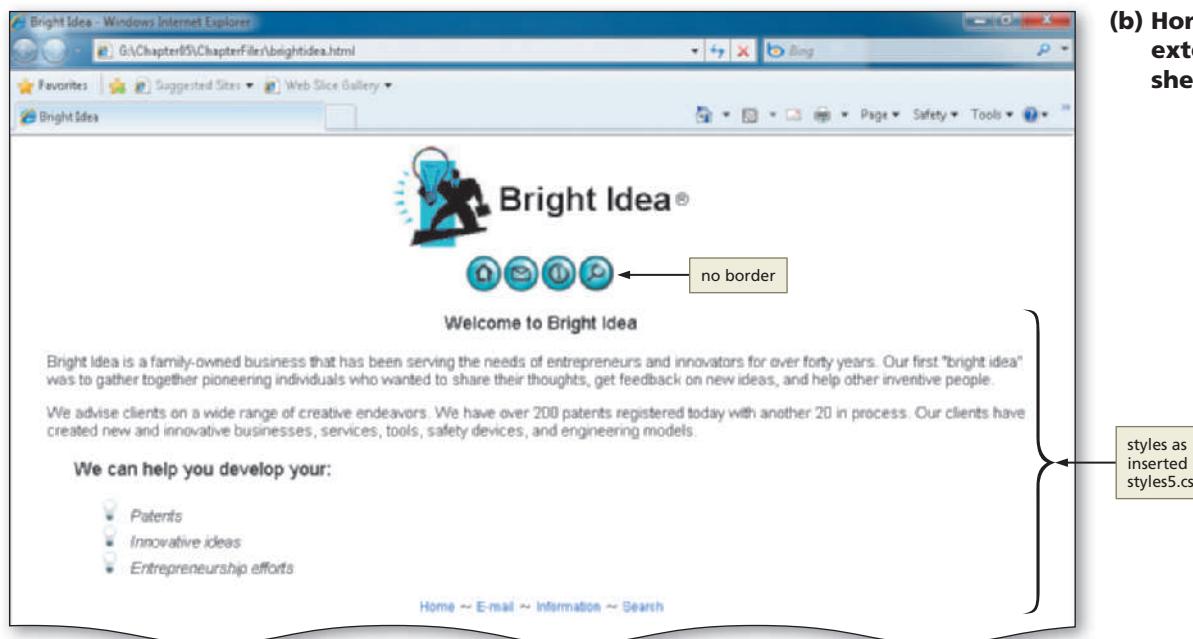
- **Determine what styles you want to use.** As you have seen, there are many styles that you can use to enhance your Web site (see Appendix D). Some that you can utilize in this Web site are:
 - Font family – maintaining a consistent font family is very important because the style of font is what you see the most across a Web site.
 - Font size – for the same reason as above, you should maintain consistency in font size; there are exceptions to this, especially in titles on the Web page.
 - Margins – it is helpful to utilize margins to help define the structure of a Web page.
 - Borders around links – you do not always want to see borders around images that you use as links; setting the border to zero is sometimes better style.

Creating an External Style Sheet

With the home page complete, the next step is to create an external style sheet that is linked to all pages in the Web site. You already added the link statement into the HTML code in the brightidea.html file on line 10. This statement tells the browser to link to the external (linked) style sheet named styles5.css. If there is no styles5.css file available, as is the case at this time, then the styles used are the default styles, as shown in Figure 5–34a. Once you create the styles5.css external style sheet, your Web page looks like that shown in Figure 5–34b.



Figure 5–34

**Figure 5–34 (continued)**

To create an external style sheet, you first have to start a new file in Notepad++. Remember from Chapter 4 that an external style sheet does not need the initial HTML DOCTYPE tags that you use for all Web pages. The external style sheet contains only the code for the various styles that you want to use across a Web site.

To Create an External Style Sheet

Table 5–11 contains the CSS code for the external style sheet that is used in all Web pages. Lines 1 and 2 define the font-family and font-size for the Web page. Lines 4 and 5 set the right and left margins to 20 points. That will give 20 points of space from the edge of the Web page to the start of the text in a paragraph. Lines 8 and 11 set the font-size for links to nine point with no text-decoration (no underline). Line 13 sets the border around images to zero.

Table 5–11 Code for External Style Sheet

Line	CSS Properties and Values	
1	body	{font-family: Arial, Verdana, Garamond;
2		font-size: 11 pt}
3		
4	p	{margin-left: 20pt;
5		margin-right: 20pt;
6		font-size: 11pt}
7		
8	a	{font-size: 9pt;
9		margin-left: auto;
10		margin-right: auto;
11		text-decoration: none}
12		
13	img	{border: 0px}

The following step shows how to create an external style sheet.

1

- If necessary, click the Notepad++ button on the taskbar.
- Click the New button on the toolbar.
- Enter the HTML code shown in Table 5–11 (Figure 5–35).

The screenshot shows the Notepad++ interface with a file named 'style.css' open. The code defines styles for the body, p, a, and img elements. Red callout boxes with arrows point to specific parts of the code:

- A box points to the 'body' rule with the text "font styles for entire Web page in body element".
- A box points to the 'p' rule with the text "20-pt margins on left and right of paragraphs".
- A brace groups the 'a' rule and the 'img' rule, with a box pointing to it labeled "styles for all links".
- A box points to the 'img' rule with the text "no borders around images".

```

1 body {font-family: Arial, Verdana, Garamond;
2   font-size: 11 pt;
3
4   p {margin-left: 20pt;
5     margin-right: 20pt;
6     font-size: 11pt;
7
8     a {font-size: 9pt;
9       margin-left: auto;
10      margin-right: auto;
11      text-decoration: none;
12
13   img {border: 0px}

```

Figure 5–35

To Save and Print the CSS File

- With a USB drive plugged into your computer, click File on the menu bar and then click Save As. Type `styles5.css` in the File name text box.
- If necessary, click UDISK (G:) or your storage device in the Save in list. Click the Chapter05 folder and then double-click the ChapterFiles folder in the list of available folders. Click the Save button in the Save As dialog box.
- Click File on the menu bar and then click Print Now on the File menu (Figure 5–36).

The screenshot shows a CSS file with the following content:

```

body{font-family: Arial, Verdana, Garamond;
  font-size: 11 pt;

p {margin-left: 20pt;
  margin-right: 20pt;
  font-size: 11pt;

a {font-size: 9pt;
  margin-left: auto;
  margin-right: auto;
  text-decoration: none;

img {border: 0px}

```

Figure 5–36

To View the Web Page

- 1 Click the Internet Explorer button on the taskbar.
- 2 Click the Information area on the menu bar image map to display the Web page, as shown in Figure 5–37.

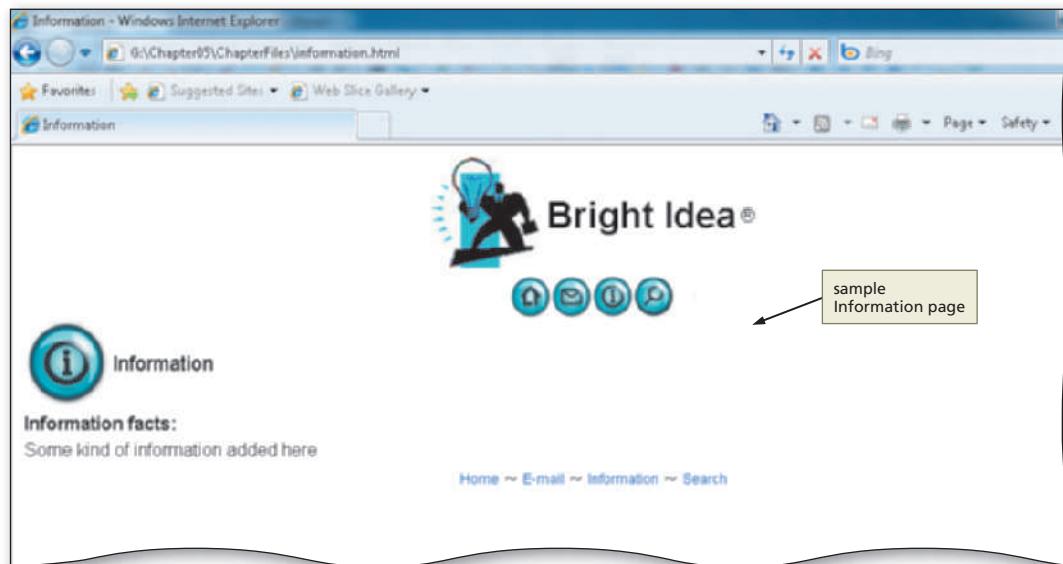


Figure 5–37

To Test and Print the Web Page

- 1 Click the Home link (either text or from the image map) on the Information Web page.
- 2 Click the Print button on the Command bar to print the Web page.
- 3 Click to test the other links, both text and those on the menu bar. If any of the links do not work correctly, return to Notepad++ to modify the HTML code, save the changes, and then retest the links in the browser.

To Quit Notepad++ and a Browser

- 1 In Notepad++, click the File menu, then Close All.
- 2 Click the Close button on the Notepad++ title bar.
- 3 Click the Close button on the browser title bar. If necessary, click the Close all tabs button.

Chapter Summary

BTW**Quick Reference**

For a list of special characters, see the Symbols and Characters Quick Reference (Appendix F) at the back of this book, or visit the Quick Reference Web page for this book (scsite.com/HTML6e/qr).

In this chapter, you have learned how to develop a Web site that utilizes image mapping from the home page to create three clickable areas. The items listed below include all the new HTML skills you have learned in this chapter.

1. Start Paint (HTML 249)
2. Open an Image File in Paint (HTML 251)
3. Locate X- and Y-Coordinates of an Image (HTML 252)
4. Insert a Special Character (HTML 260)
5. Insert an Image to Use as an Image Map (HTML 261)
6. Create an Image Map (HTML 266)
7. Create an External Style Sheet (HTML 271)

Learn It Online

Test your knowledge of chapter content and key terms.

Instructions: To complete the Learn It Online exercises, start your browser, click the Address bar, and then enter the Web address scsite.com/html16e/learn. When the HTML Learn It Online page is displayed, click the link for the exercise you want to complete and read the instructions.

Chapter Reinforcement TF, MC, and SA

A series of true/false, multiple choice, and short answer questions that test your knowledge of the chapter content.

Flash Cards

An interactive learning environment where you identify chapter key terms associated with displayed definitions.

Practice Test

A series of multiple choice questions that test your knowledge of chapter content and key terms.

Who Wants To Be A Computer Genius?

An interactive game that challenges your knowledge of chapter content in the style of a television quiz show.

Wheel of Terms

An interactive game that challenges your knowledge of chapter key terms in the style of the television show, *Wheel of Fortune*.

Crossword Puzzle Challenge

A crossword puzzle that challenges your knowledge of key terms presented in the chapter.

Apply Your Knowledge

Reinforce the skills and apply the concepts you learned in this chapter.

Adding an Image Map to a Web Page

Problem: You decide to use your image mapping skills to create a Web page that describes your company's sales figures for the year. You plan to create a Web page similar to the one shown in Figure 5–38, with the file barchart.png (note the different image type) as an image map that links to four Web pages with information on the various sales for the four quarters of the year.

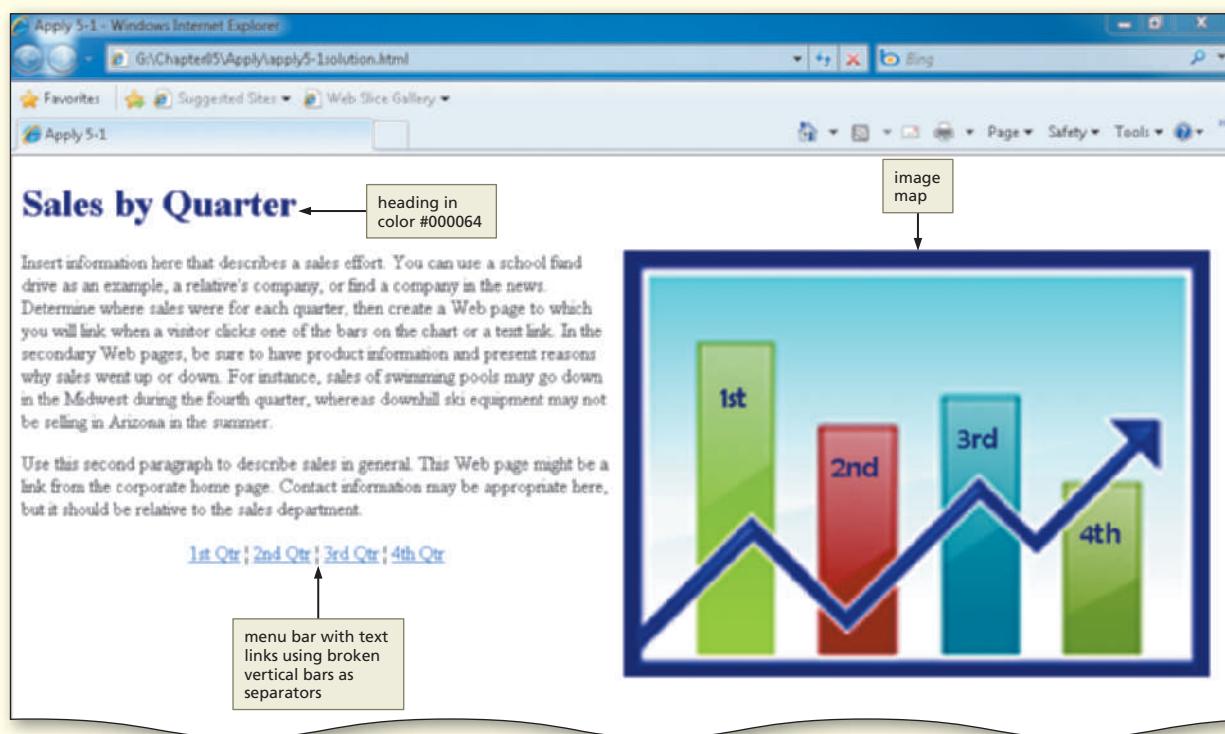


Figure 5–38

Instructions: Start Paint and Notepad++. Perform the following steps:

1. Using Paint, open the file barchart.png from the Chapter05\Apply folder of the Data Files for Students. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files.
2. Each area on the bar chart image is a rectangular area. Use good judgment when planning the shapes of your image map, ensuring that no clickable areas overlap from one rectangular shape into another and that each shape makes sense for its respective area. Using Paint, estimate the x- and y-coordinates necessary to create four clickable areas on the barchart.png image. Write down these coordinates for later use.
3. Using Notepad++, create a new HTML file with the title Apply 5-1 in the title section. Add the heading and text, as shown in Figure 5–38. Use the broken vertical bar (see Appendix F) to separate the text links.
4. Begin the body section by adding a header with #000064 as the color. Use Paint to determine the dimensions of the image for the `` tag. Align the image so that it is to the right of the text.

Continued >

Apply Your Knowledge *continued*

5. Use the `usemap` attribute `usemap="#chart"` in the `` tag.
6. Enter the `<map>` `</map>` tags required to create the image map named chart.
7. Enter the `<area>` tags required to define four clickable areas on the image barchart.png. Use the x- and y-coordinates determined in Step 2 and set the `href` attribute to link to the sample.html file from the Data Files for Students or create your own secondary Web pages.
8. Save the HTML file in the Chapter05\Apply folder using the file name apply5-1solution.html. Validate the Web page(s) using W3C. Print the HTML file.
9. Open the file apply 5-1solution.html in your browser and test the image map and text links to verify they link to the correct Web pages.
10. Print the main Web page and the three linked Web pages.
11. Submit the completed HTML files and Web pages in the format specified by your instructor.

Extend Your Knowledge

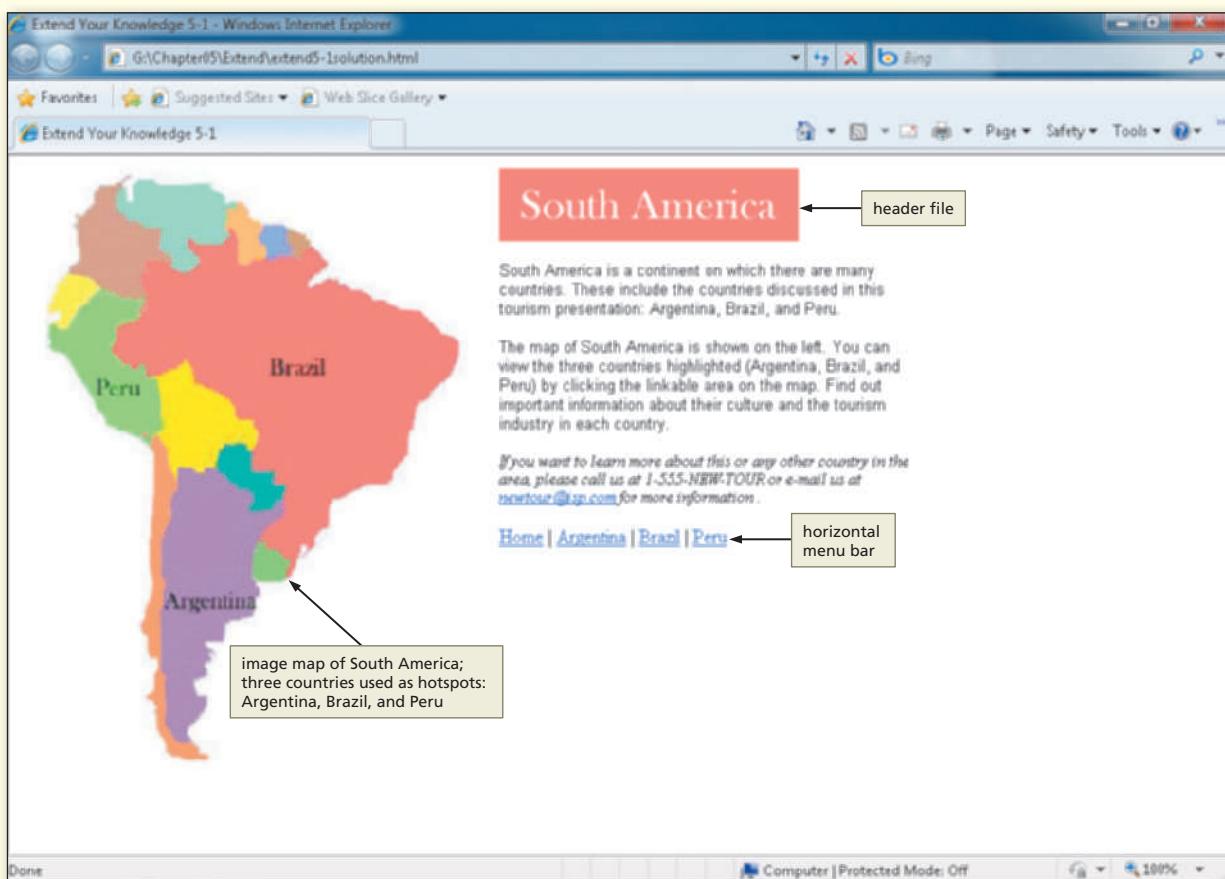
Extend the skills you learned in this chapter and experiment with new skills.

Creating an Image Map

Instructions: Start Notepad++. Open the file extend5-1.html from the Chapter05\Extend folder of the Data Files for Students. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files. The extend5-1.html file is a partially completed HTML file that needs to be completed. Figure 5–39 shows the Extend Your Knowledge Web page as it should appear in your browser after it is completed.

Perform the following tasks:

1. Enter the URL G:\Chapter05\Extend\extend5-1.html to view the Web page in your browser.
2. Examine the HTML file and its appearance as a Web page in the browser.
3. Using Paint, open the file samerica.jpg from the Chapter05\Extend folder. Determine the x- and y-coordinates necessary to create three clickable areas on the map image, one each for Argentina, Brazil, and Peru. You may either use rectangle or polygon shapes for the three areas.
4. Add HTML code to the extend5-1.html file to create an image map that links each clickable area on the map image to an external Web page of your choice.
5. Create a Web page for one of the countries. Research information about the country and include some interesting facts. Include some images on the secondary Web page, but be aware of copyright laws governing their use.
6. Add code to create a horizontal menu bar, as shown in Figure 5–39.

**Figure 5–39**

7. Save the revised file in the Chapter05\Extend folder using the file name extend5-1solution.html.
8. Validate the Web pages to assure that you are in compliance with current standards.
9. Test the links completely.
10. Print the revised HTML file.
11. Enter the URL G:\Chapter05\Extend\extend5-1solution.html to view the Web page in your browser.
12. Print the Web page.
13. Submit the completed HTML file and Web page in the format specified by your instructor.

Make It Right

Analyze a document and correct all errors and/or improve the design.

Correcting the Travel Agency Web Page

Instructions: Start Notepad++. Open the file makeitright5-1.html from the Chapter05\MakeItRight folder of the Data Files for Students. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files. The Web page is a modified version of what you see in Figure 5–40. Make the necessary corrections to the Web page to make it look like the figure. The Web page uses the image getaway.gif. Add four text links at the bottom of the Web page, as shown in Figure 5–40, using Table 5–12 for link name suggestions and URLs. Submit the completed HTML file and Web page in the format specified by your instructor.

Table 5–12 Image Map Coordinates, URLs, and Text Links

Text Link	Image Map Coordinates	URL
Ski & Snow	55,96,253,134	http://www.coloradoski.com/
Surf & Sun	301,93,497,134	http://www.nationalgeographicexpeditions.com/
Golf & Spa	55,161,244,201	http://www.seasidegolf.com/
Adventure	283,158,498,195	http://www.abercrombiekent.com/index.cfm

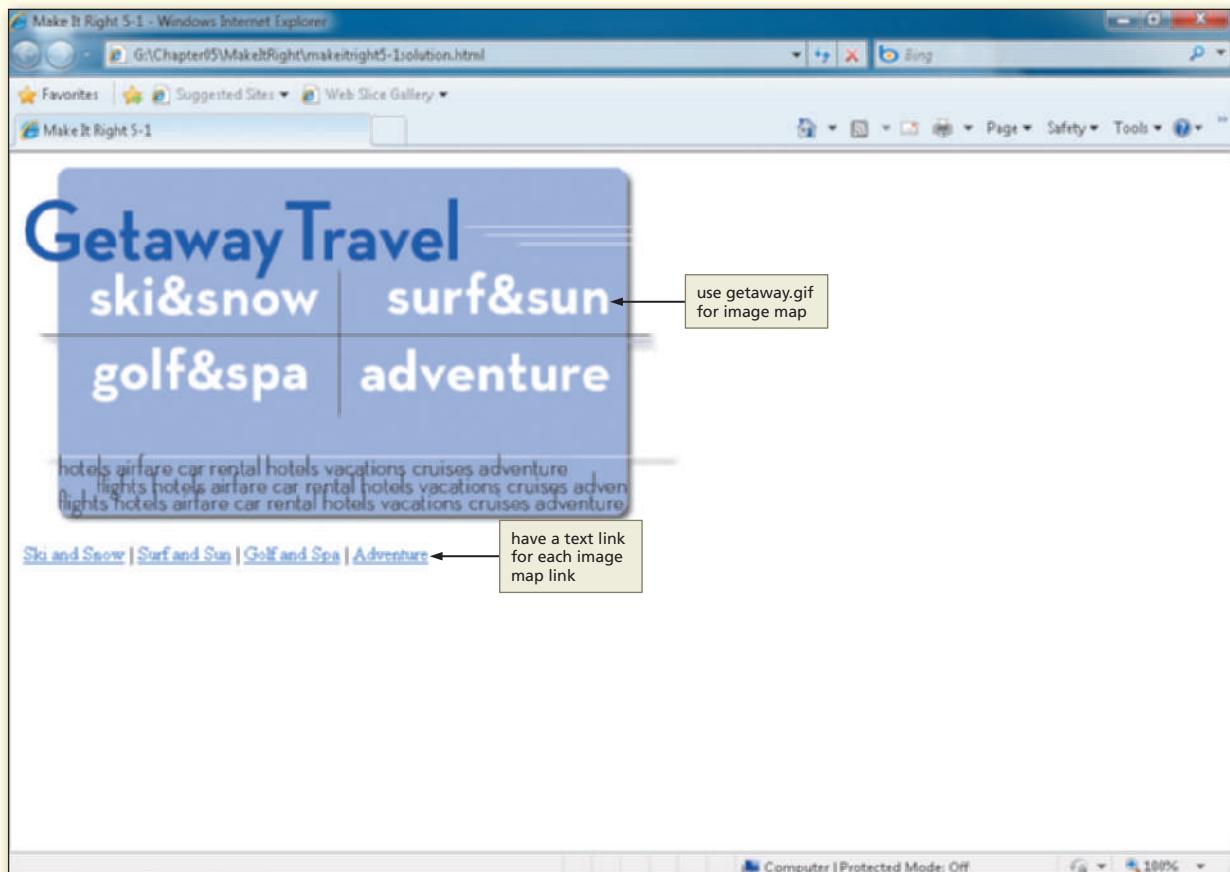


Figure 5–40

In the Lab

Lab 1: Creating an Earth-Friendly Web Page

Problem: You are very involved in making your home and school eco friendly. You decide to create a Web page similar to the Web page in Figure 5–41, with the file earthfriendly.jpg as an image map that links to six Web pages of your choosing.

Instructions: Start Paint and Notepad++. Perform the following steps:

1. Using Paint, open the file earthfriendly.jpg from the Chapter05\IntheLab folder.
2. Determine the x- and y-coordinates necessary to create six rectangular clickable areas on the graphical image, one for each of the six areas, including three recycle bags, two link bars (re-use and say no), and a trash can. Write down these coordinates for later use.
3. Using Notepad++, create a new HTML file with the title Lab 5-1 in the title section. Add the text links, as shown in Figure 5–41.
4. Insert the image earthfriendly.jpg. Use the usemap attribute usemap="#ecomenu" in the tag.
5. Enter the <map> </map> tags required to create the image map named ecomenu.
6. Enter the <area> tags required to define six rectangular clickable areas on the earthfriendly.jpg image. Use the x- and y-coordinates determined in Step 2 and set the href attribute to display the sample.html Web page provided in the Data Files for Students.
7. For a bonus project, link the elements to relevant Web sites of your choosing.

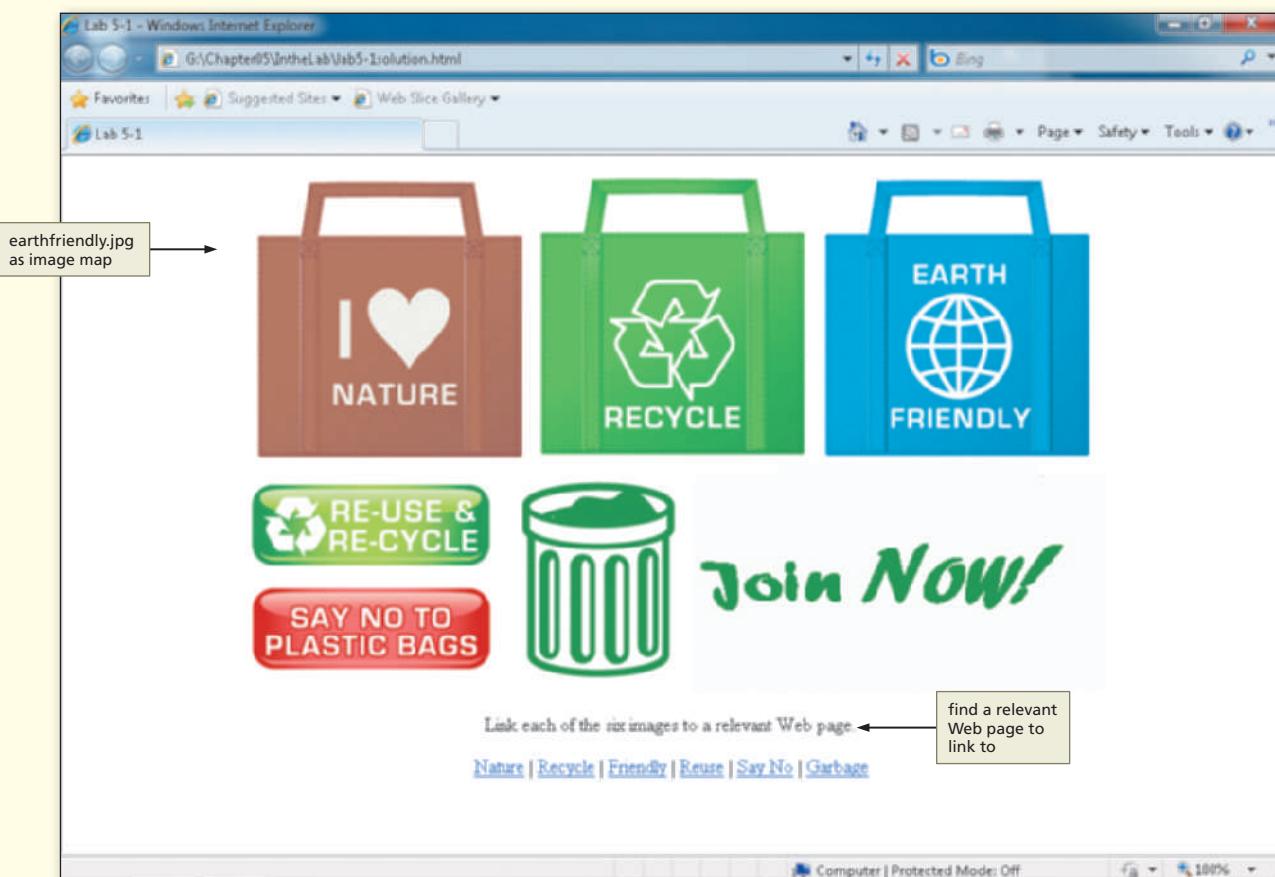


Figure 5–41

Continued >

In the Lab *continued*

8. Save the HTML file in the Chapter05\IntheLab folder using the file name lab5-1solution.html. Validate the Web page using W3C. Print the HTML file.
9. Open the file lab5-1solution.html in your browser and test the image map and text links to verify they link to the correct Web pages.
10. Print the main Web page and the three linked Web pages.
11. Submit the completed HTML files and Web pages in the format specified by your instructor.

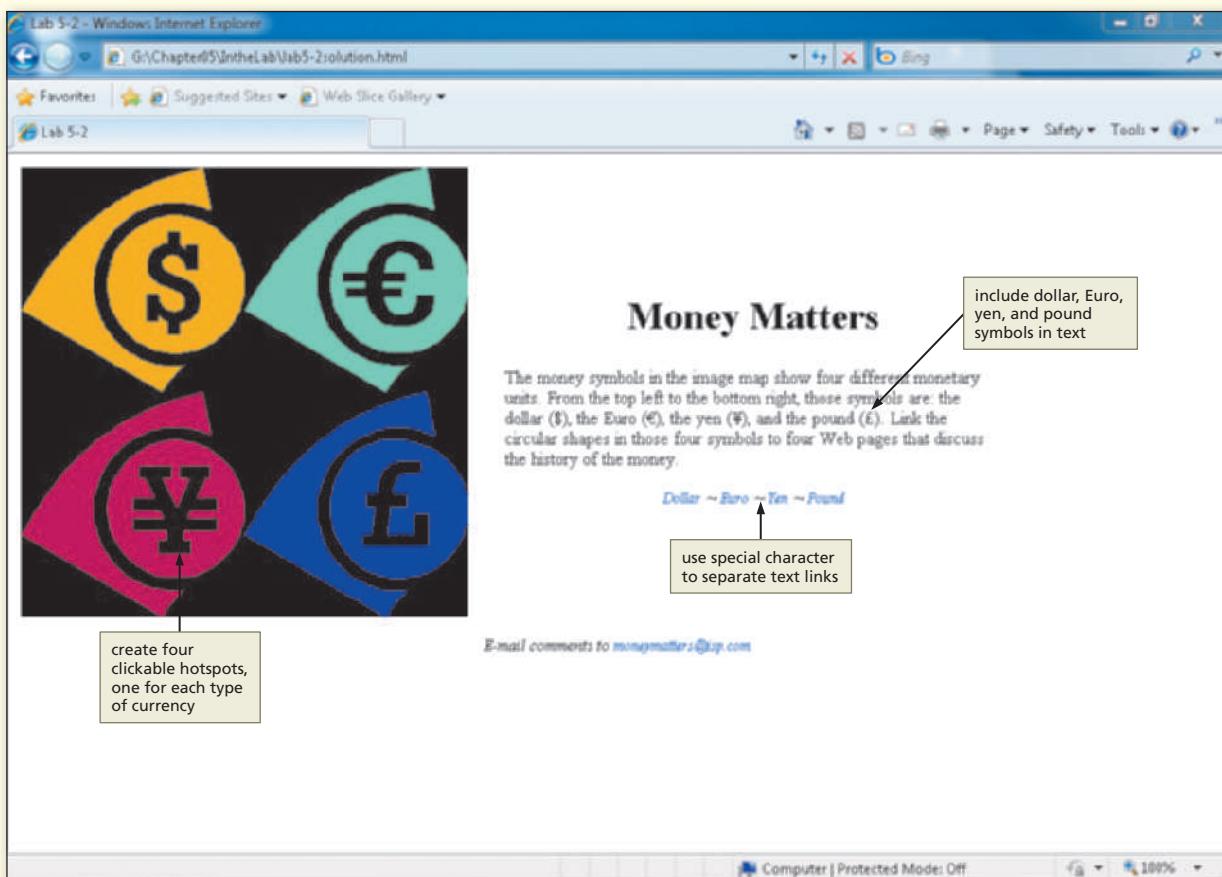
In the Lab

Lab 2: Mapping Currency

Problem: You are learning about different denominations of money in your economics class. You decide to create a Web site that displays information about four different currencies (dollar, Euro, yen, and pound). Use your image mapping skills to create a Web page that has four circular links. You plan to create a Web page similar to the one shown in Figure 5–42, with the file moneymap.jpg as an image map that links to four Web pages (dollar.html, euro.html, yen.html, and pound.html) that contain one statement each of information about the selected currency.

Instructions: Start Paint and Notepad++. Perform the following steps:

1. Using Paint, open the file moneymap.jpg from the Chapter05\IntheLab folder.
2. Each area on the image is a circular shape. Use good judgment when planning the shapes of your image map, ensuring that no clickable areas overlap from one circular shape into another and that each shape makes sense for its respective area. Using Paint, estimate the x- and y-coordinates necessary to create four circular clickable areas on the moneymap.jpg image.
3. Using Notepad++, create a new HTML file with the title Lab 5-2 in the title section. Add the heading and text, as shown in Figure 5–42.
4. Begin the body section by adding a one-row, two-data-cell table. Center the text in the right-hand data cell. Make sure to include the special characters (see Appendix F) for the Euro, yen, and pound in the paragraph of text, as shown in Figure 5–42. (Why is there no special character needed for the dollar sign?) The e-mail text and link should be a paragraph below the table.
5. Use the usemap attribute usemap="#moneymap" in the tag.
6. Enter the <map> </map> tags required to create the image map named moneymap.
7. Enter the <area> tags required to define four circular clickable areas on the image moneymap.jpg. Use the x- and y-coordinates determined in Step 2 and set the href attribute to link to the four Web pages provided as files in the Data Files for Students or create your own secondary Web page. For bonus points, insert relevant information in the four linked-to Web pages.
8. Save the HTML file in the Chapter05\IntheLab folder using the file name lab5-2solution.html. Validate the Web page(s) using W3C. Print the HTML file.
9. Open the file lab5-2solution.html in your browser and test the image map and text links to verify they link to the correct Web pages.
10. Print the main Web page and the four linked Web pages.
11. Submit the completed HTML files and Web pages in the format specified by your instructor.

**Figure 5–42**

In the Lab

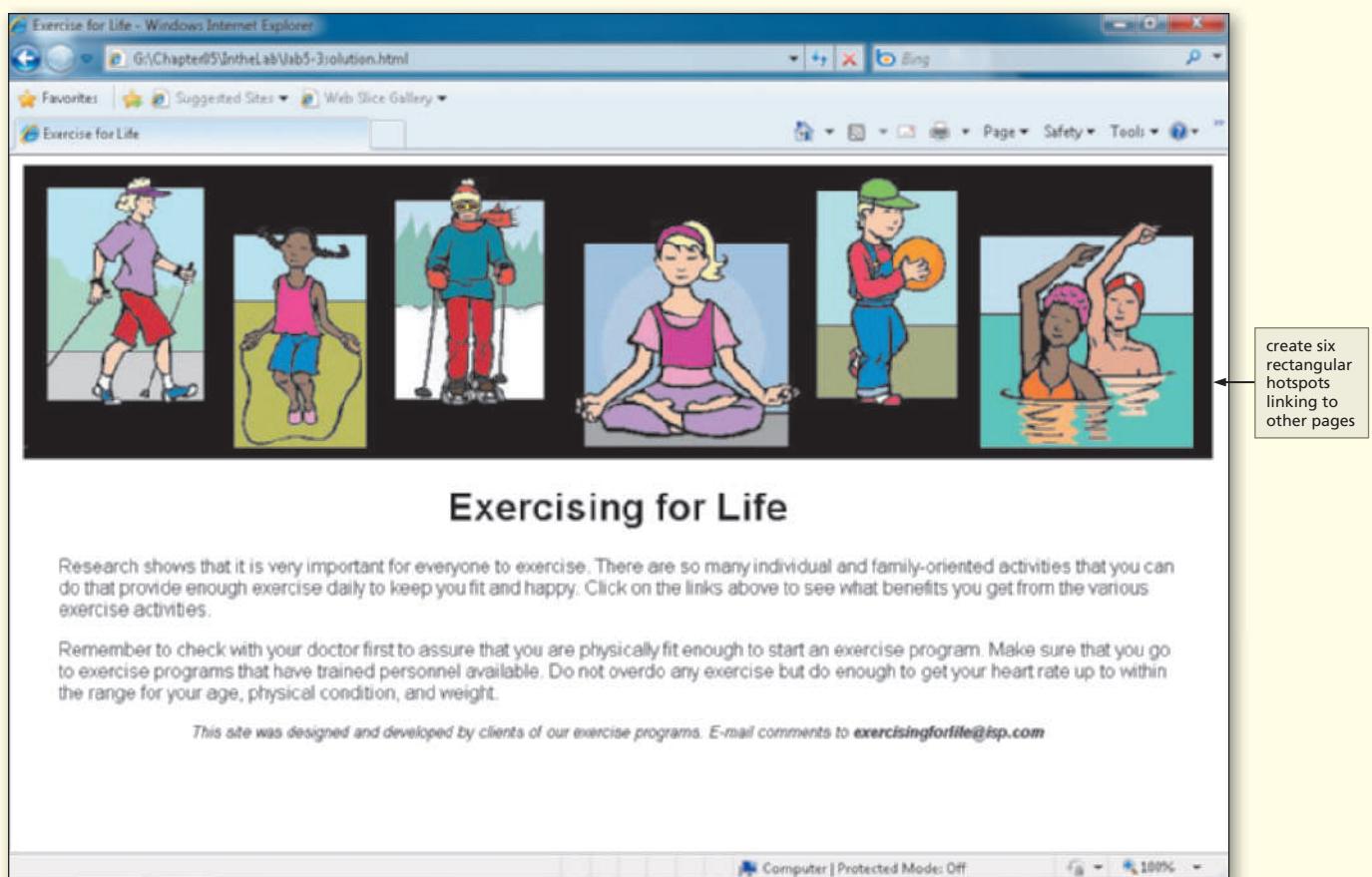
Lab 3: Creating an Exercise Web Page

Problem: You are trying to lead a healthier lifestyle, and have researched how many calories are burned for six different activities, based on a person's weight. To help others who are also interested in the benefits of exercise, you create a Web site with a home page similar to Figure 5–43 on the next page.

Instructions: Start Paint and Notepad++. Perform the following tasks:

1. Using Paint, open the file exercisemap.jpg from the Chapter05\IntheLab folder of the Data Files for Students.
2. Using Paint, determine the x- and y-coordinates necessary to create six clickable areas on the map image, using rectangular-shaped areas. Write down these coordinates for later use.
3. Using Notepad++, create a new HTML file with the title Lab 5-3 in the main heading section.
4. Begin the body section by adding the exercisemap.jpg image, as shown in Figure 5–43.
5. Use the usemap attribute usemap="#menubar" in the tag.
6. Enter the <map> </map> tags required to create the image map named menubar.

Continued >

In the Lab continued**Figure 5–43**

7. Enter the `<area>` tags required to define six clickable areas on the image `exercisemap.jpg`. Use the x- and y-coordinates determined in Step 2 and set the `href` attribute to use the URLs for the six exercise activity Web pages you have identified for the six different activities. Link the clickable areas to Web pages named: `hiking.html`, `jumping.html`, `playingball.html`, `skiing.html`, `water.html`, and `yoga.html`.
8. Save the HTML file in the `Chapter05\IntheLab` folder using the file name `lab5-3solution.html`. Validate the Web page. Print the HTML file.
9. Create the six Web pages to correspond with the links, naming them as shown above. Use the Word document `lab5-3ActivityChart.doc` from the Data Files for Students to find the calories burned by weight for each activity.
10. Open the file `lab5-3solution.html` in your browser and test the image map and text links to verify they link to the correct Web pages.
11. Print the main Web page and the six linked Web pages.
12. Submit the completed HTML files and Web pages in the format specified by your instructor.

Cases and Places

Apply your creative thinking and problem-solving skills to design and implement a solution.

- EASIER
- MORE DIFFICULT

● 1: Completing the Bright Ideas Web Site

The Board of Directors for the Bright Idea company likes the home page of the new Web site. They want you to create more final versions of the Information and Search pages. For the Information page, you should search for similar companies and add samples of what you found onto the Information page. Save and validate the Web page with the additional information. Submit it in the format specified by your instructor. For the Search page, you should review how to best handle a search function on a Web page. What resources are out there for companies to use for searches of their Web sites? (*Hint:* Try “powered by Google.”) Note any information that you find in a paper and discuss what needs to be done to have an effective search function. Submit in the format specified by your instructor.

● 2: Promoting Library Services

The marketing director of your community library wants to create a new graphical home page that highlights the different services that the library provides, including: book reserve, video rentals, available meeting rooms, and computers with Internet access. Using the books.jpg image from the Chapter05\CasesandPlaces folder of the Data Files for Students, create a Web page. First, open the books.jpg image in Paint to determine hotspots that you can use as three or four links. For a higher level of difficulty, use the Text tool in Paint to add relevant words onto the books for those links. (*Hint:* Search the Paint Help utility for the word Text.) From those hotspots, create links to subsequent Web pages that describe the services. Be sure to include text links at the bottom of the page to mirror the links in the image map.

●● 3: Browsing with Images Turned Off

As discussed in this chapter, some Web site visitors turn graphics off while browsing. Find out how extensive this practice might be by researching statistics online that show the number of Web users who turn graphics off when they browse. With this information in hand, search the Web to find three Web sites that utilize image maps. Track the time that it takes to load the three Web pages with image maps. Turn off graphics in your browser (in Internet Explorer, click Tools on the Command bar and click Internet Options, and then click the Advanced tab; scroll down and click Show pictures under Multimedia to deselect it). Next, clear the browser’s history (in Internet Explorer, click Tools on the Command bar and Internet Options, and then click the Delete button under Browsing history). Reload each of the three Web pages and again track the time it takes for the pages to load, this time without images. Determine if the Web pages load more quickly with images off. Review each Web page and determine if you can use all of the links despite having graphics turned off.

●● 4: Researching Image Mapping Software

Make It Personal

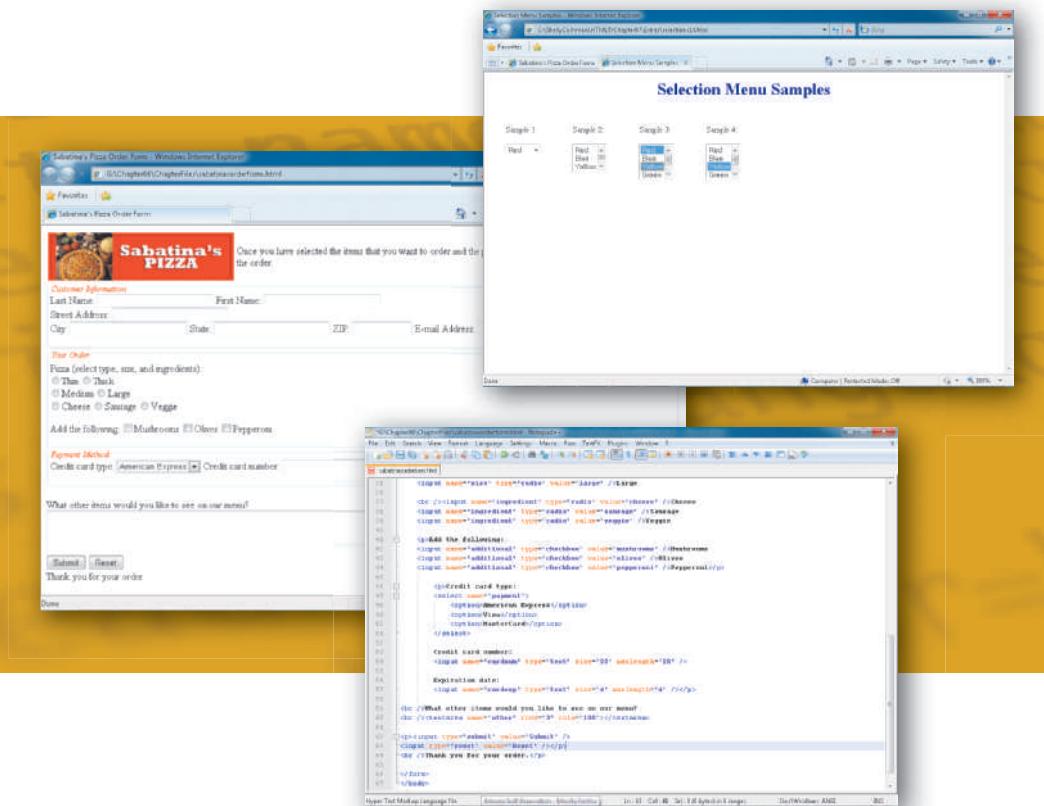
You are very interested in finding out more information about some of the image mapping software tools listed in Table 5–2 on page HTML 254. Review the information about each tool listed, including any associated costs, free trial version availability, platform(s) supported, and ease of use. If a free trial version is offered at any of the Web sites and you are using your own computer (or your instructor or lab coordinator allows it), download the software and use it to create an image map. Compare the technique of using these tools to the technique used in this chapter using Paint. Write a synopsis of the available products, including any associated costs, free trial version availability, platform(s) supported, and ease of use.

Continued >

Cases and Places *continued***•• 5: Creating a Team Image Map****Working Together**

Each member of your team should find an image that depicts the members of the team and can be used as an image map. The image could have areas that represent the different areas in which you are majoring, the sports or clubs of which you are members, or even personal interests. If you cannot find one image that suffices, you can use Paint (or another graphic editing tool) to combine relevant images that you found, making your own image map. Once you have an image map, sketch out the areas that you would use as hotspots. Use that image map to link these hotspots to your own personal Web pages.

6 Creating a Form on a Web Page



Objectives

You will have mastered the material in this chapter when you can:

- Define terms related to forms
- Describe the different form controls and their uses
- Use the `<form>` `</form>` tags
- Use the `<input>` tag
- Create a text box
- Create check boxes
- Create a selection menu with multiple options
- Use the `<select>` tag
- Use the `<option>` tag
- Create radio buttons
- Create a textarea box
- Create a Submit button
- Create a Reset button
- Use the `<fieldset>` and `<legend>` tags to group form information

6 | Creating a Form on a Web Page

Introduction

The goal of the projects completed thus far has been to present information *to* Web site visitors. In this chapter, you learn how to get information *from* Web site visitors by adding a form for user input.

Using a Web page form for user input reduces the potential for errors because customers enter data or select options from the form included directly on the Web page. A form has input fields to remind users to enter information and limits choices to valid options to avoid incorrect data entry. Forms provide an easy way to collect needed information from Web visitors.

In this chapter, you will learn how to use HTML to create a form on a Web page. The form will include several controls, including check boxes, a drop-down list, radio buttons, and text boxes. You also will learn to add Submit and Reset buttons that customers can use to submit the completed form or clear the information previously entered into the form. Finally, you will learn to use the `<fieldset>` tag to group information on a form in a user-friendly way with the `<legend>` tag, which is used to add labels to the groups within the form.

Project — Creating Forms on a Web Page

The Sabatina's Pizza Web site (Chapter 4) has been a great success. Many customers have visited the Web site to review the menu choices. Sabatina's owners decided to add an online order form, but it has been cumbersome for the customers. They have to follow the instructions on the Order Form Web page, print out the form, fill in their menu item choices, and fax orders in. Although the orders are usually complete, sometimes they are missing key information, or the customers request options that are not available. Sabatina's owners ask you if an easier, less error-prone way exists for customers to order menu items online.

The Sabatina's Pizza Order Form Web page is provided in the Data Files for Students for this chapter. In this project, you enter HTML tags to modify the text-based Order Form Web page that the owner of Sabatina's Pizza originally created (Figure 6–1a) and from it, create a Web-based Order Form, as shown in Figure 6–1b. This page requests the same information as the text-based Web page, but is created with a Web-based form that allows users to enter data, select options, and then submit the form to the e-mail address indicated on the form.

Sabatina's Pizza Order Form - Windows Internet Explorer

G:\Chapter06\ChapterFiles\sabatinapizzaorderform.html

Sabatina's Pizza Order Form

Customer Information

Last Name, First Name: _____

Street address: _____

City, State, ZIP, E-mail Address: _____

Your Order

Appetizers:	Salad:	Pizza (select type, size, and ingredients):
Breadsticks	Caesar	Thin Thick
Calamari	Mediterranean	Medium Large
Mozzarella Sticks	Greek	Cheese Sausage Veggie
Stuffed Mushrooms	Japanese	

Payment Method

Credit card type: American Express, Visa, MasterCard

Credit card number: _____ Expiration date: _____

What other items would you like to see on our menu? _____

Thank you for your order.

(a) **Sabatina's text-based order form.**

Sabatina's Pizza Order Form - Windows Internet Explorer

G:\Chapter06\ChapterFiles\sabatinapizzaorderform.html

Sabatina's Pizza Order Form

Once you have selected the items that you want to order and the payment type, click the Submit button to process the order.

Customer Information

Last Name: _____ First Name: _____

Street Address: _____

City: _____ State: _____ ZIP: _____ E-mail Address: _____

Your Order

Pizza (select type, size, and ingredients):

Thin Thick
 Medium Large
 Cheese Sausage Veggie

Add the following: Mushrooms Olives Pepperoni

Payment Method

Credit card type: American Express Credit card number: _____ Expiration date: _____

What other items would you like to see on our menu?

Submit **Reset** **Submit and Reset buttons**

Thank you for your order.

(b) **Sabatina's Web-based order form.**

Figure 6–1

Overview

As you read this chapter, you will learn how to create the Web page shown in Figure 6–1b on the previous page by performing these general tasks:

- Enter HTML code into the Notepad++ window.
- Save the file as an HTML file.
- Enter basic HTML tags and add text to the file.
- Insert tags to create a form with several input controls.
- Create Submit and Reset buttons on the form.
- Add interest and organization to the form using `<fieldset>` and `<legend>` tags.
- Add an embedded style sheet to format specific elements on a Web page.
- Save and print the HTML code.
- Validate, view, and print the Web pages.

Plan Ahead

General Project Guidelines

As you create Web pages, such as the project shown in Figure 6–1 on the previous page, you should follow these general guidelines:

1. **Plan the Web site.** You should plan the information that you hope to collect before you begin to write your HTML code. Refer to Table 1–4 on page HTML 15 for information on the planning phase of the Web Development Life Cycle. In this phase, you determine the purpose of the Web form, identify the users of the form and their computing environment, and decide how best to capture the information sought using a Web page.
2. **Analyze the need.** In the analysis phase of the Web Development Life Cycle, you should analyze what content to include in the Web page form. The Web development project in this chapter is different than the ones completed in other chapters because it contains a form. Part of the analysis phase then includes determining what information to collect and the best form input controls to use for this collection.
3. **Determine the types of controls to use.** The type of information a form is intended to gather dictates what controls are used in the form. For instance, in the case in which only one option from a list can be selected, you should use the radio button control. In the case in which more than one option can be selected, you can use check boxes or selection controls. If you want users to be able to add their own comments, you can use a textarea box. Most forms use a combination of controls, not just a single type.
4. **Establish what other form options are necessary.** Form organization is an important aspect of Web page form development. You want to be sure that the user understands what information to provide. You also want the form to be attractive and easy to use. Consider using `fieldset` and `legend` tags to divide the form attractively and segregate information into logical subsets.
5. **Create the Web page form and links.** Once the analysis and design is complete, the Web developer creates the Web page form using HTML. Good Web development standard practices should be followed in this step. Examples of good practices include utilizing the form controls that are appropriate for specific needs.
6. **Test the Web page form.** An important part of Web development is testing to assure that you are following XHTML standards. In this book, we use the World Wide Web Consortium (W3C) validator that allows you to test your Web page and clearly explains any errors you have. Additionally when testing, you should verify that all controls work as intended. Finally, both the Submit and the Reset buttons should be tested.

When necessary, more specific details concerning the above guidelines are presented at appropriate points in the chapter. The chapter also will identify the actions performed and decisions made regarding these guidelines during the creation of the Web page shown in Figure 6–1.

Web Page Forms

The Sabatina's Order Form Web page shown in Figure 6–1b on page HTML 287 shows an example of a Web page form designed to request specific information from the Web page visitor. A Web page form has three main components:

- Input controls
- A <form> tag, which contains the information necessary to process the form
- A Submit button, which sends the data to be processed

Input Controls

An **input control** is any type of input mechanism on a form. A form may contain several different input controls classified as data or text input controls. A **data input control** can be a radio button (radio), a check box (checkbox), a Submit button (submit), a Reset button (reset), or a selection menu (select). A **text input control** allows the user to enter text through the following:

- A **text box** (text), for small amounts of text
- A **textarea box** (textarea), for larger amounts of text
- A **password text box** (password), for entering a password

As shown in Figure 6–1b, the form developed in this chapter uses several different data and text input controls.

Of the available input controls, the eight listed in Table 6–1 are used most often in form creation.

Table 6–1 Form Input Controls

Control	Function	Remarks
text	• Creates a single-line field for a relatively small amount of text	• Indicates both the size of the field and the total maximum length
password	• Identical to text boxes used for single-line data entry	• Echoes (or masks) the entered text as bullets
textarea	• Creates a multiple-line field for a relatively large amount of text	• Indicates the number of rows and columns for the area
select	• Creates a drop-down list or menu of choices from which a visitor can select an option or options	• Indicates the length of the list in number of rows
checkbox	• Creates a single item or a list of items	• Indicates a single item that can be checked • Indicates a list of more than one item that can be chosen
radio	• Creates a list item	• Indicates only one item in a list can be chosen
submit	• Submits a form for processing	• Tells the browser to send the data on the form to the server
reset	• Resets the form	• Returns all input controls to the default status

A **text control** creates a text box that is used for a single line of input (Figure 6–2 on the next page). The text control has two attributes:

- **size**, which determines the number of characters that are displayed on the form
- **maxlength**, which specifies the maximum length of the input field

BTW

Forms

Several HTML guides on the Internet discuss the use of forms on Web pages. Many of these sites are created and maintained at universities. The guides give practical tips on the purpose and use of HTML tags and attributes. To view an HTML guide, use a search engine to search for the phrase "HTML Guide" or a related phrase.

The maximum length of the field may exceed the size of the field that appears on the form. For example, consider a field size of three characters and a maximum length of nine characters. If a Web page visitor types in more characters than the size of the text box (three characters), the characters scroll to the left, to a maximum of nine characters entered. For example, this code creates a text box to input the user's last name:

```
<p>Last Name: <input name="lastname" type="text" size="25" />
```

A **password control** also creates a text box used for a single line of input (Figure 6–2), except that the characters entered into the field can appear as asterisks or bullets. A password text box holds the password entered by a visitor. The password appears as a series of characters, asterisks, or bullets as determined by the Web developer, one per character for the password entered. This feature is designed to help protect the visitor's password from being observed by others as it is being entered.



Figure 6–2 Text and password text controls.

BTW

Radio Buttons

Old-time car radios were operated by a row of large black plastic buttons. Push one button, and you would get one preset radio station. You could push only one button at a time. Radio buttons on forms work the same way as the old-time radio buttons—one button at a time. With check boxes, more than one option can be selected at a time.

A **radio control** limits the Web page visitor to only one choice from a list of choices (Figure 6–3). Each choice is preceded by a **radio button**, or option button, which typically appears as an open circle. When the visitor selects one of the radio buttons, all other radio buttons in the list automatically are deselected. By default, all radio buttons are deselected. To set a particular button as the default, you use the checked value within the `<input />` tag. Here is example code to create two radio controls:

```
<input name="ingredient" type="radio" value="sausage" />Sausage
<input name="ingredient" type="radio" value="veggie" />Veggie
```

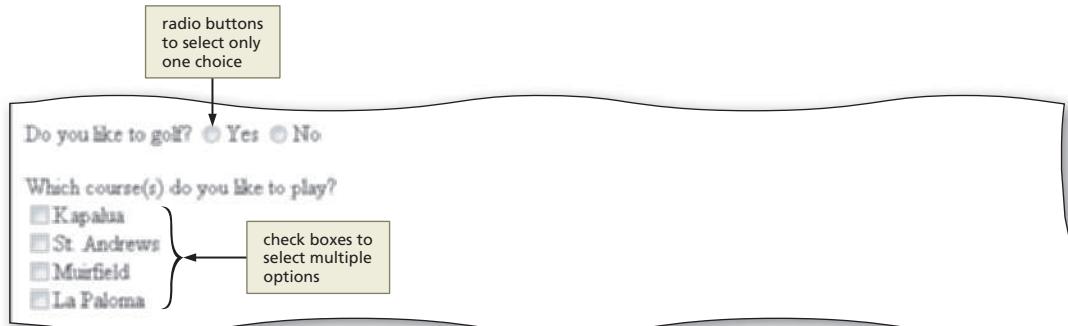


Figure 6–3 Radio and checkbox controls.

A **checkbox control** allows a Web page visitor to (a) select one item from a single-item list or (b) select more than one choice from a list of choices (Figure 6–3). Each choice in a check box list can be either on or off. By default, all check boxes are deselected.

The default can be changed so a particular check box is preselected as the default, by using the checked value within the `<input />` tag. Here is sample code for a checkbox control:

```
<input name="additional" type="checkbox" value="mushrooms" />
Mushrooms
```

A **select control** creates a selection menu from which the visitor selects one or more choices (Figure 6–4). This prevents the visitor from having to type information into a text or textarea field. A select control is suitable when a limited number of choices are available. The user clicks the list arrow to view all the choices in the menu. When clicked, the default appears first and is highlighted to indicate that it is selected. Here is sample code for a select control:

```
<select name="payment">
<option>American Express</option>
<option>Visa</option>
<option>MasterCard</option>
</select>
```

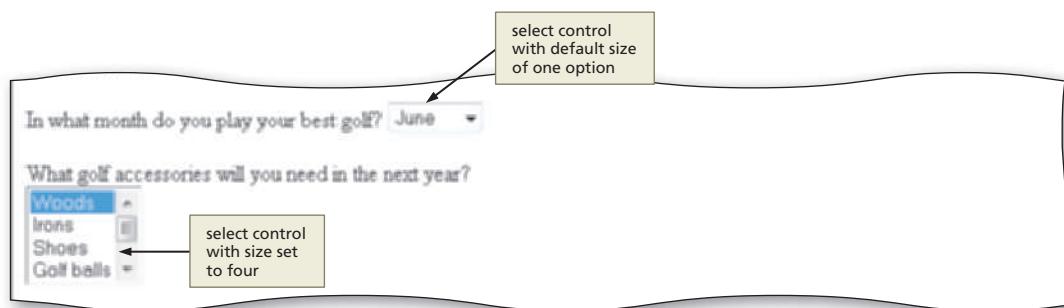


Figure 6–4 Different options for select controls.

A **textarea control** creates a field that allows multiple lines of input (Figure 6–5). Textarea fields are useful when an extensive amount of input is required from, or desired of, a Web page visitor. The textarea control has two primary attributes:

- **rows**, which specifies the number of rows in the textarea field
- **cols**, which specifies the number of columns in the textarea field

A textarea control is created with the following code:

```
What other items would you like to see on our menu?
<textarea name="other" rows="3" cols="100"></textarea>
```

The **fieldset control** (Figure 6–5 on the next page) helps to group related form elements together. This makes the form easier to read and complete. The form segment in Figure 6–5 shows two groupings: one with a left-aligned legend and the other with a right-aligned legend. Using fieldset tags to segregate information allows the Web page visitor immediately to see that two (or more) categories of information are included in the form. The easier that it is for a user to complete a form, the more likely it is that he or she will complete it. A fieldset control is created with the following code:

```
<fieldset><legend>Payment Method</legend></fieldset>
```

BTW

Textareas

To create a textarea, the Web developer specifies the number of rows and columns in which the Web page visitor can enter information. The maximum number of characters for a textarea is 32,700. It is a good rule to keep the number of columns in a textarea to 50 or fewer. Using that as a limit, the textarea will fit on most screens.

The diagram illustrates a web form structure. It features a large rectangular container with rounded corners. Inside, there are two sections: 'Login Information' on the left and 'Golf Playing Habits' on the right. The 'Login Information' section contains two input fields: 'Please give us your name:' and 'And your password:'. The 'Golf Playing Habits' section contains a legend labeled 'right-aligned legend' and a large text area labeled 'textarea with multiple rows and columns'. The entire form is enclosed in a 'first grouping' element, which is itself contained within a larger 'second grouping' element.

Figure 6-5 Fieldset and textarea controls.

The **submit control** and the **reset control** create the Submit and Reset buttons, respectively (Figure 6-6). The **Submit button** sends the information to the appropriate location for processing. The **Reset button** clears any input that was entered in the form, resetting the input controls back to the defaults. A Web page form must include a Submit button, and most also include a Reset button. The submit and reset controls are created with the following code:

```
<input type="submit" value="Submit" />
<input type="reset" value="Reset" />
```

The diagram shows a simplified form structure. It consists of a large rectangular container with rounded corners. Inside, there are two buttons labeled 'Submit' and 'Reset'. Below the buttons, a message reads 'Thank you for your input.' The 'Submit' and 'Reset' buttons are grouped together and labeled 'Submit and Reset buttons'.

Figure 6-6 Submit and Reset button controls.

Regardless of the specific type, each input control has one or two attributes:

- **name**, which identifies the specific information that is being sent when the form is submitted for processing. All controls have a name.
- **value**, which is the type of data that is contained in the named input control (that is, the data that the Web page visitor enters). All controls except textarea also have a value attribute. For a textarea field, no value attribute is possible because of the variability of the input.

When a Web page visitor clicks the Submit button on the form, both the control name and the value of the data contained within that control are sent to the server to be processed.

HTML Tags Used to Create Forms

Form statements start with the `<form>` tag and end with the `</form>` tag. The input controls in a form are created using either HTML tags or attributes of HTML tags. For example, the select and textarea controls are created using the HTML tags `<select>` and `<textarea>`, respectively. Other input controls are created using attributes of HTML tags. For example, the text boxes, check boxes, radio buttons, and Submit and Reset buttons all are created using the `type` attribute of the `<input />` tag. Table 6–2 lists the HTML tags used to create the order form in this chapter. Any combination of these elements can be used in a Web page form.

Table 6–2 HTML Tags Used to Create Forms

Tag	Function	Remarks
<code><fieldset></code> <code></fieldset></code>	Groups related controls on a form	Optionally used for readability
<code><form></code> <code></form></code>	Creates a form that allows user input	Required when creating forms
<code><input /></code>	Defines the controls used in the form, using a variety of <code>type</code> attribute values	Required for input controls
<code><legend></code> <code></legend></code>	Defines the text that is displayed in the grouping borders	Optionally used when using <code><fieldset></code> tags
<code><select></code> <code></select></code>	Creates a menu of choices from which a visitor selects	Required for selection choices
<code><option></code> <code></option></code>	Specifies a choice in a <code><select></code> tag	Required, one per choice
<code><textarea></code> <code></textarea></code>	Creates a multiple-line text input area	Required for longer text inputs that appear on several lines

Attributes of HTML Tags Used to Create Forms

Many of the HTML tags used to create forms have several attributes. Table 6–3 lists some of the HTML tags used to create forms, along with their main attributes and functions.

Table 6–3 Attributes and Functions of HTML Tags Used to Create Forms

Tag	Attribute	Function
<code><form></code> <code></form></code>	action method target	<ul style="list-style-type: none"> URL for action completed by the server HTTP method (post) Location at which the resource will be displayed
<code><input /></code>	type name value checked disabled readonly size maxlength src alt tabindex	<ul style="list-style-type: none"> Type of input control (text, password, checkbox, radio, submit, reset, file, hidden, image, button) Name of the control Value submitted if a control is selected (required for radio and checkbox controls) Sets a radio button to a checked state (only one can be checked) Disables a control Used for text passwords Number of characters that appear on the form Maximum number of characters that can be entered URL to the location of an image stored on the server Alternative text for an image control Sets tabbing order among control elements

BTW

Form Tutorial

What better way to learn more about the HTML form tag than using a tutorial on the Web? Many Web sites have lessons grouped by topic, starting with initial HTML tags. An index is generally provided for ease of use. Most tutorials use illustrative examples to teach the important points of HTML to create Web pages. To find HTML tutorials, search the Web using a popular search engine.

Table 6–3 Attributes and Functions of HTML Tags Used to Create Forms (continued)

Tag	Attribute	Function
<legend> </legend>	align	• Indicates how a legend should be aligned
<select> </select>	name size multiple disabled tabindex	• Name of the element • Number of options visible when Web page is first opened • Allows for multiple selections in select list • Disables a control • Sets the tabbing order among control elements
<option> </option>	selected disabled value	• Specifies whether an option is selected • Disables a control • Value submitted if a control is selected
<textarea> </textarea>	name rows cols disabled readonly tabindex	• Name of the control • Height in number of rows • Width in number of columns • Disables a control • Used for text passwords • Sets the tabbing order among control elements

Creating a Form on a Web Page

In this chapter, you will modify the text-based Order Form Web page used in the Sabatina's Pizza Web site. The file, *sabatinasorderform.html*, currently contains only text and does not utilize a form or form controls (Figure 6–1a on page HTML 287). Using this text-based order form is inconvenient for the user, who must print the form, complete the required order information, and then fax that information to the phone number listed in the opening paragraph of text.

The file, *sabatinasorderform.html*, is stored in the Data Files for Students for this chapter. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required file. After opening this file in Notepad++, you will enter HTML code to convert this text-based Web page into the Web page form shown in Figure 6–1b on page HTML 287.

Plan Ahead

Processing form information.

One of the most important issues to determine when creating a Web page form is what to do with the information once it is entered. One way to process the information is to use a CGI script, which is code that has been previously written in a language other than HTML. The information collected from forms is often used to feed databases. A CGI script provides a much better way to process that information. For the scope of this book, we use the second method to process information, the post. The post can be used to send the information to an e-mail address.

- **Using a CGI script.** This action is beyond the scope of this book, but it is the more efficient way to handle the information input into the Web page form. A Web developer would have to find out what script capabilities reside on the server in order to utilize it.
- **Posting to an e-mail address.** Because we do not know what CGI scripts are available on the Web servers at your location, we will utilize the e-mail posting technique in this chapter. The information posted to an e-mail address is not readily usable, so other steps will have to be taken to utilize the data coming in via e-mail.

To Start Notepad++ and Open an HTML File

The following steps illustrate how to start Notepad++ and open the HTML file, sabatinasorderform.html.

- 1** Start Notepad++ and, if necessary, maximize the window.
- 2** With a USB drive plugged into your computer, click File on the menu bar and then click Open.
- 3** If necessary, navigate to the Chapter06\ChapterFiles folder on the USB drive.
- 4** Click sabatinasorderform.html in the list of files.
- 5** Click the Open button to open the sabatinasorderform.html file in Notepad++ (Figure 6–7).



```

1 <!DOCTYPE html
2   PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
6 <head>
7   <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
8   <title>Sabatina's Pizza Order Form</title>
9   <!-- Insert embedded style sheet here -->
10 </head>
11 <body>
12
13   <!-- Insert form method statement here -->
14
15   <table>
16     <tr>
17       <td></td>
18       <td>Print this order form, complete the necessary information, circle the items that you want to order and the payment type, and fax the order to us at 1-800-sabatina.</td>
19     </tr>
20   </table>
21
22   Customer Information
23   <p>Last Name, First Name: _____</p>
24   <p>Street address: _____</p>
25   <p>City, State, ZIP, E-mail Address: _____</p>
26
27   Your Order
28   <table>
29     <tr>
30       <td class="colspec"><br />Appetizers:<br />Breadsticks<br />Cannoli<br />Mozzarella Sticks<br />Stuffed Mushrooms</td>
31       <td class="colspec"><br />Salads:<br />Caesar<br />Mediterranean<br />Greek<br />Japanese</td>
32       <td class="colspec"><br />Pizza (select type, size, and ingredient):<br />Thin<br />Medium<br />Large<br />Cheese<br />Sausage<br />Veggie</td>
33     </tr>
34   </table>
35
36   <p>Payment Method<br />Credit card type: American Express, Visa, MasterCard<br />Credit card number: _____ Expiration date: _____<br />What other items would you like to see on our menu? _____<br />Insert Submit and Reset buttons here --><br />Thank you for your order.</p>
37
38 </body>
39 </html>

```

sabatinasorderform.html
file from Data Files for
Students

Figure 6–7

Creating a Form and Identifying the Form Process

When adding a form to a Web page, the first steps are creating the form and identifying how the form is processed when it is submitted. The start `<form>` and end `</form>` tags designate an area of a Web page as a form. Between the `<form>` and `</form>` tags, form controls can be added to request different types of information and allow the appropriate input responses. A form can include any number of controls.

BTW CGI Scripts

Using CGI scripts to process forms is a much more efficient way to handle the data that is sent from a form. Many Web sites have free sample CGI scripts for Web developers to use. Search the Web for relevant CGI information and free samples.

BTW

mailto Restrictions

Some schools and organizations restrict the use of the `mailto` action. This is something that you need to test in order to determine whether it will work in your situation.

The **action attribute** of the `<form>` tag specifies the action that is taken when the form is submitted. Information entered in forms can be sent by e-mail to an e-mail address or can be used to update a database. Although the e-mail option is functional, many Web sites process information from forms using Common Gateway Interface (CGI) scripting. A **CGI script** is a program written in a programming language (such as PHP or Perl) that communicates with the Web server. The CGI script sends the information input on the Web page form to the server for processing. Because this type of processing involves programming tasks that are beyond the scope of this book, the information entered in the order form created in this chapter will be submitted in a file to an e-mail address. The e-mail address will be specified as the action attribute value in the `<form>` tag.

The **method attribute** of the `<form>` tag specifies the manner in which the data entered in the form is sent to the server to be processed. Two primary ways are used in HTML: the get method and the post method. The **get method** sends the name-value pairs to the end of the URL indicated in the action attribute. The **post method** sends a separate data file with the name-value pairs to the URL (or e-mail address) indicated in the action attribute. Most Web developers prefer the post method because it is much more flexible. You need to be cautious when using the get method. Some Web servers limit a URL's size, so you run the risk of truncating relevant information when using the get method. The post method is used for the forms in this chapter.

The following HTML code creates a form using the post method and an action attribute to indicate that the form information should be sent to an e-mail address in an attached data file:

```
<form method="post" action="mailto: sabatinas@isp.com">
```

When the form is submitted, a file containing the input data is sent as an e-mail attachment to the e-mail address `sabatinas@isp.com`.

To Create a Form and Identify the Form Process

The following step shows how to enter HTML code to create a form and identify the form process.

1

- Highlight the words `<!--Insert form` `method statement` `here -->` on line 13.
- Type `<form` `method="post"` `action="mailto:` `sabatinas@isp` `.com">` to replace the highlighted words with the new tag.
- Click the blank line 62 and press the ENTER key.
- Type `</form>` but do not press the ENTER key (Figure 6–8).

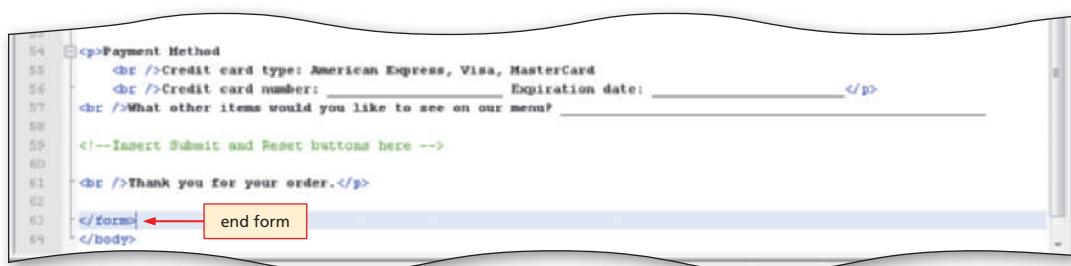


Figure 6–8

To Change the Text Message

The next step in updating the text-based Order Form Web page is to modify the text that tells the user to submit the questionnaire by e-mail. Table 6–4 shows the new HTML code used to provide instructions to users on how to submit the information on the order form.

Table 6–4 HTML Code to Change the Text Message

Line	HTML Tag and Text
18	<td>Once you have selected the items that you want to order and the payment type, click the Submit button to process the order.</td>

The following step illustrates how to change the text message to provide instructions on how to use the order form.

1

- Highlight line 18.
- Enter the HTML code shown in Table 6–4, indenting as shown, but do not press the ENTER key (Figure 6–9).
- Highlight lines 22 through 57 (ending above the blank line 58) and then press the DELETE key (Figure 6–9).

```

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Sabatina's Pizza Order Form</title>
    <!-- Insert embedded style -->
  </head>
  <body>
    <form method="post" action="mailto:sabatinas@isp.com">
      <table>
        <tr>
          <td></td>
          <td>Once you have selected the items that you want to order and the payment type, click the Submit button to process the order.</td>
        </tr>
      </table>
      <!-- Insert Submit and Reset buttons here -->
    </form>
  </body>
</html>

```

Figure 6–9

Form controls.

Before creating a Web page form, you should plan how you want to format it. By effectively utilizing input controls, you can call attention to important data-collection areas on the Web page without overpowering it. Creating an effective form includes:

1. **Determine what data to collect.** In the case of a form designed to sell a product, you need the visitor's name and address information. Make sure to provide enough space for each field so that you do not cut out important information. For instance, an address field only 10 characters long may cut out much of the street name.
2. **Determine what types of control to use.** For data such as name and address, you need text input areas. For data such as credit card type, there is a limited subset (i.e., American Express, Visa, and MasterCard), so a selection control is appropriate. When you ask what kind of pizza the visitor is interested in buying, you can use check boxes, which allow multiple selection. In the case of an answer with only two choices (i.e., Yes/No or Thick or Thin pizza?), a radio button is more appropriate.

Plan Ahead

(continued)

(continued)

3. **Lay out the input areas effectively.** One of the first input items you may want is the visitor's name and address information. That should go to the top of the page. Also, you can group information together on the same line if it makes sense to make the Web page form short enough that visitors do not have to scroll much. Notice in our order form that the city/state/ZIP are on one line of the Web page. Another thing to consider is the use of e-mail addresses beyond the initial contact with a visitor or customer. An e-mail address is a great way to continue communication with visitors or customers. A company can e-mail newsletters, coupons, and general information to customers once they have their e-mail addresses.
4. **Use grouping techniques for clarity.** The last thing that you may want to do on a Web page form is group like input items together. Use the fieldset tag to segregate personal information from order information and from other comments that the visitor might make.

Adding Text Boxes

As previously discussed, a text box allows for a single line of input. The HTML code below shows an example of the code used to add a text box to a form:

```
<input name="address" type="text" size="25" maxlength="25" />
```

The `<input />` tag creates an input control, while the attribute and value `type="text"` specifies that the input control is a text box. The `name` attribute of the input control is set to the value `address`, to describe the information to be entered in this text box. When the form is submitted, the `name` is used to distinguish the value associated with that field from other fields.

The `size` attribute indicates the size of the text box that appears on the form. In the following HTML code, `size="25"` sets the text field to 25 characters in length, which means that only 25 characters will appear in the text box. The `maxlength` attribute `maxlength="25"` limits the number of characters that can be entered in the text box to 25 characters. The `maxlength` attribute specifies the same number of characters (25) as the `size` attribute (25), so all characters entered by a user will appear in the text box. If you specify a maximum number of characters that is greater than the number of characters specified in the `size` attribute, the additional characters scroll to the right in the text box as the user enters them.

To Add Text Boxes

The next step in creating the order form is to add seven text boxes to the form for users to enter first name, last name, street address, city, state, ZIP, and e-mail address. Table 6–5 shows the HTML code to add seven text boxes to the form. Each text box has a size of 25 characters, except the ZIP text box, with only 10 characters. No `maxlength` attribute is specified, which means users can enter text items longer than 25 characters, but only 25 characters will display in the text box.

Table 6–5 HTML Code to Add Text Boxes

Line	HTML Tag and Text
22	<code><p>Last Name: <input name="lastname" type="text" size="25" /></code>
23	<code>First Name: <input name="firstname" type="text" size="25" /></code>
24	<code>
Street Address: <input name="address" type="text" size="25" /></code>
25	<code>
City: <input name="city" type="text" size="25" /></code>
26	<code>State: <input name="state" type="text" size="25" /></code>
27	<code>ZIP: <input name="zip" type="text" size="10" /></code>
28	<code>E-mail Address: <input name="email" type="text" size="25" /></p></code>

The following step illustrates how to add text boxes to the form.

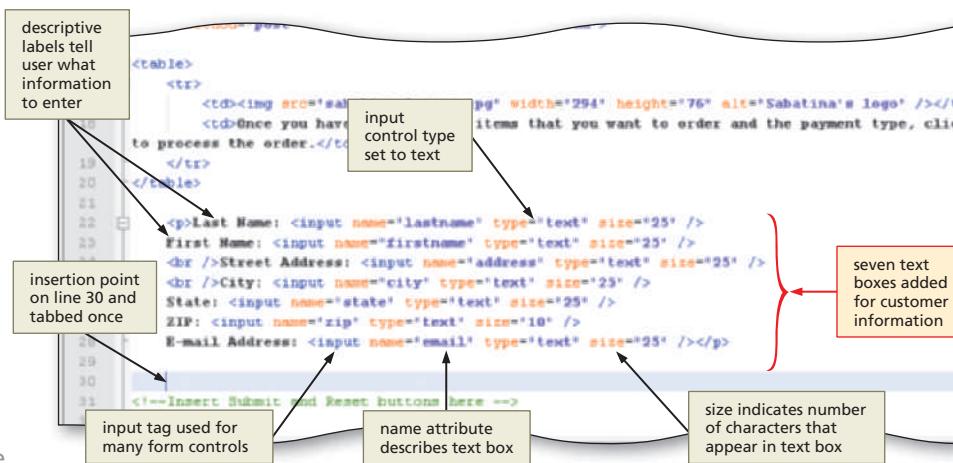
1

- If necessary, click line 22.
- Enter the HTML code shown in Table 6–5 and then press the ENTER key twice (Figure 6–10).

Q&A

How do I know what size to make each field?

Determine a reasonable field size for the various input areas. For instance, it would not be wise to allow only 10 characters for the last name, because many people now hyphenate their last names and last names can be more than 10 characters. To improve your judgment for field sizes, observe online and paper forms that you complete. Also, think of long street or city names and try those in the forms that you create.

**Figure 6–10****Q&A**

What is the default value if I do not specify the type in my <input /> tag?

The default type for the <input /> tag is a text box. Therefore, if the type attribute is not used in the <input /> tag, it creates a text box.

To Save an HTML File

Because you opened a file that was already created, you can press the **CTRL+S** keys to save what you have done so far. Do this frequently throughout your projects.

- With the USB flash drive connected to one of the computer's USB ports, hold down the **CTRL** key and then press the **S** key.

Other Ways

- Click File, then Save
- Click Save icon on Notepad++ toolbar

Adding Radio Buttons

The next step is to add radio buttons to the form. Remember that radio buttons are appropriate to use when a user can select only one choice from a set of two or more choices. Questions with a Yes or No answer are perfect for the use of radio buttons. In the case of Thick or Thin (pizza) and Medium and Large, you have one of two choices to make. More than two options are possible with radio buttons also. In the case of Cheese, Sausage, or Veggie, you can only select one of the three choices. On the Order Form Web page, radio buttons allow users to select a one-choice answer to a question.

To Add Radio Buttons

Table 6–6 contains the HTML code to add a set of radio buttons to the Order Form Web page.

Table 6–6 HTML Code to Add Radio Buttons

Line	HTML Tag and Text
30	 Pizza (select type, size, and ingredients):
31	 <input name="pizzatype" type="radio" value="thin" />Thin
32	<input name="pizzatype" type="radio" value="thick" />Thick
33	
34	 <input name="size" type="radio" value="medium" />Medium
35	<input name="size" type="radio" value="large" />Large
36	
37	 <input name="ingredient" type="radio" value="cheese" />Cheese
38	<input name="ingredient" type="radio" value="sausage" />Sausage
39	<input name="ingredient" type="radio" value="veggie" />Veggie

The following step illustrates how to add three sets of radio buttons to the form.

1

- Check that the insertion point is on Line 30, indented one Tab stop.
- Enter the HTML code shown in Table 6–6 and then press the ENTER key twice (Figure 6–11).

Q&A

Could I have used check boxes for this control, rather than radio buttons?

You could have used check boxes, but it would not make sense for this information. In this case, this is a clear yes or no answer. With check boxes, you are assuming that they can make multiple selections. Again, look at the standards used in most Web development.

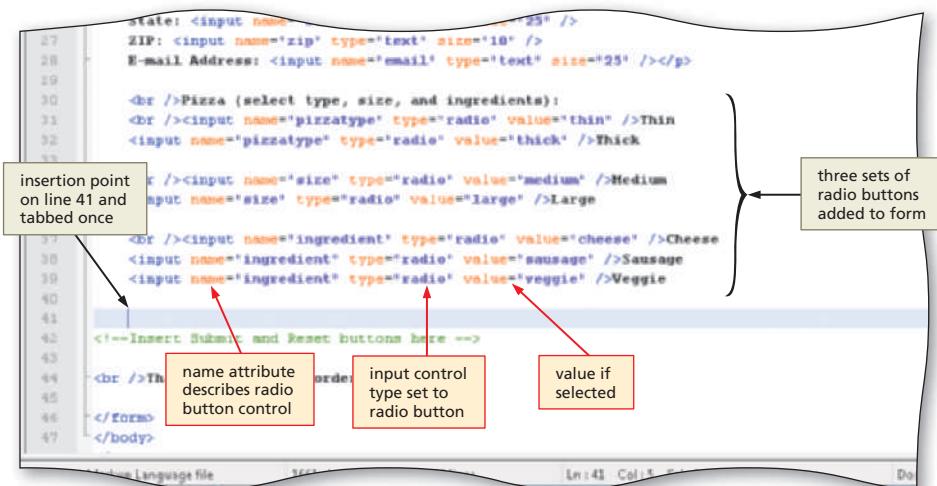


Figure 6–11

Adding Check Boxes

Check boxes are similar to radio buttons, except they allow multiple options to be selected. Radio buttons should be used when only one option can be selected, while check boxes should be used when the user can select more than one option.

The HTML code below shows an example of the code used to add a check box to a form:

```
<input name="additional" type="checkbox" value="mushrooms" />
Mushrooms
```

The `<input />` tag creates an input control, while the attribute and value `type="checkbox"` specifies that the input control is a check box. The name attribute of the input control is set to the value `additional`. When the form is submitted, the name is used to distinguish the values associated with these checkbox fields from other fields. The value attribute "mushrooms" indicates the value submitted in the file, if this check box is selected.

To Add Check Boxes

In the Order Form Web page, three check boxes are used to allow the user to select one or more types of ingredients to add. Table 6–7 shows the HTML code to add three check boxes to the form.

Table 6–7 HTML Code to Add Check Boxes

Line	HTML Tag and Text
41	<code><p>Add the following:</code>
42	<code><input name="additional" type="checkbox" value="mushrooms" />Mushrooms</code>
43	<code><input name="additional" type="checkbox" value="olives" />Olives</code>
44	<code><input name="additional" type="checkbox" value="pepperoni" />Pepperoni</p></code>

The step that follows illustrates how to enter HTML code to add check boxes to the form.

1

- Check that the insertion point is on Line 41, indented one Tab stop.
- Enter the HTML code shown in Table 6–7 and then press the ENTER key twice (Figure 6–12).

Q&A

How do I determine whether to list fields on the same line or use a line break or paragraph break between fields?

Consider the "real estate" (the amount of space available) of the Web page itself. If you have an especially long form that the visitor has to scroll down, consider positioning the fields across, rather than down the form. You do not want to crowd the information, but you also do not want to force the visitor to scroll excessively.

The screenshot shows a code editor with the following HTML code:

```

<p>Add the following:<br />
<input name='size' type='checkbox' value='medium' />Medium<br />
<input name='size' type='checkbox' value='large' />Large<br />
<br />
<input name='ingredient' type='radio' value='cheese' />Cheese<br />
<input name='ingredient' type='radio' value='sausage' />Sausage<br />
<input name='ingredient' type='radio' value='veggie' />Veggie<br />
<br />
<!-- Add the following: --><br />
<input name='additional' type='checkbox' value='mushrooms' />Mushrooms<br />
<input name='additional' type='checkbox' value='olives' />Olives<br />
<input name='additional' type='checkbox' value='pepperoni' />Pepperoni</p>
<br />
<!-- Insert Submit and Reset buttons here -->
<br />
<br />
<-->Thank you for your order.</p>

```

Annotations in the screenshot include:

- A callout box labeled "input control type set to checkbox" points to the first `<input>` tag on Line 42.
- A callout box labeled "three check boxes added to form" points to the three `<input>` tags on Lines 42, 43, and 44.
- An arrow labeled "insertion point on line 46 and tabbed once" points to the cursor position at the start of Line 46.

Figure 6–12

Adding a Selection Menu

A select control is used to create a selection menu from which the visitor selects one or more choices. A select control is suitable when a limited number of choices are available. Figure 6–13 on the next page shows the basic selection menu used in the order form, with three credit card types (Visa, MasterCard, and American Express) as the choices in the list. The topic of accepting credit cards via online forms is a serious one. See BTW on this page for more information.

BTW

Security

Security is an important issue to understand, especially when you are collecting credit card information on an online form. Search the Web for specific information concerning the usage of https SSL sites versus http sites. Also search for information regarding the vulnerability of sending unencrypted e-mail.



Figure 6–13 Select control.

If you do not specify a size attribute, only one option is displayed, along with a list arrow, as shown in Figure 6–13. When the list arrow is clicked, the selection menu displays all selection options. When the user selects an option, such as Visa, in the list, it appears as highlighted.

To Add a Selection Menu

Table 6–8 shows the HTML code used to create the selection menu shown in Figure 6–13.

Table 6–8 HTML Code to Add a Selection Menu

Line	HTML Tag and Text
46	<p>Credit card type:
47	<select name="payment">
48	<option>American Express</option>
49	<option>Visa</option>
50	<option>MasterCard</option>
51	</select>

The following step illustrates how to add a selection menu to the Web page form.

1

- Check that the insertion point is on Line 46, indented one Tab stop.
- Enter the HTML code shown in Table 6–8 and then press the ENTER key twice (Figure 6–14).

Q&A

How do I know what control type to use?

Consider the Web page “real estate,” together with usability. If you have 20 options, it may not make sense to use a select control. With the three credit card options (American Express, Visa, MasterCard) it makes sense to use a select control, as users are familiar with this model. You can also look at the types of controls other Web developers use, and apply those that make sense for your situation.

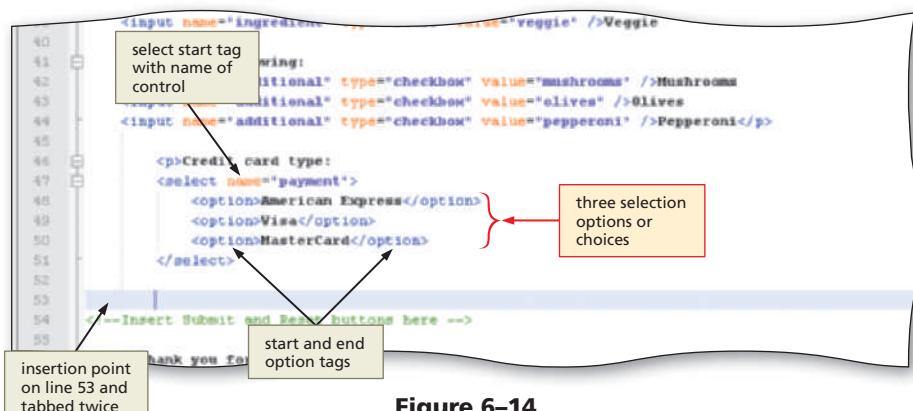


Figure 6–14

Adding More Advanced Selection Menus

Selection menus have many variations beyond the simple selection menu used in the Order Form Web page. Table 6–3 on pages HTML 293 and HTML 294 lists several attributes for the <select> tag. Using these attributes, a selection menu can be set to display multiple choices or only one, with a drop-down list to allow a user to select another choice. A selection menu also can be defined to have one choice preselected as the default.

Figure 6–15 shows samples of selection menus. The HTML code used to create each selection menu is shown in Figure 6–16 on the next page.

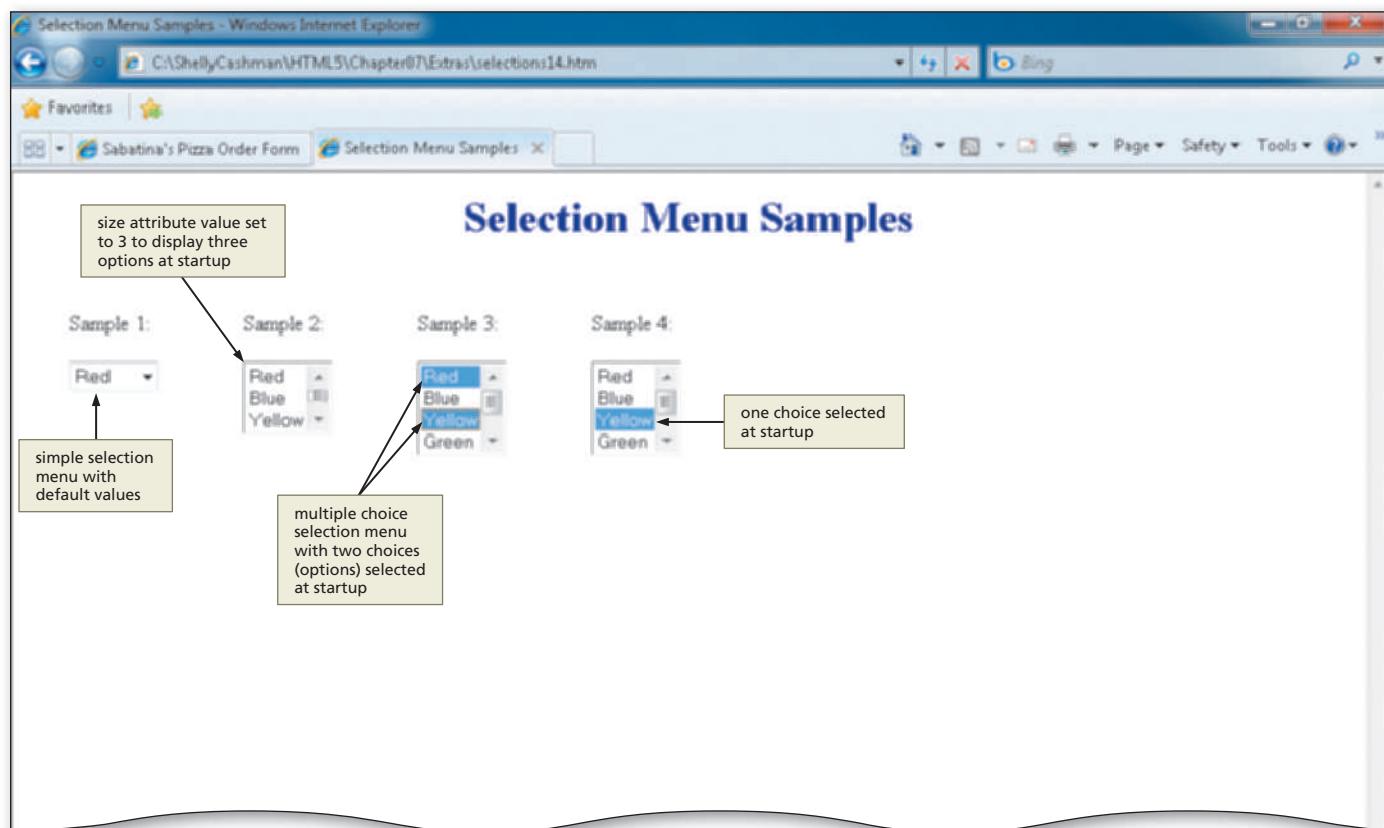


Figure 6–15 Sample selection controls with variations.

The selection menu in Sample 1 is a basic selection menu, with no attributes specified other than the name and the list options. This resulting selection menu uses a list menu that allows users to select one choice from the list. No choice is selected by default. The selection menu in Sample 2 uses a size attribute value of 3 to indicate that three choices should appear in the menu at startup. A user can use the up and down scroll arrows to view other choices in the list. The selection menu in Sample 3 uses the multiple attribute to allow a user to select more than one choice in the list. To select multiple choices, a user first must select one choice and then press and hold the **CTRL** key while clicking other choices in the list. If a user wants to select several consecutive choices, he or she can select the first choice and then press and hold the **SHIFT** key while selecting the last choice. All choices between the first choice and last choice automatically will be selected. The selection menu in Sample 4 also contains the multiple attribute, so one or more choices can be selected. In addition, Sample 4 provides an example of one choice (in this case, Yellow) being selected at startup. As shown in the HTML code in Figure 6–16, the selected attribute is included in the <option> tag for Yellow, to indicate that Yellow should be selected at startup.

BTW

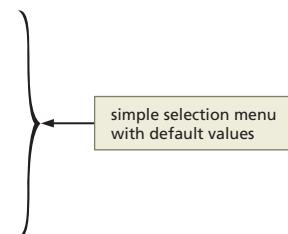
Options

The `<select>` and `<option>` tags are useful when you have a limited number of choices from which a Web page visitor can select. If the number of options becomes too large, however, the list is difficult to read. A better idea might be to group together like options into submenus. You can use the `<optgroup>` tag before the first `<option>` tag in the first group that you want to use in a submenu. After the last option in that group, use the `</optgroup>` tag.

The purpose of the selection menu dictates the type of selection menu that should be used and the HTML code required to create that select control. Using the basic tags and attributes shown in Figure 6–16, you can create a wide variety of selection menus to suit almost any purpose.

Sample 1:

```
<select name="color">
    <option>Red</option>
    <option>Blue</option>
    <option>Yellow</option>
    <option>Green</option>
    <option>Black</option>
    <option>White</option>
</select>
```



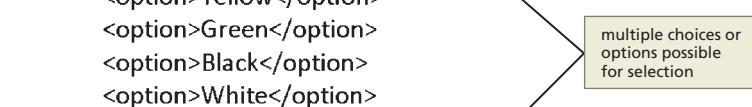
Sample 2:

```
<select name="color" size="3">
    <option>Red</option>
    <option>Blue</option>
    <option>Yellow</option>
    <option>Green</option>
    <option>Black</option>
    <option>White</option>
</select>
```



Sample 3:

```
<select name="color" multiple="multiple">
    <option>Red</option>
    <option>Blue</option>
    <option>Yellow</option>
    <option>Green</option>
    <option>Black</option>
    <option>White</option>
</select>
```



Sample 4:

```
<select name="color" multiple="multiple">
    <option>Red</option>
    <option>Blue</option>
    <option selected="selected">Yellow</option>
    <option>Green</option>
    <option>Black</option>
    <option>White</option>
</select>
```

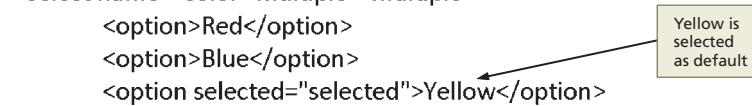


Figure 6–16 HTML code to create selection menus.

To Add Additional Text Boxes

The next step in creating the Order Form Web page is to add two more text boxes for credit card number and expiration date. Table 6–9 shows the HTML code used to add the additional text boxes. A text field is used rather than a textarea field because the user needs to enter only one row of characters.

Table 6–9 HTML Code to Add Additional Text Boxes

Line	HTML Tag and Text
53	Credit card number:
54	<input name="cardnum" type="text" size="20" maxlength="20" />
55	
56	Expiration date:
57	<input name="cardexp" type="text" size="4" maxlength="4" /></p>

The following steps illustrate how to add two additional text boxes to the Web page form.

- 1 Check that the insertion point is on Line 53, indented two Tab stops.
- 2 Enter the HTML code shown in Table 6–9 and then press the ENTER key twice (Figure 6–17). Use the BACKSPACE key, if necessary, to align the insertion point at the beginning of Line 59.

```

46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
<p>Credit card type:<br/>
<select name='payment'>
<option>American Express</option>
<option>Visa</option>
<option>MasterCard</option>
</select>
<br/>
Credit card number:<br/>
<input name='cardnum' type='text' size='20' maxlength='20' />
<br/>
Expiration date:<br/>
<input name='cardexp' type='text' size='4' maxlength='4' /></p>
<br/>
<!--Insert Submit button here -->
<br/>
for your order.</p>

```

Figure 6–17

Adding a Textarea Box

The next step is to add a textarea to the form. Remember that a textarea is used when you want a multiple-row input area. The text control only allows a user to input one row of information. For multiple rows, use the textarea control.

The order form includes a textarea that allows the user to add additional comments about what other items could be included on the menu. Because the response can be longer than just one line, a textarea control is used.



Feedback

One good use of forms is to get feedback from your visitors. Suggestions from visitors not only can help improve the Web site, but can give your visitors the sense that you care about their opinions. Taking visitor feedback into account provides for better customer satisfaction.

To Add a Textarea Box

The next step is to add a textarea to the form. You use a textarea because you want the user to be able to input more than one line. Table 6–10 contains the tags and text to specify a textarea for multiple-line input.

Table 6–10 HTML Code to Add a Textarea

Line	HTML Tag and Text
59	 What other items would you like to see on our menu?
60	 <textarea name="other" rows="3" cols="100"></textarea>

The following step illustrates how to add a textarea to the order form.

1

- If necessary, click line 59.
- Enter the HTML code shown in Table 6–10 and then press the ENTER key (Figure 6–18).

Q&A How do I know how big to make the textarea box?
Again, you need to look at the standards used in most Web development. You also have to view the textarea box in the browser to see how the size affects the Web page form. For instance, if we had made the number of rows 4, rather than 3, the Web page visitor would not have seen the “Thank you for your order.” message on the bottom of the page. That is something that you want the visitor to see without scrolling.

Figure 6–18

BTW

Submit and Reset Buttons

A simplistic, default button is created when you use the `type="submit"` attribute and value or the `type="reset"` attribute and value within the `<input>` tag. The `<button>` tag also can be used to create a submit button. The `<button>` tag gives you the option of using an image for the button, rather than using the default button style. The appearance of the button text can be changed with the `<style>` tag. These tags give you more flexibility when creating submit or reset buttons.

Adding Submit and Reset Buttons

The form controls are useless unless the information entered in the form can be submitted for processing. The next step in creating the order form is to add two buttons at the bottom of the Web page form. The first button, Submit, is for submitting the form. When the visitor clicks this button, the data entered into the form is sent to the appropriate location for processing. The second button, Reset, clears any data that was entered in the form.

The HTML code below shows the `<input />` tags used to create the Submit and Reset buttons on a Web page form:

```
<p><input type="submit" value="Submit" />
<input type="reset" value="Reset" /></p>
```

The first line of HTML code creates a Submit button on the Web page form. A Submit button is created by using the attribute `type="submit"` in an `<input />` tag. The `value` attribute is used to indicate the text that should appear on the button face — in this case, Submit.

When a user clicks the Submit button, all data currently entered in the form is sent to the appropriate location for processing. The action taken when a form is submitted is based on the method and action attributes specified in the `<form>` tag. In the `<form>` tag at the start of this form, the HTML code set the form attributes to `method="post"` and `action="mailto:sabatinas@isp.com"`. Thus, when a user clicks the Submit button, a data file that contains all the input data automatically is sent as an e-mail attachment to the e-mail address `sabatinas@isp.com`. By default, the data file is named `Postdata.att`.

The code below shows a sample of the data file that is sent to the e-mail address, when using the post method. In this sample, the user checked the radio button for a Thin crust, Sausage pizza, and is paying with an American Express card:

```
payment=American Express&pizzatype=thin&ingredient=sausage
```

The data entered in the form appears in the data file as name-value pairs — the name of the control as specified in the name attribute, followed by the value entered or selected in the control. In the above example, the user selected American Express in the selection menu, which is controlled by the field named payment, and clicked the Thin radio button, which is controlled by the field named pizzatype. The user also selected the Sausage radio button, which is controlled by the field named ingredient. An ampersand (&) strings together all of the name-value pairs to make them easier to read. The receiver of this e-mail could take the information in this format and decipher what is being requested. A simple program, written in a different programming language such as C++ or Java, could be developed that reads the information in the e-mail and transforms it into a more usable format. As mentioned earlier in the chapter, a CGI script is a better way to collect large amounts of information from forms, but that is beyond the scope of this book.

The Reset button also is an important part of any form. Resetting the form clears any information previously typed into a text box or textarea and resets radio buttons, check boxes, selection menus, and other controls to their initial values. As shown in the second line of the HTML code above, a Reset button is created by using the attribute type="reset" in an <input /> tag. The value attribute is used to indicate the text that should appear on the button face — in this case, Reset.

To Add Submit and Reset Buttons

The following step illustrates how to add a Submit button and a Reset button to the form.

1

- Highlight the statement <!--Insert Submit and Reset buttons here --> on line 62.
- Type <p><input type="submit" value="Submit" /> to create the Submit button and then press the ENTER key.
- Type <input type="reset" value="Reset" /> to create the Reset button. Do not press the ENTER key (Figure 6–19).

```

53 Credit card number:
54 <input name="cardnum" type="text" size="20" maxlength="20" />
55
56 Expiration date:
57 <input name="cardexp" type="text" size="4" maxlength="4" /></p>
58
59 <p>other items would you like to
60 <area name="other" rows="3" cols="40" /></p>
61
62 <p><input type="submit" value="Submit" />
63 <input type="reset" value="Reset" /></p>
64 </p>you for your order</p>
65
66 </body>
67

```

Figure 6–19

Q&A

That submit option seems very easy to use. Do I need to do anything else in order to process the data?

No, the Submit button works in conjunction with the statements that you provided in your form tag in order to process the data entered.

Q&A

Why do I need the Reset button?

It is best always to provide a Reset button next to the Submit button. This is useful to clear all of the data entered in case your visitors want to start over or if they change their minds or make mistakes.

Q&A

If a visitor uses the Reset button, what does that do to default values that I have included in the tags?

Reset will set those default values back to the original values included in the tags. In other words, if you use a default value, Reset does not clear that value.

Plan Ahead

Organizing a form.

When using fieldset tags to separate and organize information on a form, consider the following:

- **Required vs. optional information.** You can group all required information into one section of the form and place all optional information into another grouping. By doing this, you call attention immediately to the required information on the form.
- **General organization.** It can be helpful to enhance the look and feel of the form with groupings. Especially in the case of a long form, using separators helps direct the visitors' attention.

BTW

Groupings

An important part of good Web design is to make a form easy to use. Your Web site visitors are more likely to complete a form if they readily understand the information that is being requested. You can use the <fieldset> tag to group like information together.

Organizing a Form Using Form Groupings

An important aspect of creating a Web page form is making the form easy for Web site visitors to understand. Grouping similar information on a form, for example, makes the information easier to read and understand — and, as a result, easier to complete. Grouping is especially helpful in cases where some information is required and some is optional. In the order form, for example, all the personal information is required (for example, name, address, and credit card number). The order information (i.e., pizza type, size, and ingredient) is also required. The final question on the form, however, (What other items would you like to see on our menu?) is optional. The form thus should be modified to group required and optional information.

A **fieldset** control is used to group similar information on a form. The HTML code below shows the <fieldset> tag used to add a fieldset control to a Web page form:

```
<fieldset><legend align="left">Customer Information</legend>
</fieldset>
```

The <legend> tag within the fieldset tag is optional. Using the <legend> tag creates a legend for the fieldset, which is the text that appears in the grouping borders, as shown in the example in Figure 6–20. The align attribute is used to align the legend to the left or right of the fieldset control.

The screenshot shows a web browser displaying the 'Bill Thomas Illustrations Order Form'. The form includes several input fields and legends. A legend labeled 'left-aligned legend' is positioned above a group of text input fields for Name, Street Address, City, State, ZIP, and E-mail Address. Another legend, 'right-aligned legend', is positioned to the right of a group of checkboxes for selecting picture types: Cross Hatch, Full Color, and Ink Wash. A third legend, 'right-aligned legend', is located above a section asking if the user wants to receive an e-mail newsletter. Brackets on the right side of the form group these three sections together and label them as the 'first fieldset control'. Below this, another bracket groups a text area for additional purchase interests and labels it as the 'second fieldset control'. At the bottom of the form are 'Submit the Form' and 'Reset the Form' buttons.

Figure 6–20 Fieldset controls on a form.

In the Order Form Web page that you will create, three fieldset controls are added to group similar information on the form. The first fieldset control is used to group customer information, as shown in Figure 6–21. The first fieldset control has the legend, Customer Information, aligned to the left. The second fieldset control is used to group order information. The second fieldset control has the legend, Your Order, aligned to the left. The third fieldset control, used to group the payment information, has the legend, Payment Method, aligned to the left. These groupings divide the form so it is more readable and clearly defines what information is required and what is optional.

Figure 6–21

To Add Fieldset Controls to Create Form Groupings

The following step shows how to add three sets of fieldset tags to create information groupings on the Web page form.

1

- Click just before <p>Last Name at the beginning of line 22, and then press the ENTER key.

```

<table>
  <tr>
    <td><img src='sabatinaslogo.jpg' width='294' height='76' alt='Sabatina's logo' /></td>
    <td>Once you have selected the items that you want to order and the payment type, click the Submit button to process the order.</td>
  </tr>
</table>
<fieldset><legend>Customer Information</legend>
<p>Last Name: <input name="lastname" type="text" size="25" />
First Name: <input name="firstname" type="text" size="25" />
<br />Street Address: <input name="address" type="text" size="25" />
<br />City: <input name="city" type="text" size="25" />
State: <input name="state" type="text" size="25" />
ZIP: <input name="zip" type="text" size="10" />
E-mail Address: <input name="email" type="text" size="25" /></p>
</fieldset>
<fieldset><legend>Your Order</legend>
</fieldset>
<fieldset><legend>Payment Method</legend>
</fieldset>

```

Figure 6–22 (a)

Customer Information</legend> as the tag to begin the first fieldset (Figure 6–22a).

- Click just after the </p> on line 29, and then press the ENTER key.
- If necessary, use the BACKSPACE key to move back to the left margin, type </fieldset> on line 30 to end the first fieldset, and then press the ENTER key.

2

- With the insertion point on line 32, type `<fieldset>` `<legend>Your Order</legend>` to start the second fieldset.
- Click to the right of the `<p>` on line 47 (at the end of the `Pepperoni</p>` line), and then press the ENTER key.
- If necessary, use the BACKSPACE key to move back to the left margin, type `</fieldset>`, and then press the ENTER key.

```

<legend>Your Order</legend>
<br /><input name="pizzatype" type="radio" value="thin" />Thin
<input name="pizzatype" type="radio" value="thick" />Thick

<br /><input name="size" type="radio" value="medium" />Medium
<input name="size" type="radio" value="large" />Large

<br /><input name="ingredient" type="radio" value="cheese" />Cheese
<input name="ingredient" type="radio" value="sausage" />Sausage
<input name="ingredient" type="radio" value="veggie" />Veggie

<p>Add the following:
<input name="additional" type="checkbox" value="mushrooms" />Mushrooms
<input name="additional" type="checkbox" value="olives" />Olives
<input name="additional" type="checkbox" value="pepperoni" />Pepperoni</p>

<legend>Payment Method</legend>
<p>Credit card type:
<select name="payment">
<option>American Express</option>
<option>Visa</option>
<option>MasterCard</option>
</select>

Credit card number:
<input name="cardnum" type="text" size="20" maxlength="20" />

Expiration date:
<input name="cardexp" type="text" size="4" maxlength="4" /></p>

```

The code editor shows the creation of three fieldsets. A legend is added above the first fieldset. The second fieldset starts at line 32, and the third fieldset starts at line 50. The code is color-coded for syntax highlighting.

Figure 6–22 (b)

Q&A

What is the default value for the `<legend>` alignment?

If you do not indicate otherwise, the legend will align left.

Q&A

Are there other options such as colored borders that I can use with the `<fieldset>` tag?

Yes, you can set the margins, font, colors, etc. for the `<fieldset>` tag. You can use an inline, embedded, or external Cascading Style Sheet (CSS) for those options.

Adding an Embedded Style Sheet

The next step in creating the Order Form Web page is to add an embedded style sheet to improve the look of the legends in all three field groupings. Figure 6–23a shows the Web page as it displays currently without the change in legend style. Note that the default position for the legend is on the left, and the legend is the default color, black. Figure 6–23b shows the Web page with the new style for the legends as inserted in the embedded style sheet. Remember that an embedded style sheet changes the style for a single Web page. The embedded style sheet is inserted within the `<style></style>` container in the `<head>` section of the Web page.

(a) This screenshot shows a standard form layout. It includes sections for Customer Information, Your Order, and Payment Method. Each section has a label followed by input fields. The 'Your Order' section contains a legend: 'Pizza (select type, size, and ingredients):' with radio buttons for Thin, Thick, Medium, Large, Cheese, Sausage, and Veggie.

(b) This screenshot shows the same form layout but with CSS applied to the legends. A legend is labeled 'small, italic, red legends floated on the left'. The 'Your Order' section now has an italicized, red-colored legend: 'Your Order'.

Figure 6–23

In this style sheet, you change the color, font-style, and font-size of the legend (lines 13 through 15), making it red (to match the Sabatina's logo) and smaller than the surrounding text. You also use the float property (line 12) to align the legend on the left. In Chapter 3, you used the float property to align images relative to the Web page text. The float property indicates in which direction (in this case left or right) to display (or float) an element (the legend) being inserted on a Web page. It can be an effective Web development practice to use an alignment element such as float so that you can easily change the alignment if you choose.

To Add an Embedded Style Sheet

Table 6–11 shows the HTML code to add an embedded style sheet to format the legends for the form fields.

Table 6–11 HTML Code to Add an External Style Sheet

Line	HTML Tag and Text
9	<style type="text/css">
10	<!--
11	
12	legend {float: left;
13	color: #ff1828;
14	font-style: italic;
15	font-size: small}
16	
17	-->
18	</style>

The following steps illustrate how to add an embedded style sheet to the Web page form.

- 1 Highlight the statement <!--Insert embedded style sheet here --> on line 9.
- 2 Enter the HTML code shown in Table 6–11 but do not press the ENTER key on line 18 (Figure 6–24).

The screenshot shows a code editor with the following annotations:

- start style tag on line 9**: Points to the opening `<style type="text/css">` tag.
- float property added for alignment**: Points to the `float: left;` declaration in the style block.
- red color added to match logo**: Points to the `color: #ff1828;` declaration in the style block.
- font style and size altered**: Points to the `font-style: italic;` and `font-size: small` declarations in the style block.
- end style tag**: Points to the closing `</style>` tag.
- embedded style sheet added to enhance legends**: A bracket groups the entire style block, indicating its purpose.

```

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Sabatina's Pizza</title>
<style type="text/css">
<!--
    legend {float: left;
        color: #ff1828;
        font-style: italic;
        font-size: small}
-->
</style>
</head>
<body>
<form method="post" action="mailto:sabatinas@isp.com">
<table>

```

Figure 6–24

To Save the HTML File

With the Order Form Web page complete, the HTML file should be saved. The following step illustrates how to again save the `sabatinasorderform.html` file on the USB drive.

- 1** With a USB drive plugged into your computer, click **File** on the menu bar and then click **Save** to again save the `sabatinasorderform.html` file.
-

To Validate, View, and Test a Web Page

After completing the Sabatina's Order Form Web page, you should validate the code, and view and test it in a browser to confirm that the Web page appears as desired and that the controls function as expected. Note that you cannot test the **Submit** button because it automatically generates an e-mail message to `sabatinas@isp.com`, which is a nonexistent e-mail address. When you are collecting information from an online form, it is very important to test that the information is accurate. It is beyond the scope of this book to address validation issues related to server-side processing. After testing the controls, each Web page and the HTML code for each should be printed for future reference. The following steps illustrate how to validate, view, and test a Web page.

- 1** Validate the `sabatinasorderform.html` file by file upload at validator.w3.org.
 - 2** In Internet Explorer, click the **Address** bar to select the URL on the **Address** bar, type `g:\Chapter06\ChapterFiles\sabatinasorderform.html` or the location of your file and then press the **ENTER** key to display the completed order form for Sabatina's Pizza (Figure 6–25 on the next page).
 - 3** Review the form to make sure all spelling is correct and the controls are positioned appropriately.
 - 4** Test all of the text boxes on the form. Try to type more than the maximum number of allowable characters in the `cardnum` and `cardexp` boxes.
 - 5** Click the check boxes to test them. You should be able to choose one, two, or three of the boxes at the same time because check boxes are designed to select more than one option.
 - 6** Test the selection control by clicking the list arrow and selecting one of the three options.
 - 7** Click the radio buttons to test them. You should be able to make only one choice (Thick or Thin; Medium or Large; Cheese, Sausage, or Veggie).
 - 8** Test the `textarea` by entering a paragraph of text. Verify that it allows more characters to be entered than are shown in the `textarea`.
 - 9** Click the **Reset** button. It should clear and reset all controls to their original (default) state.
-

The screenshot shows a pizza order form titled "Sabatina's Pizza Order Form". It includes sections for "Customer Information", "Your Order", "Payment Method", and a feedback section. Various form controls like text inputs, radio buttons, checkboxes, and a textarea are highlighted with callout boxes containing test instructions.

- A callout box labeled "test selection control" points to a group of radio buttons under "Your Order". Another callout box labeled "test that only one radio button can be selected per section" also points to these buttons.
- A callout box labeled "test that more than one check box can be selected" points to a group of checkboxes for toppings.
- A callout box labeled "test all text controls in first fieldset grouping" points to the "Customer Information" section.
- A callout box labeled "test Reset button" points to the "Reset" button in the "Payment Method" section.
- A callout box labeled "test textarea control" points to the large text area for feedback.

Figure 6–25

To Print a Web Page and HTML

- 1 Click the Print icon on the Command bar to print the Web page.
- 2 Click the Notepad++ button on the taskbar to activate `sabatinasorderform.html`.
- 3 Click File on the menu bar and then click Print. Click the Print button in the Print dialog box to print the HTML file (Figure 6–26).

```

<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Sabatina's Pizza Order Form</title>
<style type="text/css">
</style>
</head>
<body>

legend {float: left;
color: #ff1828;
font-style: italic;
font-size: small; }

-->
</style>
</head>
<body>

<form method="post" action="mailto:sabatinas@isp.com">

<table>
<tr>
<td>
</td>
<td>Once you have selected the items that you want to order and the payment type, click the Submit button to process the order.</td>
</tr>
</table>

<fieldset><legend>Customer Information</legend>
<p>Last Name: <input name="lastname" type="text" size="25" />
First Name: <input name="firstname" type="text" size="25" />
<br />Street Address: <input name="address" type="text" size="25" />
<br />City: <input name="city" type="text" size="25" />
State: <input name="state" type="text" size="25" />
ZIP: <input name="zip" type="text" size="10" />
E-mail Address: <input name="email" type="text" size="25" /></p>
</fieldset>

<fieldset><legend>Your Order</legend>
<br />Pizza (select type, size, and ingredients):
<br /><input name="pizzatype" type="radio" value="thin" />Thin
<input name="pizzatype" type="radio" value="thick" />Thick

<br /><input name="size" type="radio" value="medium" />Medium
<input name="size" type="radio" value="large" />Large

<br /><input name="ingredient" type="radio" value="cheese" />Cheese
<input name="ingredient" type="radio" value="sausage" />Sausage
<input name="ingredient" type="radio" value="veggie" />Veggie
-1-
```

Add the following:

<input name="additional" type="checkbox" value="mushrooms" />Mushrooms
<input name="additional" type="checkbox" value="olives" />Olives
<input name="additional" type="checkbox" value="pepperoni" />Pepperoni

<fieldset><legend>Payment Method</legend>

<p>Credit card type:</p>
<select name="payment">
<option>American Express</option>
<option>Visa</option>
<option>MasterCard</option>
</select>

Credit card number:
<input name="cardnum" type="text" size="20" maxlength="20" />

Expiration date:
<input name="cardexp" type="text" size="4" maxlength="4" /></p>

What other items would you like to see on our menu?

<textarea name="other" rows="3" cols="100"></textarea>

<p><input type="submit" value="Submit" />
<input type="reset" value="Reset" /></p>

Thank you for your order.</p>

```

</form>
</body>
</html>

```

-2-
Figure 6-26

BTW**Quick Reference**

For a list of HTML tags and their associated attributes, see Appendix A at the back of this book, or visit the HTML Quick Reference Web page (scsite.com/html6e/qr). For a list of CSS properties and values, see Appendix D at the back of this book, or visit the CSS Quick Reference Web page (scsite.com/html6e/qr).

To Quit Notepad++ and a Browser

- 1** In Notepad++, click the File menu, then Close All.
- 2** Click the Close button on the Notepad++ title bar.
- 3** Click the Close button on the browser title bar. If necessary, click the Close all tabs button.

Chapter Summary

In this chapter, you have learned how to convert a text-based Web page to a Web page form with various controls for user input. The items listed below include all the new HTML skills you have learned in this chapter.

1. Create a Form and Identify the Form Process (HTML 296)
2. Change the Text Message (HTML 297)
3. Add Text Boxes (HTML 298)
4. Add Radio Buttons (HTML 300)
5. Add Check Boxes (HTML 301)
6. Add a Selection Menu (HTML 302)
7. Add a Textarea Box (HTML 305)
8. Add Submit and Reset Buttons (HTML 307)
9. Add Fieldset Controls to Create Form Groupings (HTML 309)

Learn It Online

Test your knowledge of chapter content and key terms.

Instructions: To complete the Learn It Online exercises, start your browser, click the Address bar, and then enter the Web address scsite.com/html6e/learn. When the HTML Learn It Online page is displayed, click the link for the exercise you want to complete and read the instructions.

Chapter Reinforcement TF, MC, and SA

A series of true/false, multiple choice, and short answer questions that test your knowledge of the chapter content.

Flash Cards

An interactive learning environment where you identify chapter key terms associated with displayed definitions.

Practice Test

A series of multiple choice questions that test your knowledge of chapter content and key terms.

Who Wants To Be a Computer Genius?

An interactive game that challenges your knowledge of chapter content in the style of a television quiz show.

Wheel of Terms

An interactive game that challenges your knowledge of chapter key terms in the style of the television show, *Wheel of Fortune*.

Crossword Puzzle Challenge

A crossword puzzle that challenges your knowledge of key terms presented in the chapter.

Apply Your Knowledge

Reinforce the skills and apply the concepts you learned in this chapter.

Creating a Course Evaluation Web Page Form

Instructions: Start Notepad++. Open the file apply6-1.html from the Chapter06\Apply folder of the Data Files for Students. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files. This sample HTML file contains all of the text for the Course Evaluation Survey Web page shown in Figure 6–27. You will add the necessary tags to make the Web page form, as shown in Figure 6–27.

The screenshot shows a Windows Internet Explorer window displaying a course evaluation survey. The title of the page is "Course Evaluation". The page content includes a text box labeled "Course number", five sets of radio buttons grouped by statements, a text area for comments, and buttons for "Submit" and "Reset". Callouts point to each of these elements.

This is an anonymous course survey. Your instructor will not get copies of individual results. Please complete the survey and click the Submit button.

Course number: text box

The content of the course was relevant to my degree.
 Strongly agree Agree Neither agree nor disagree Disagree Strongly disagree

The assignments reinforced the content of the course.
 Strongly agree Agree Neither agree nor disagree Disagree Strongly disagree

My instructor was well organized.
 Strongly agree Agree Neither agree nor disagree Disagree Strongly disagree

It was easy for me to contact my instructor.
 Strongly agree Agree Neither agree nor disagree Disagree Strongly disagree

This course was among the best I have taken.
 Strongly agree Agree Neither agree nor disagree Disagree Strongly disagree

What other comments do you have about this course?
 textarea

Submit and Reset buttons

Thank you for your input.
Office of Academic Affairs

Figure 6–27

Perform the following tasks:

1. Using Notepad++, add the HTML code necessary to make the Web page look similar to the one shown in Figure 6–27. Controls used in the form include:
 - a. One text box for Course number
 - b. Five sets of radio buttons for course evaluation options
 - c. A textarea box with 3 rows and 80 columns
2. Add the HTML code to add Submit and Reset buttons.
3. Use the post method to send an e-mail to email@isp.com.
4. Save the revised document as apply6-1solution.html.
5. Validate your HTML code and test all controls.
6. Print the Web page and HTML.
7. Submit the solution in the format specified by your instructor.

Extend Your Knowledge

Extend the skills you learned in this chapter and experiment with new skills.

Creating a Web Page Restaurant Questionnaire

Instructions: Start Notepad++. Create the form by adding the form controls and groupings, as shown in the Extend Your Knowledge Web page in Figure 6–28.

The screenshot shows a Windows Internet Explorer window displaying a form titled "Jose's Fine Dining™ Questionnaire". The form includes the following elements:

- Header:** "Jose's Fine Dining™ Questionnaire". A callout box labeled "special character for trademark symbol" points to the TM symbol.
- Text:** "In order to better serve our customers, we are looking for new restaurant opportunities. Please take a moment to let us know about your food preferences. We will then e-mail you a coupon worth \$10 off of your next meal as a thank you."
- Section Header:** "Dining Preferences". A callout box labeled "legend style changed" points to the red color used for this section.
- Text:** "How often do you currently dine out each week? Once Twice Three or more times". A callout box labeled "radio buttons and check boxes" points to the radio buttons.
- Text:** "Which type(s) of food do you like to eat?" followed by a list of five check boxes: Italian, Mexican, German, French, and Eastern European. A callout box labeled "radio buttons and check boxes" also points to these.
- Text:** "What determines your restaurant selection? ". A callout box labeled "selection control" points to the dropdown arrow of the button.
- Text:** "Please make other comments below." followed by a with three rows and 60 columns. A callout box labeled "textarea" points to this input field.
- Text:** "Additional Information". A callout box labeled "legend style changed" points to the red color used for this section.
- Buttons:** "Submit" and "Reset".
- Message:** "Thank you for your input."
- Status Bar:** "Done", "Computer | Protected Mode: Off", and "100%".

Figure 6–28

Perform the following steps:

1. Using Notepad++, create the Web page, adding any HTML code necessary to make the Web page look similar to the one shown in Figure 6–28. The code should include:
 - a. An h2 heading that includes a special character (*Hint:* See Appendix F.)
 - b. Three radio buttons for the number of times the visitor eats out
 - c. Five check boxes for types of food the visitor eats
 - d. A selection box with three options for factors that determine where they eat: Price, Proximity to home, Quality of food
 - e. A textarea box with three rows and 60 columns
2. Add the HTML code necessary to add two groupings with the legends as shown. Use an embedded style sheet with classes or inline styles to change the legends to red and italics. Align the first legend to the left and the second to the right.

3. Add the HTML code necessary to add Submit and Reset buttons.
4. Save the revised file using the file name apply6-1solution.html.
5. Validate your HTML code and test all controls.
6. Print the revised HTML file.
7. View the Web page in your browser.
8. Print the Web page.
9. Submit the files in the format specified by your instructor.

Make It Right

Analyze a document and correct all errors and/or improve the design.

Correcting the Tennis Courts Survey Web Page

Instructions: Start Notepad++. Open the file makeitright6-1.html from the Chapter06\MakeItRight folder of the Data Files for Students. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files. The Web page is a modified version of what you see in Figure 6–29. Make the necessary corrections to the Web page to make it look like the figure. Use the image tennis.png at the top of the Web page to the left of the h2 heading. Save the file as makeitright6-1solution.html.

The screenshot shows a Windows Internet Explorer window with the title "Make it Right 6 - Windows Internet Explorer". The address bar displays the path "G:\Chapter06\MakeItRight\makeitright6-1solution.html". The main content area contains the following elements:

- A large image of a tennis ball on the left.
- Tennis Courts Unlimited
Annual Customer
Survey**
- Please enter your name:** (labeled "text boxes")
- E-mail address:** (labeled "text boxes")
- Do you like to play tennis?** Yes No (labeled "radio buttons")
- At which court do you like to play?**
 - City courts
 - Tennis Masters
 - Courts on Court Street
 - At the local high school
(labeled "radio buttons and check boxes")
- In what month do you play the most tennis?** (labeled "selection control")
- What tennis training programs would you be interested in attending?** (labeled "textarea")
- Submit** **Reset** (labeled "Submit and Reset buttons")
- Thank you for your input.**

At the bottom of the window, the status bar shows "Computer | Protected Mode: Off" and "100%".

Figure 6–29

In the Lab

Lab 1: Creating a School Bookstore Survey

Problem: The staff of the school bookstore wants to survey students about their book-buying habits to determine where they purchase their books. The staff has asked you to create a Web page form that contains the questions shown in Figure 6–30.

The screenshot shows a Windows Internet Explorer window titled "Lab 6-1 - Windows Internet Explorer". The address bar displays "G:\Chapter06\IntheLab\Lab6-1solution.html". The main content area is titled "Bookstore Survey". It contains the following elements:

- A group of five text input fields (First name, Last name, Address, City, State) enclosed in a bracket, labeled "text boxes".
- A horizontal line labeled "horizontal rule to divide page".
- A set of two radio buttons ("Yes" and "No") labeled "radio buttons".
- A dropdown menu labeled "selection control".
- A second set of two radio buttons ("Yes" and "No") labeled "radio buttons".
- A large text area labeled "textarea".
- At the bottom are "Submit" and "Reset" buttons.

Figure 6–30

Instructions: Perform the following steps:

1. Using Notepad++, create a new HTML file with the title Lab 6-1 in the main heading section. Add the Web page heading Bookstore Survey at the top of the page.
2. Create a form and identify the form process using the post method with the action attribute set to mailto:email@isp.com.
3. Add seven text boxes for first name, last name, home or school address, city, state, ZIP, plus e-mail address.
4. Add two radio buttons for users to say whether or not they use the campus bookstore.
5. Add a selection menu with three options of your choosing (or use Arnie's Books, Lafollet Shops, and University Bookstore) for users to select the bookstore in which they shop, as shown in Figure 6–30.
6. Create a second set of radio buttons for users to say whether they have purchased books on the Internet, as shown in Figure 6–30.
7. Create a textarea for additional comments and set it to 6 rows and 100 columns.
8. Add Submit and Reset buttons at the bottom of the Web page form.

9. Save the HTML file in the Chapter06\IntheLab folder using the file name lab6-1solution.html.
- Validate the Web page. Print the HTML file.
10. Open the lab6-1solution.html file in your browser and test all controls except the Submit button.
11. Print the Web page.
12. Submit the files in the format specified by your instructor.

In the Lab

Lab 2: Recording Studio Survey

Problem: Abuizam Recording Studios is looking for information on their customers' musical tastes. They want to know what type of music and radio stations they listen to. The company has asked you to create the survey as a Web page form, as shown in Figure 6–31.

The screenshot shows a Microsoft Internet Explorer window with the following details:

- Title Bar:** Lab6-2 - Windows Internet Explorer, G:\Chapter06\IntheLab\lab6-2solution.html
- Content Area:**
 - Image:** A musical note graphic.
 - Title:** Abuizam Recording Studios® Survey
 - Text:** Please complete the survey and click Submit when finished
 - Form Fields:**
 - Your name: [Text Box]
 - Your e-mail address: [Text Box]
 - Do you frequently listen to music? Yes No [Radio Buttons]
 - What type(s) do you prefer?
 Jazz Rock Easy Listening Country Opera Ethnic [Check Boxes]
 - To what radio station(s) do you listen?

 [Selection Control]
 - Why do you tune in to those stations? [Text Area]
 - Buttons:** Submit, Reset
 - Message:** Thank you for your input.

Figure 6–31

Instructions: Perform the following steps:

1. Using Notepad++, create a new HTML file with the title, Lab 6-2, in the main heading section.
2. Create a form and identify the form process using the post method with the action attribute set to mailto your e-mail address (if you do not have an e-mail address, use email@isp.com).
3. Add two text boxes for name and e-mail address, as shown in Figure 6–31.
4. Add a set of radio buttons and six check boxes for users to select their musical preferences.
5. Add a selection menu that initially displays four rows and allows multiple input. One of the menu options should be selected at startup. Use local radio stations' call letters and numbers as your options.

Continued >

In the Lab *continued*

6. Insert a 2-row, 60-column textarea for users to provide additional suggestions.
7. Add a Submit button and a Reset button at the bottom of the Web page form.
8. Add the music.png graphic and a registered trademark symbol in the heading, as shown in Figure 6–31 on the previous page.
9. Save the HTML file in the Chapter06\IntheLab folder using the file name lab6-2solution.html. Validate the Web page. Print the HTML file.
10. Open the lab6-2solution.html file in your browser and test all controls. Test the Submit button only if you used your own e-mail address as the value for the form action attribute.
11. Print the Web page.
12. Submit the files in the format specified by your instructor.

In the Lab

Lab 3: Using Fieldset Controls to Organize a Form

Problem: Your manager at Horizon Learning has asked you to create a Web page form that novice HTML developers can use as a model for a well-designed, user-friendly form. Having created forms for several different Web sites, you have learned that using fieldset controls to group form controls results in a well-organized, easily readable form. Create a Web page form that utilizes three fieldset controls, like the one shown in Figure 6–32.

The screenshot shows a Windows Internet Explorer window with the title "Lab6-3 - Windows Internet Explorer". The URL in the address bar is "G:\Chapter06\IntheLab\Lab6-3solution.html". The main content of the page is titled "Nested Groupings and Controls Within a Table". The form contains several input fields and controls:

- A radio button group labeled "Your question here:" with "Choice 1" and "Choice 2". "Choice 2" is selected.
- A checkbox group labeled "Your question here:" with "Choice 1", "Choice 2", "Choice 3", and "Choice 4". "Choice 4" is checked.
- A dropdown menu labeled "Your question here:" with items "Choice 1", "Choice 2", and "Choice 3". "Choice 1" is selected.
- A large text area labeled "Your question here:" with the placeholder text "Your question here...".
- At the bottom are two buttons: "Submit the Form" and "Reset the Form".

Annotations with callouts highlight specific elements:

- "radio button with choice selected" points to the radio button for "Choice 2".
- "check box with choice selected" points to the checked checkbox for "Choice 4".
- "selection control with three items in startup list" points to the dropdown menu with items "Choice 1", "Choice 2", and "Choice 3".
- "textarea" points to the large text area.
- A large callout on the right indicates "two fieldset groupings nested in Overall grouping".

Figure 6–32



Instructions: Perform the following steps:

1. Using Notepad++, create a new HTML file with the title Lab 6-3 in the main heading section.
2. Add the Web page heading Nested Groupings and Controls Within a Table.
3. Create a form and identify the form process using the post method with the action attribute set to mailto your e-mail address (if you do not have an e-mail address, use email@isp.com).
4. Add two text boxes for name and e-mail address.
5. Add two radio buttons, with Choice 2 preselected, as shown in Figure 6–32, together with four check boxes with Choice 4 selected.
6. Add a 5-row, 35-column textarea, as shown in Figure 6–32.
7. Insert a selection menu with options of Choice 1 through Choice 4. Set the selection menu to display three rows and have Choice 1 preselected as the default option.
8. Add a Submit button that says Submit the Form and a Reset button that says Reset the Form at the bottom of the Web page form.
9. Add three fieldset controls to group the other form controls, as shown in Figure 6–32. Nest the two subgroupings within the main grouping.
10. Save the HTML file in the Chapter06\IntheLab folder using the file name lab6-3solution.html. Validate the Web page. Print the HTML file.
11. Open the lab6-3solution.html file in your browser and test all controls. Test the Submit button only if you used your own e-mail address as the value for the form action attribute.
12. Print the Web page.
13. Submit the files in the format specified by your instructor.

Cases and Places

Apply your creative thinking and problem-solving skills to design and implement a solution.

• EASIER •• MORE DIFFICULT

• 1: Creating a Travel Form

The marketing director at Getaway Travel asked you to create a Web page form to allow customers to request information on the four travel packages offered by the agency: ski & snow, surf & sun, golf & spa, and adventure. Using the techniques learned in this chapter, create a Web page form with input controls to allow customers to request information on one or more travel packages. By default, have all of the travel packages selected on the form. In addition, include input controls for customers to provide a mailing address, an e-mail address, and any suggestions for new travel packages. Include Submit and Reset buttons and use your e-mail address in the action attribute for the form. After creating the Web page, enter information and submit the form. Print the data file with the information and indicate which name-value pairs are related to which controls on the form.

• 2: Changing a Paper Form to an Online Form

As part of your Web development project, your instructor has asked you to find a text-based form that is currently in use by your school, a club, or another organization. Convert this text-based form to a Web page form. Start by designing the form on paper, taking into consideration the fields that are the most appropriate to use for each input area. Once your design is complete, use HTML to develop the Web page form. Test the form, and once testing is done, show the form to several people from the organization that controls the form. Explain to them why it is better to collect information using a Web page form, rather than a printed, text-based form.

Cases and Places *continued***•• 3: Collecting Information with a Form**

The owners of Sabatina's Pizza were very excited about their new online order form. There are a few limitations to ordering on the form, however. As one example, the appetizers and salads (see Chapter 4) are not listed on the form, so no one can order them online. Also, the way the form is set up now, a customer can only order one of each item. A customer may want to order two or more of the same type of pizza. Analyze these requirements and begin a design that allows customers to order appetizers and salads and more than one of each item. Also, think of other ways the form could be improved. (*Hint:* Review other pizza restaurant order forms online.) Put all of your design ideas together and create a paper design. After showing this design to your instructor, add the new controls to the form you created in this chapter.

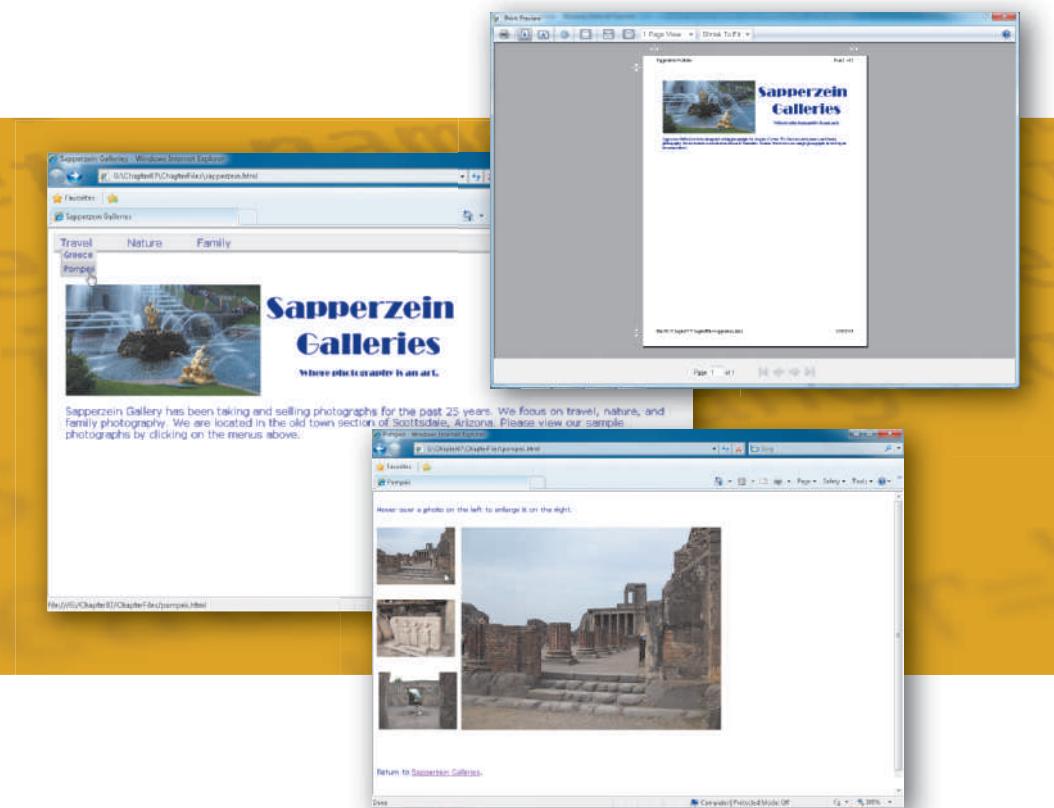
•• 4: Making a Form Easier to Use**Make It Personal**

Your uncle's car club wants to collect information from club members. Search the Internet for two or three examples of Web page forms used to collect information from club members. Print the forms as examples. If you were the Web developer for these Web sites, how would you update the forms to gather more information or make the forms easier to use? Using the example Web pages that you have found, draw a sketch of a Web page form design for a car club. Develop the Web page form as an example to share with your uncle.

•• 5: Creating a Travel Journal**Working Together**

Your team works in the Web development department for a small company in your community. You are interested in learning the latest programming techniques so you can stay current with the technology. In this chapter, data from a form was sent in a file to an e-mail address. The chapter mentioned CGI scripts and the PHP and Perl programming languages as better, more secure methods to use for processing the information submitted in a form. While CGI scripts and Perl programming are beyond the scope of this book, they are important topics to study. Search the Web to find additional information about CGI scripts, PHP, and Perl used in conjunction with forms. Try to find online tutorials that explain how to use these techniques. What other options are available for collecting information online? Develop a Web page that lists links to various Web sites that discuss these topics. Under each link, write a brief paragraph explaining the purpose of each Web site and why it is important to review.

7 Using Advanced Cascading Style Sheets



Objectives

You will have mastered the material in this chapter when you can:

- Add an embedded style sheet to a Web page
- Change the body and link styles using an embedded style sheet
- Create a drop-down menu bar using an embedded style sheet
- Change the color and font styles of the drop-down menus
- Create an external style sheet
- Change the paragraph margins and font styles using an external style sheet
- Create a hover pop-up using an external style sheet
- Use classes, pseudoclasses, and divisions for the pop-up function
- Use the `<link>` tag to insert a link to an external style sheet
- Add an external style sheet for printing Web pages

7 | Using Advanced Cascading Style Sheets

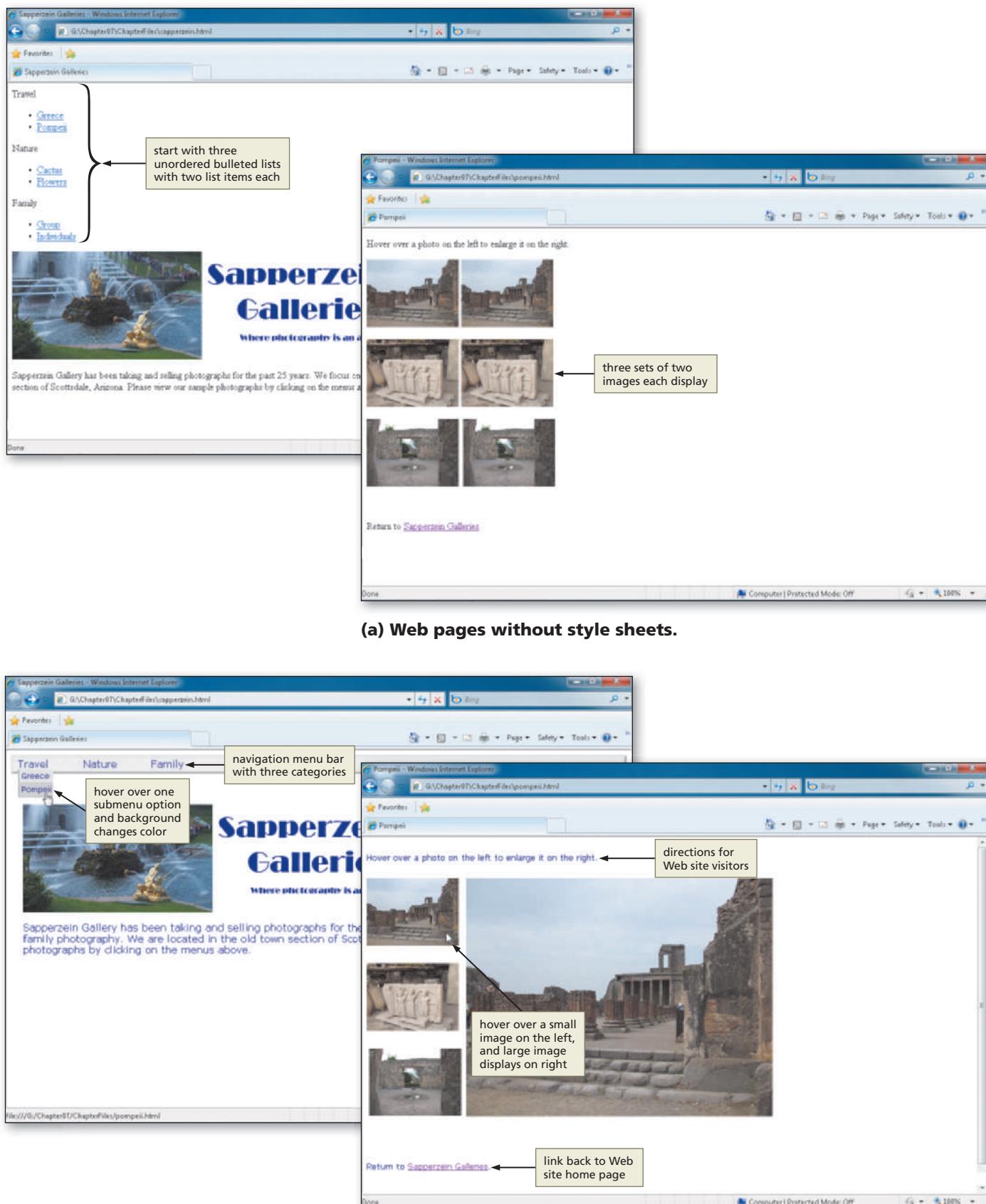
Introduction

In previous chapters, you used HTML tags and Cascading Style Sheets (CSS) to change the way a Web page appears in a Web browser, such as adding italic, bold, colors, headings, and tables. This is also known as the style of the Web page. In this chapter, you expand your knowledge of CSS to give added functionality to your Web pages. You insert drop-down menus for your Web site visitors to use for navigation. This advanced technique is done with more complex CSS code. You also add pop-up image effects using an advanced CSS technique. This method shows you a better way to format your Web pages as opposed to using tables. You have more flexibility using CSS versus tables for Web page structure.

Project — Using Advanced Cascading Style Sheets

Sapperzein Galleries had a Web site created several years ago. Although the Web site is well-designed and effective, they now want to improve their Web site with drop-down menus and image effects. They hire you to enhance their Web site using advanced Cascading Style Sheets (CSS).

At Mr. Sapperzein's request, you use advanced Cascading Style Sheets to add a drop-down menu structure, as shown in Figure 7–1b. Recognizing that the Sapperzein Galleries Web site will continue to grow, you suggest that you modify the Web site to use embedded and external Cascading Style Sheets (CSS). You explain to him that Cascading Style Sheets maintain a consistent look across a Web site — especially Web sites that contain many pages, and can give the pages a more polished look. You suggest that you create an external style sheet that is linked to the other Web pages. This style sheet, which can be easily linked into all pages in the Web site, is used to give the images on the Web page a pop-up effect (Figure 7–1b). Additionally, you would like to create a second external style sheet that can be used to print only the content of a Web page and not the menu bar. Mr. Sapperzein is supportive of the plan and encourages you to start as soon as possible.



(b) Web pages with style sheets.

Figure 7-1

Overview

As you read this chapter, you will learn how to create the Web page shown in Figure 7–1 on the previous page by performing these general tasks:

- Plan the CSS structure.
- Enter HTML code into the Notepad++ window.
- Save the file as an HTML file.
- Enter basic HTML tags and add text to the file.
- Use the `<style>` tag in an embedded style sheet.
- Create external CSS files that are linked into Web pages with a `<link />` tag.
- View the Web pages and HTML code in your browser.
- Validate the Web pages.
- Test and print the Web pages.

Plan Ahead

General Project Guidelines

As you create Web pages, such as the project shown in Figure 7–1 on page HTML 327, you should follow these general guidelines:

1. **Plan the Web site.** First, you should determine if using Cascading Style Sheets (CSS) is appropriate for your Web site. If you have several Web pages and need a consistent style that can be easily updated, CSS is a good choice. If you have a single page with mostly static content and formatting, CSS might not be needed.
2. **Analyze the need.** In the analysis phase of the Web Development Life Cycle, you should analyze what content to include in the Web page. Chapter 7 introduces advanced CSS techniques that can be used for Web development. Using style sheets can eliminate the need to edit multiple Web pages for simple changes. An external style sheet can be edited to make changes across a Web site. Part of the analysis phase includes determining how the multiple Web pages work together using CSS. In this chapter, you create both embedded and external style sheets.
3. **Choose the content for the Web page.** With a multiple-page Web site, you can distribute the content as needed throughout the Web site.
4. **Determine the type of style sheets to use for the pages and their precedence.** If you determine that CSS is appropriate, then you must decide which type(s) of style sheet is best. For Web sites with many Web pages that have a common look, an external style sheet may be the best option. For Web sites with few common looks to the pages, using embedded or inline style sheets may be a better option. Also, knowing style sheet precedence helps you to understand how each style interacts with the others.
5. **Create the style sheets.** Once the analysis and design is complete, the Web developer creates the Web page using CSS. Good Web development standard practices should be followed in this step. Embedded and inline style sheets are used within particular Web pages. An external style sheet must first be created and saved as a .css file. Then a link statement must be inserted into all Web pages in which you want to use the external style sheet.

(continued)

6. **Test all Web pages within the Web site.** An important part of Web development is testing to assure that you are following XHTML standards. In this book, you use the World Wide Web Consortium (W3C) validator that allows you to test your Web page and clearly explains any errors you have. When testing, you should check all content for accuracy. Finally, all of the Web pages with style sheets (external as well as embedded) should be validated as per the standard set throughout this book.

When necessary, more specific details concerning the above guidelines are presented at appropriate points in the chapter. The chapter also will identify the actions performed and decisions made regarding these guidelines during the creation of the Web page shown in Figure 7–1 on page HTML 327.

Plan Ahead

Using Style Sheets

As you learned in earlier chapters, although HTML allows Web developers to make changes to the structure, design, and content of a Web page, HTML is limited in its ability to define the appearance, or style, across one or more Web pages. As a result, Cascading Style Sheets (CSS) were created. With CSS, you can establish a standard look for all Web pages in a Web site. Using CSS, you avoid the tedious steps of adding repetitive codes to format the same types of information. For example, instead of making all paragraphs of text 10pt Verdana in individual `<p>` tags, you can define that style in an external style sheet (.css file) and link that external file to all Web pages. CSS is also perfect for formatting Web pages with tabular material, but without using HTML table tags.

A style is a rule that defines the appearance of an element on a Web page, and a style sheet is a series of rules that defines the style for a Web page or an entire Web site. There are three types of CSS: inline, embedded, and external. In previous chapter projects, you have used all three types to alter the appearance of a Web page or pages by changing characteristics such as font family, font size, margins, and link specifications. In this chapter, you learn to use more advanced CSS features to add functionality to Web pages.

First, an embedded style sheet is used to add a drop-down menu to the home page of the Web site (Figure 7–2a). You use an embedded style sheet in this case because the menu appears only on the home page. An external style sheet (Figure 7–2b) is created for printing and is linked into the home page, `sapperzein.html`. With these style sheets added, the Sapperzein Galleries home page is more attractive, polished, and professional looking (Figure 7–2c). An external style sheet is then created to add a pop-up function in the other Web pages in the Web site (Figure 7–2d). You use an external style sheet because the style will be the same across several Web pages. That external style sheet is linked into the other Greece and Pompeii pages to give the pop-up effect shown in Figure 7–2e.

BTW

CSS

The World Wide Web Consortium (W3C) has a wealth of information about Cascading Style Sheets (CSS). You can find out what is new with CSS, access CSS testing suites, and find links to CSS authoring tools from this Web site. For more information, visit the W3C Web site and search for CSS.

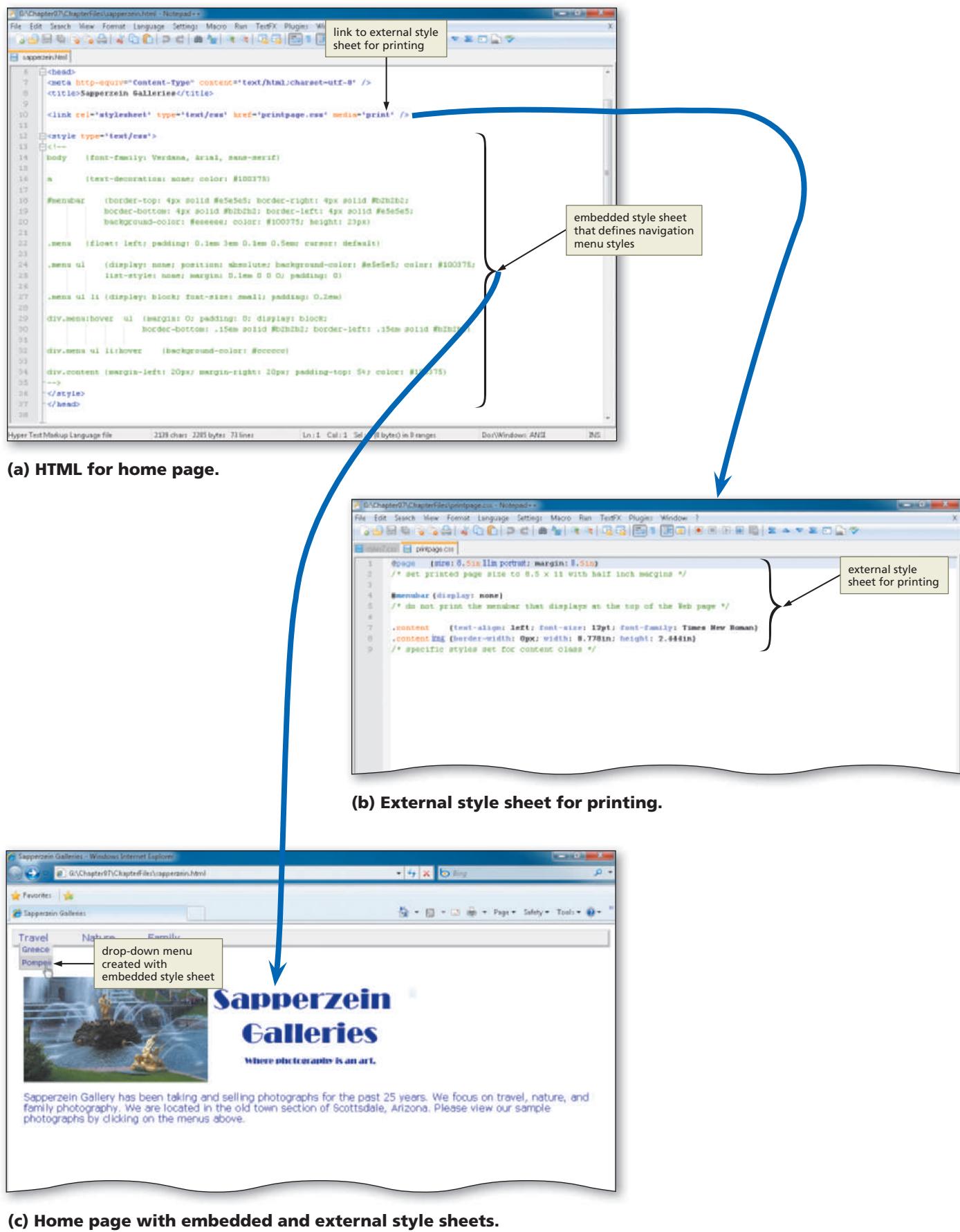
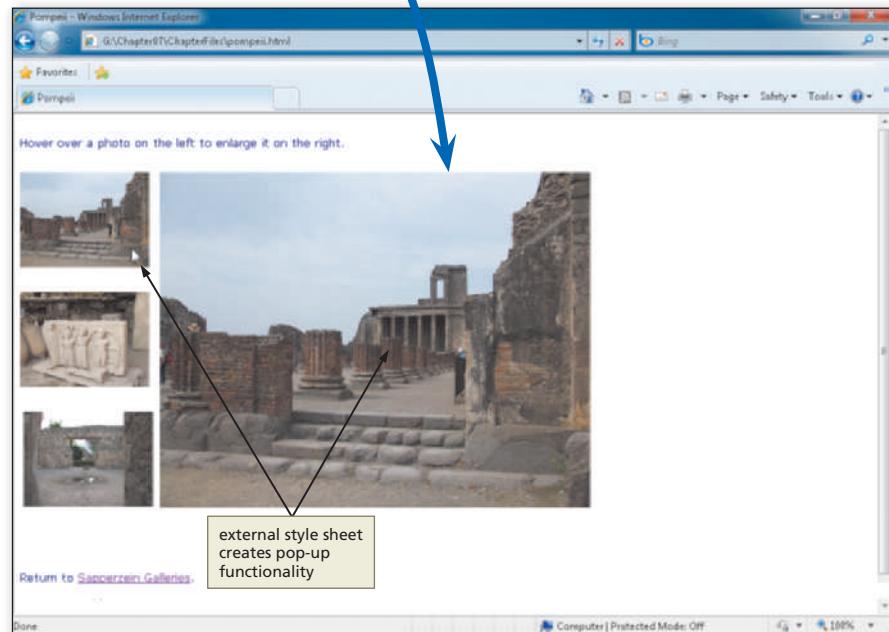


Figure 7-2

```

1  img,small {border: none; text-decoration: none;
2   /* displays no border around images on the left */
3
4  div#large>img.large {height: 0; width: 0; border-width: 0}
5   /* hide the larger image by setting its height and width to zero */
6
7  div#large>img.large {position: absolute; top: 60px; left: 170px; height: 309px;
8   width: 500px; border:none}
9   /* make the larger image appear on the right when a user hovers over one of the smaller images
10  on the left side of the Web page */
11
12 p {margin-bottom: 26px; margin-top: 26px; font-size: small;
13  font-family: Verdana, Arial, sans-serif; color: #100375}
14   /* add bottom and top margins to p elements so they are somewhat aligned with the large image */
15

```

(d) External style sheet for pop-up function.**(e) Secondary Web page with link to external style sheet.****Figure 7-2 (continued)**

Style Sheet Precedence Review

As discussed in previous chapters, each style sheet type has a different level of precedence or priority in relationship to the others. Table 7–1 reviews style sheet precedence.

Table 7–1 Style Sheet Precedence

Type	Level and Precedence
Inline	<ul style="list-style-type: none"> To change the style within an individual HTML tag Overrides embedded and external style sheets
Embedded	<ul style="list-style-type: none"> To change the style of one Web page Overrides external style sheets
External	<ul style="list-style-type: none"> To change the style of multiple pages in a Web site

Plan Ahead

Identify what style sheets to use.

The first step to consider when using style sheets is to lay out a plan that takes style sheet precedence into account. This project uses only embedded and external style sheets.

- Use external style sheets for styles that you want across the Web site.** As mentioned, the greatest benefit of CSS is the ability to identify a style across a Web site. For Web pages in which you want a common look, use external style sheets.
- Use embedded style sheets for single Web page styles.** This type of style sheet is good to use if you want the style to just affect one (or a few) Web pages, and not all pages across the Web site.
- Use inline style sheets for individual styles.** If you want to change the style of one or a few sections of one Web page, then using inline style sheets is the most appropriate. If the style is intended for most (or all) of the Web pages, you may want to switch to embedded or external style sheets.

Adding Style Sheets to the Sapperzein Galleries Site

The Sapperzein Galleries Web site for this chapter consists of nine files, as shown in Table 7–2. The first Web page, sapperzein.html, is the home page of the Web site. The sapperzein.html file contains the navigation menu at the top of the Web page. It also contains the company logo and the home page content.

Table 7–2 Files Used for the Chapter 7 Project

File Name	Purpose and Display Specifics at Startup	Changes Made in Chapter 7
sapperzein.html	<ul style="list-style-type: none"> Home page of the Web site Contains an unordered list of menu options, an image logo, and a paragraph of text 	<ul style="list-style-type: none"> Add an embedded style sheet that creates a drop-down menu
greece.html	<ul style="list-style-type: none"> Contains the text and images that are needed for the pop-up function 	<ul style="list-style-type: none"> Add a link to an external style sheet that displays a large pop-up image when user hovers over the smaller image

Table 7–2 Files Used for the Chapter 7 Project (continued)

File Name	Purpose and Display Specifics at Startup	Changes Made in Chapter 7
pompeii.html	• Contains the text and images that are needed for the pop-up function	• Add a link to an external style sheet that displays a large pop-up image when a user hovers over the smaller image
styles7.css	• Nothing at startup; created from scratch	• Create an external style sheet • Save as a .css file
printpage.css	• Nothing at startup; created from scratch	• Create an external style sheet • Save as a .css file
cactus.html flowers.html group.html individuals.html	• Dummy Web pages that contain initial HTML code and link statement to external style sheet	• Nothing is done to any of these Web pages in the chapter; you can use them as starting points for additional Web pages

In this project, you will add different types of style sheets to the Web pages in the Sapperzein Galleries Web site. To add the style sheets, you will make changes to three of the Web pages stored in the Chapter07/ChapterFiles folder of the Data Files for Students: sapperzein.html, greece.html, and pompeii.html. You also will create two external style sheet files, styles7.css and printpage.css. In addition to the files listed in Table 7–2, all image files needed for the chapter project are stored in the Data Files for Students. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files.

Creating an Embedded Style Sheet.

You would use an embedded style sheet if you want to set the styles within a Web page. In the case of this Web site, you want to create a drop-down navigation menu on the home page only. You design that menu on the home page by creating an embedded style sheet.

- **Determine which Web pages vary enough that an embedded style sheet makes sense.** You may have only one or even just a few Web pages in a Web site that will vary slightly from all other pages. In this case, an embedded style sheet makes sense. If there are styles that are to be repeated in that one Web page (or in a few pages), you would be better off using an embedded style sheet rather than a series of inline style sheets. For instance, if you want all paragraphs of text to have the same style within one Web page, then it makes more sense to embed that style rather than add the style to each paragraph tag within the Web page.
- **Copy an embedded style sheet into other Web pages.** If you have a few Web pages that should have the same style, insert the embedded sheet in one Web page, save, validate, and test it. Once you have verified that it works as you intend, then you can copy/paste the embedded style sheet into the other Web pages.
- **Change to an external style sheet when necessary.** If you find that the style from the embedded style sheet is used on more Web pages as time goes on, you should create an external style sheet and link it into all Web pages in which you had previously inserted an embedded style sheet. For instance, if you decide that you want to have the drop-down menu on all Web pages in the Web site, you should move the code in the embedded style sheet to an external style sheet and link that external style sheet into all Web pages in the Web site.

Plan Ahead

Adding a Navigation Menu with an Embedded Style Sheet

The first step in adding style sheets to the Sapperzein Galleries Web site is to add an embedded style sheet to the home page of the Web site, sapperzein.html. First, the HTML file sapperzein.html must be opened in Notepad++. Then you enter the code for the embedded style sheet. Figure 7–3a shows the home page in the default style (without a style sheet) as provided in the Data Files for Students, and Figure 7–3b shows the same Web page after the embedded style sheet has been added.

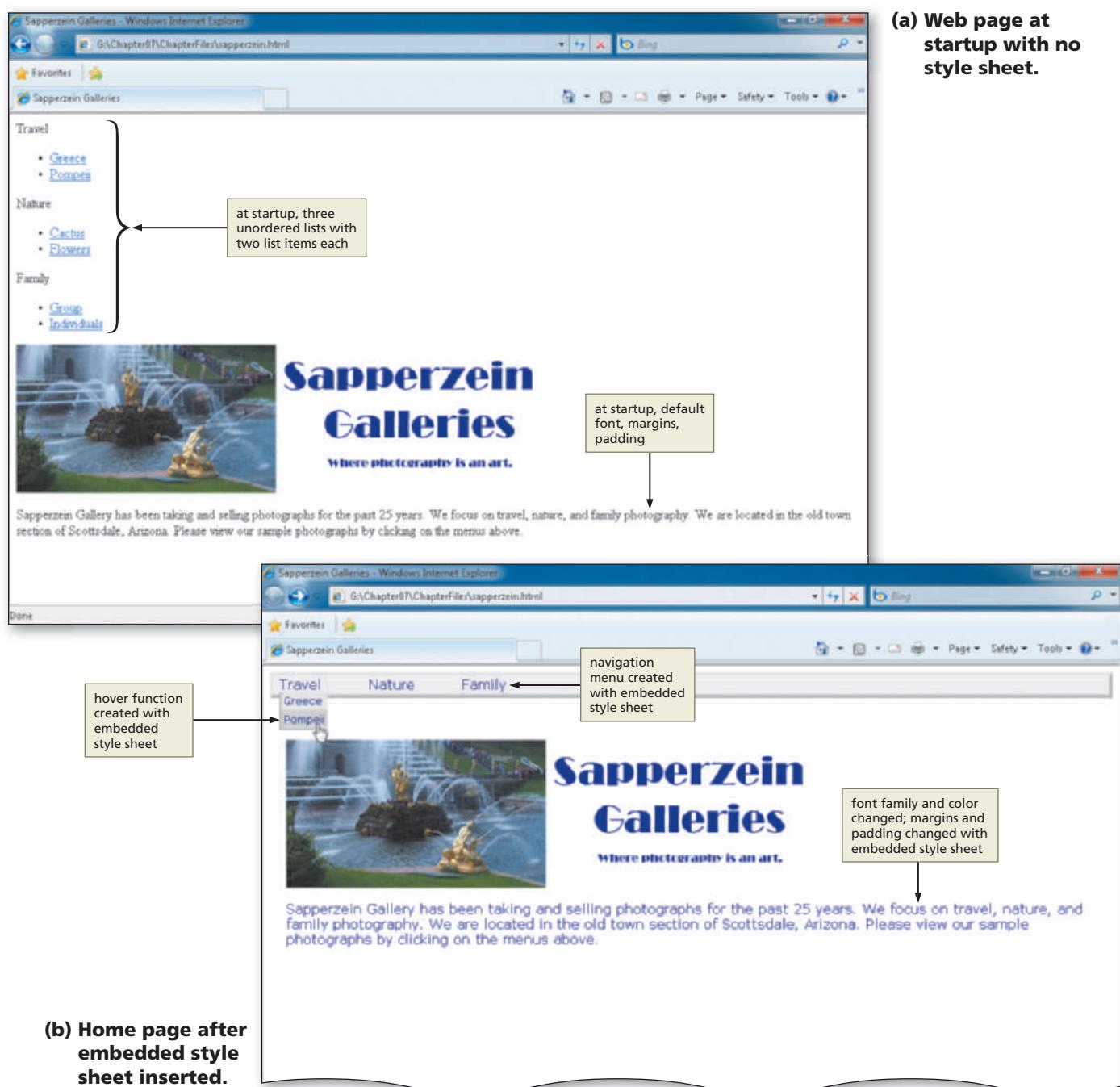


Figure 7-3

To Start Notepad++ and Open an HTML File

- 1** Start Notepad++ and, if necessary, maximize the window.
- 2** With the USB drive plugged into your computer, click File on the menu bar and then click Open.
- 3** If necessary, navigate to the Chapter07\ChapterFiles folder on the USB drive.
- 4** Double-click sapperzein.html in the list of files to open the file shown in Figure 7–4.

The screenshot shows the Notepad++ interface with the file `sapperzein.html` open. The code is as follows:

```

1 <!DOCTYPE html
2   PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
4
5 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
6 <head>
7 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
8 <title>Sapperzein Galleries</title>
9
10 <!--Insert embedded style sheet here --> ← insert embedded
11 </head> style sheet
12 </body> starting on line 10
13
14 <div id="menubar">
15   <div id="menu1" class="menu">Travel ←
16     <ul>
17       <li><a href="greece.html">Greece</a></li>
18       <li><a href="pompeii.html">Pompeii</a></li>
19     </ul>
20   </div>
21
22   <div id="menu2" class="menu">Nature ←
23     <ul>
24       <li><a href="cactus.html">Cactus</a></li>
25       <li><a href="flowers.html">Flowers</a></li>
26     </ul>
27   </div>
28
29   <div id="menu3" class="menu">Family ←
30     <ul>
31       <li><a href="group.html">Group</a></li>
32     </ul>
33   </div>

```

Annotations in the screenshot:

- A callout box points to line 10 with the text: **insert embedded style sheet starting on line 10**.
- A callout box points to the word **Travel** in the first menu with the text: **three words displayed on navigation menu bar**.
- A callout box points to the entire code area with the text: **HTML code in sapperzein.html file at startup**.

Figure 7–4

Setting the Body Style and Link Style, and Adding a Drop-down Menu

The code you will be entering for the embedded style sheet is shown in Figure 7–5 on the next page. Before entering the code, however, you should understand a little more about the styles you are setting.

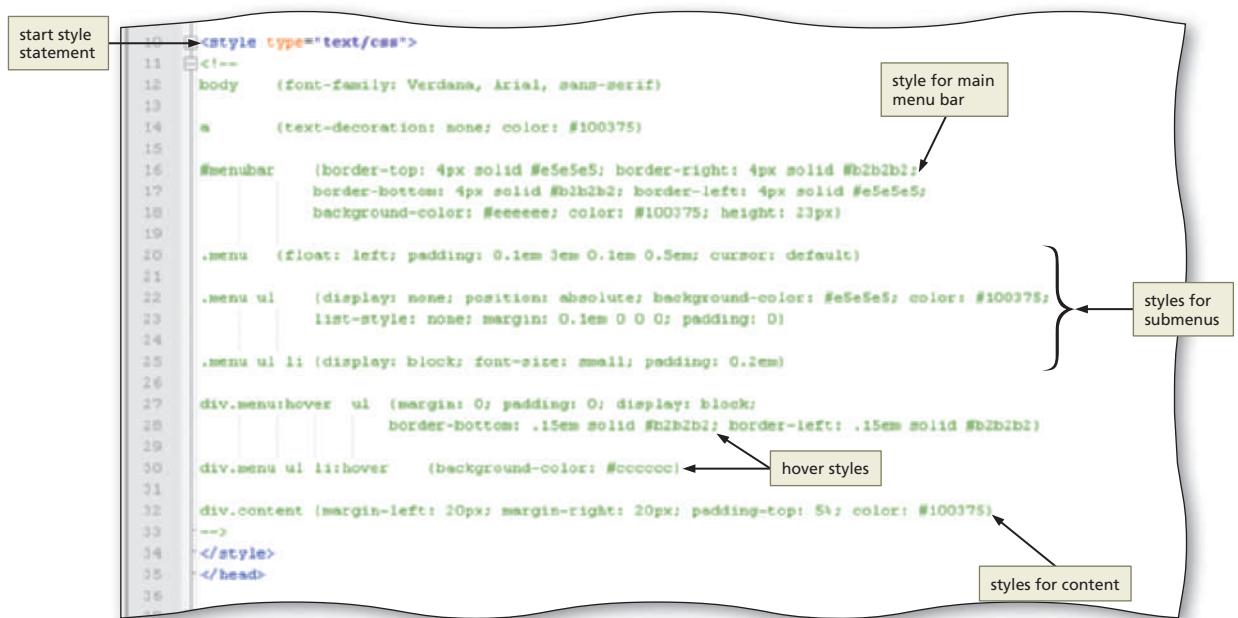


Figure 7-5

BTW**Word Spacing**

The word-spacing property is a good way to add additional space between words. You can use any of the length units including: inches, centimeters, millimeters, points, picas, ems, x-height, and pixels.

The code for an embedded style sheet must be inserted between a start `<style>` tag (line 10) and an end `</style>` tag (line 34), which are positioned within the head element. Within the style tag container, Web developers generally follow the coding practice to add an HTML start comment code `<!--` (line 11) and end comment code `-->` (line 33). The beginning and ending HTML comment lines hide any script language that an older browser cannot interpret.

Let's first take a look at what the CSS code in Figure 7-5 will create on the home page of Sapperzein Galleries. This code will enhance the three unordered lists displayed at the top of the Web page by creating a navigation menu bar (see Figure 7-3b on page HTML 334) with the words Travel, Nature, and Family. It will also improve the appearance of the two list items within each unordered list that are used as submenus. Figure 7-3b on page HTML 334 shows the two list items, Greece and Pompeii. Notice the different background colors on the menu bar and the submenu. This technique utilizes CSS to give the appearance of a menu and submenu system. You could use graphical images such as menu bars here as well. For instance, you could utilize the `menubar.jpg` image that you used in Chapter 5 to give this same menu and submenu effect.

It may help for you to compare the uses of these menu styles (in Figure 7-5) to the section of HTML code (below) provided in the `sapperzein.html` file (Figure 7-4), together with viewing the resulting Web page (Figure 7-3b on page HTML 334). The structure of the divisions and unordered lists used in the `sapperzein.html` file at start-up are as follows (line numbers have been added):

```

16 <div id="menubar">
17     <div id="menu1" class="menu">Travel
18     <ul>
19         <li><a href="greece.html">Greece</a></li>
20         <li><a href="pompeii.html">Pompeii</a></li>
21     </ul>
22 </div>
23

```

```

24     <div id="menu2" class="menu">Nature
25         <ul>
26             <li><a href="cactus.html">Cactus</a></li>
27             <li><a href="flowers.html">Flowers</a></li>
28         </ul>
29     </div>
30
31     <div id="menu3" class="menu">Family
32         <ul>
33             <li><a href="group.html">Group</a></li>
34             <li><a href="individuals.html">Individuals</a></li>
35         </ul>
36     </div>
37 </div>

```

Notice that all three menu bar options (Travel, Nature, Family) are within a division (section) of the Web page, as indicated on lines 16 <div> and 37 </div> above. This division is given the id name menubar. Note in Figure 7–5 that lines 16 through 18 define the styles that are to be used for id="menubar" in the CSS code. There are also three unordered lists within this same division (for Travel, Nature, and Family), each containing two list items. Those unordered lists each use the class="menu" (see lines 20, 22, and 25 in Figure 7–5). Therefore, the styles defined in the CSS code lines 20 through 30 are reflected in the submenu list items from the three unordered lists.

This menu structure is further discussed here per lines of CSS code. Refer to Figure 7–5, lines 11 through 33 of HTML code in the file sapperzein.html, and Figures 7–7a and 7–7b on page HTML 339 as you review this information.

Line 12 in the embedded style sheet (Figure 7–5) sets the font family throughout the Web page by use of the body element. Line 14 sets the link color and turns text-decoration (i.e., underline) off. Lines 16 through 30 set the styles for the drop-down menu bar that will display at the top of the Web page, as shown in Figure 7–7a on page HTML 339. In lines 16 through 18, you set the borders (top, right, bottom, and left) of an element named menubar that will be used in a <div> tag. All of the borders are set to 4px solid style. The top and left borders are set to the same color, while the bottom and right are set to another color. The background color is set to #eeeeee, while the text color is set to #100375. This portion of the statement on line 18:

height: 23px

sets the height of the area to 23 pixels. This is something that you adjust as you develop the Web page. You need to determine how much space you want relative to the default font size.

Line 20 (Figure 7–5) sets the styles for a class named menu. The statement:

.menu {float: left; padding: 0.1em 3em 0.1em 0.5em; cursor: default}

floats the text to the left. The padding property uses the “shorthand property” code. The **shorthand property** allows a Web developer to shorten the code. Instead of using padding-top, padding-bottom, padding-right, and padding-left, you can specify all the padding values in one property as line 20.

BTW

Line Height

The CSS property line-height property gives you the ability to control line height. With this property, you can control the vertical spacing between lines of text. There are three different ways to add the line-height value: by number, by length unit, and by percentage. If you specify by number, the browser uses the font-size property to determine the space. You also can use em and pt to set the height by unit. Finally, you can determine the line spacing by a percentage.

BTW

Em Units

The em is a very useful unit in CSS, because it adapts automatically to the font size that the Web page visitor uses.

BTW

Shorthand Properties

Shorthand properties are great to use and make your CSS code very efficient. You can use shorthand properties with many different elements, including padding, borders, margins, and fonts. See the w3.org Web site for helpful information about shorthand properties.

The padding property can have from one to four values. Table 7–3 shows the shorthand statement together with the resulting values.

Table 7–3 Shorthand Properties

Padding Property Statement	Resulting Values
<code>padding: 25px 50px 75px 100px;</code>	top padding is 25px right padding is 50px bottom padding is 75px left padding is 100px
<code>padding: 25px 50px 75px;</code>	top padding is 25px right and left paddings are 50px bottom padding is 75px
<code>padding: 25px 50px;</code>	top and bottom paddings are 25px right and left paddings are 50px
<code>padding: 25px;</code>	all four paddings are 25px

Line 20 also sets the cursor to the default value using cursor: default as the statement. If you do not have this statement, then you would not see a cursor as shown in Figure 7–6a. With this cursor code statement inserted, you see the cursor as shown in Figure 7–6b. Notice also that you are setting the padding in the statement to an “em” measurement value. Table 7–4 describes the units that can be used by Web developers.

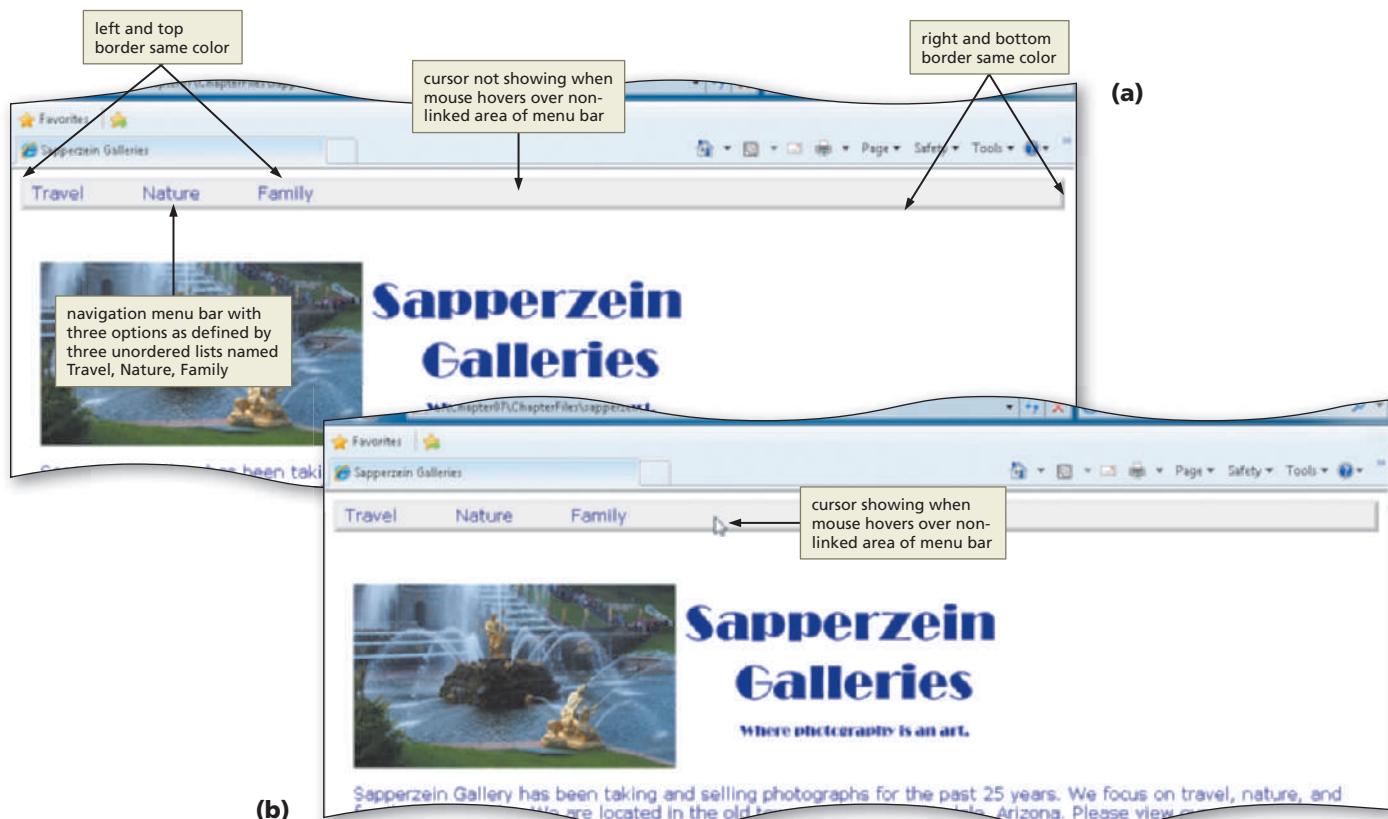
**Figure 7–6**

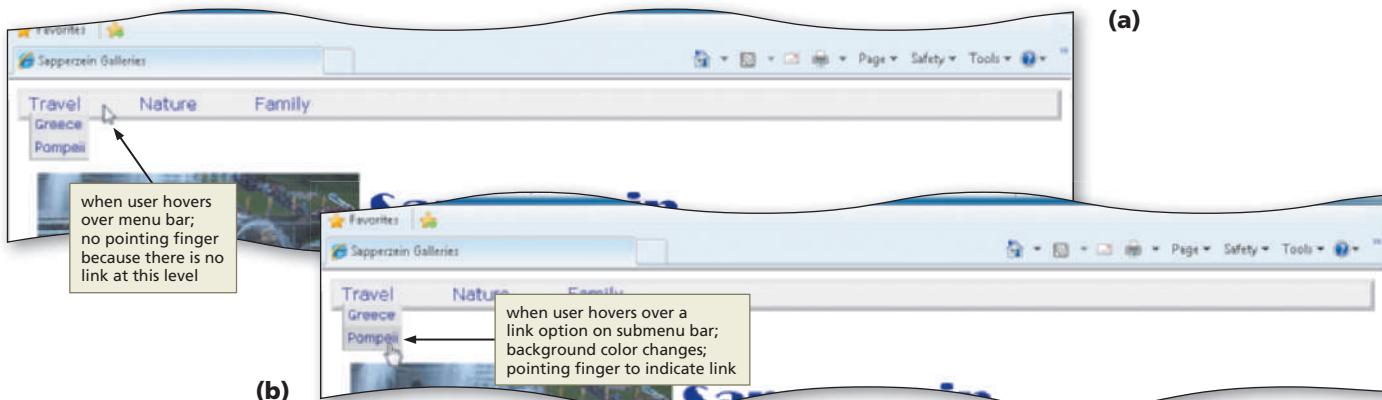
Table 7–4 Measurement Values

Unit	Description
%	percentage
in	inch
cm	centimeter
mm	millimeter
em	1em is equal to the current font size; 2em means 2 times the size of the current font. For example, if an element is displayed with a font of 12 pt, then 2em is 24 pt
ex	one ex is the x-height of a font (x-height is usually about half the font size)
pt	point (1 pt = 1/72 inch)
pc	pica (1 pc = 12 points)
px	pixels (a dot on the computer screen)

The next section of code in the embedded style sheet (lines 22 through 30 in Figure 7–5 on page HTML 336) specifies additional styles for the menu bar and submenus. Figure 7–6b shows the menu bar at startup. Figure 7–7a shows the menu bar when the pointer is on one of the three main menu options (Travel, Nature, Family). This is defined with the style statements on lines 22 and 23 in the CSS code. Figure 7–7b shows the menu bar when a Web site visitor hovers over one of the two list items in each of three unordered lists. This style is defined on lines 27 and 30 where the hover pseudoclass is used.

**CSS Pseudoclasses**

To expand the possibilities of CSS, you can use pseudoclasses (e.g., the link hover). When used effectively, CSS pseudoclasses are used to add special effects to some selectors.

**Figure 7–7**

Next, there are some properties with which you are not yet familiar. Line 22 in the CSS code sets a style for the unordered lists (ul) in the menu class. It starts by setting display to none. That means at Web page startup (i.e., before the user hovers over the menu), the unordered lists will not display (Figure 7–7a). The position property is used to position an element on the Web page. If you do not use the position property, the elements display on the Web page in the order in which they appear. For example, if you have a line of text entered in your HTML code, and then you insert an image in the code, the text appears on the Web page before the image. The position: absolute code sets the style so that the menu text remains constant and does not move. Table 7–5 lists the available property values for position.

Table 7–5 Position Property Values

Property Values	Value Description
absolute	Generates an absolutely positioned element, positioned relative to the first parent element that has a position other than static
fixed	Generates an absolutely positioned element, positioned relative to the browser window
relative	Generates a relatively positioned element, positioned relative to its normal position, so "left: 10;" adds 10 pixels to the element's left position
static	Default; no position; the element occurs in the normal flow
inherit	Specifies that the value of the position property should be inherited from the parent element

BTW

Measurement Values
When is it best to use the "em" measurement versus using the "pt" measurement? There are many advanced Web page design resources available that discuss this topic. The main goal with any measurement value is to design a Web page so that it is legible.

When you add the list-style: none code in line 23, you turn off the display of the disc that is the default bullet for an unordered list. The margin: 0.1em 0 0 0 code is a shorthand property statement for margins. When you set the style display: block on lines 25 and 27, the ul element will generate a block box (line break) before and after the element.

Finally, line 32 sets a new division id name called content. This division is used in the sapperzein.html file that begins with the <div> tag on line 39. This will be the style just for the paragraph of text on the Sapperzein Galleries home page. Again, review this material while looking at Figure 7–5 together with the open sapperzein.html file and Figure 7–7 for a comprehensive view of this embedded style sheet.

To Add an Embedded Style Sheet

Table 7–6 shows the CSS code for the embedded style sheet to be entered directly in the header section of the HTML code for the home page, sapperzein.html.

Table 7–6 Code for an Embedded Style Sheet

Line	CSS Code
10	<style type="text/css">
11	<!--
12	body {font-family: Verdana, Arial, sans-serif}
13	
14	a {text-decoration: none; color: #100375}
15	
16	#menuBar {border-top: 4px solid #e5e5e5; border-right: 4px solid #b2b2b2;
17	border-bottom: 4px solid #b2b2b2; border-left: 4px solid #e5e5e5;
18	background-color: #eeeeee; color: #100375; height: 23px}
19	
20	.menu {float: left; padding: 0.1em 3em 0.1em 0.5em; cursor: default}
21	
22	.menu ul {display: none; position: absolute; background-color: #e5e5e5; color: #100375;
23	list-style: none; margin: 0.1em 0 0 0; padding: 0}
24	

Table 7–6 Code for an Embedded Style Sheet (continued)

Line	CSS Code	
25	.menu ul li	{display: block; font-size: small; padding: 0.2em}
26		
27	div.menu:hover ul	{margin: 0; padding: 0; display: block; border-bottom: .15em solid #b2b2b2; border-left: .15em solid #b2b2b2}
28		
29		
30	div.menu ul li:hover	{background-color: #cccccc}
31		
32	div.content	{margin-left: 20px; margin-right: 20px; padding-top: 5%; color: #100375}
33	-->	
34		</style>

The following step illustrates how to add an embedded style sheet to the Web page sapperzein.html.

1

- Highlight the comment `<!--Insert embedded style sheet here -->`, on line 10.
- Enter the CSS code shown in Table 7–6 (Figure 7–8).

Q&A

What is an easy way to find out what fonts are supported on your computer system?

One way is to review the font names and examples as they appear in an application such as in the Font menu in Microsoft Word. You may want to try different fonts and sizes in an application such as Word to see what they look like.

You can save a document as a Web page from Word and view it in the browser as well.

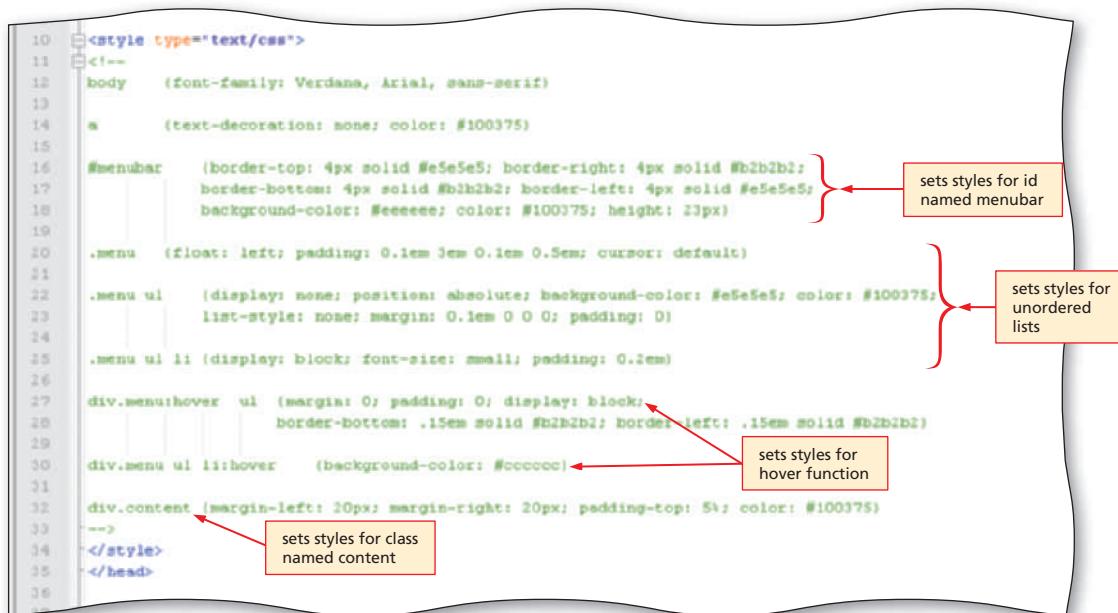
Q&A

Why would I want to use the “hover” technique for links?

It adds a bit of interactivity and in this case helps to highlight the menu and submenu structure for the user.

 **Experiment**

- Remove the `position: absolute` statement in line 22 and see what it does to the words Nature and Family when you hover over the word Travel. Put the statement back in.



```

10 <style type="text/css">
11 <!--
12 body (font-family: Verdana, Arial, sans-serif)
13 a (text-decoration: none; color: #100375)
14
15 #menubar (border-top: 4px solid #e5e5e5; border-right: 4px solid #b2b2b2;
16 border-bottom: 4px solid #b2b2b2; border-left: 4px solid #e5e5e5;
17 background-color: #eeeeee; color: #100375; height: 23px)
18
19 .menu (float: left; padding: 0.1em 0em 0.1em 0.5em; cursor: default)
20
21 .menu ul (display: none; position: absolute; background-color: #e5e5e5; color: #100375;
22 list-style: none; margin: 0.1em 0 0 0; padding: 0)
23
24 .menu ul li (display: block; font-size: small; padding: 0.2em)
25
26 div.menu:hover ul (margin: 0; padding: 0; display: block;
27 border-bottom: .15em solid #b2b2b2; border-left: .15em solid #b2b2b2)
28
29 div.menu ul li:hover (background-color: #cccccc)
30
31 div.content (margin-left: 20px; margin-right: 20px; padding-top: 5%; color: #100375)
32 -->
33 </style>
34 </head>

```

Figure 7–8

BTW

Font Families

You also can specify font-weight using numerical values from 100 to 900. Normal text that is not bold has a value of 400. Each larger number is at least as bold as the one above it, and 900 is the boldest option of the font. The browser determines how bold each value is as it is displaying the Web page.

To Save, Validate, and View an HTML File

After you have added the embedded style sheet to the sapperzein.html Web page, you should save the HTML file, and view the Web page to review the style changes.

- 1** With the USB drive plugged into your computer, click File on the menu bar and then click Save.
- 2** Validate the Web page using the W3C validation service.
- 3** Open the sapperzein.html file in the Web browser to show the completed navigation menu, as shown at the top of the sapperzein.html Web page (Figure 7–9).

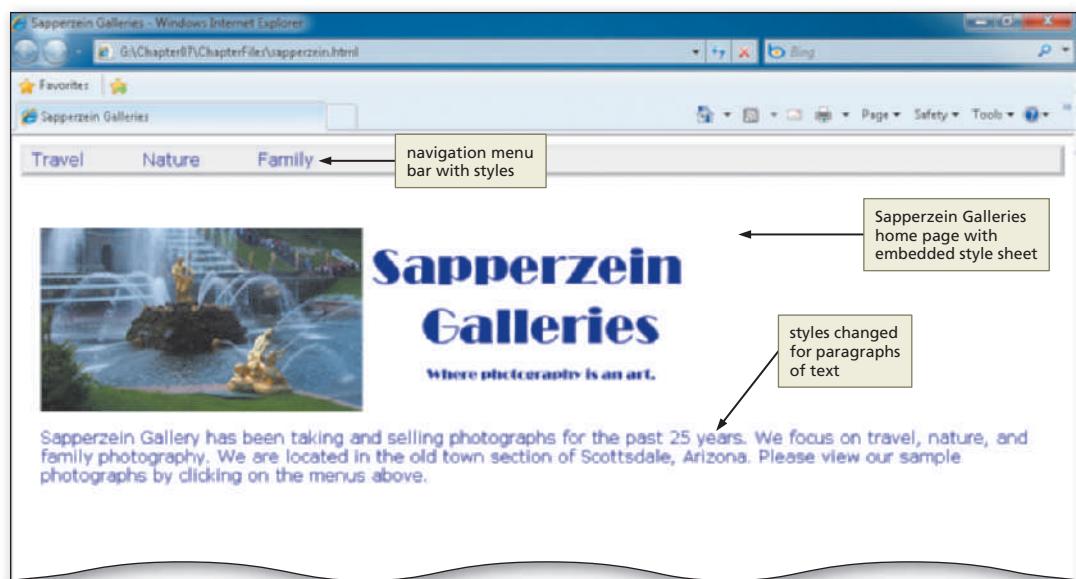


Figure 7–9

Plan Ahead**Creating an External Style Sheet**

The external style sheet is the most powerful and lowest precedence style sheet. With this style sheet, you can easily create a common look across a Web site by creating the external (.css) style sheet and linking it into all other Web pages. In this chapter project, you have two list items per main category (Travel, Nature, Family), or six Web pages that you want to have a common look.

- **Create the external style sheet.** The first step is to create the file itself. This file, which contains all of the style statements that you want, has to be saved with a file name extension of .css. Make sure to store this file in the same folder as the other Web pages.
- **Link the external style sheet into the Web pages.** The second step is to link the external style sheet (.css file) into the Web pages where you want it. The link statement is placed between the <head> and </head> tags.
- **Add comments to your code as needed.** The CSS code added to this external style sheet is complex. Comments help you remember and explain what you have done. The comments will not display on the Web page, but they will stay in the file with the CSS code.

Adding Pop-ups with an External Style Sheet

As you learned in previous chapters, an external style sheet is a separate text file that contains the style statements that define how the Web page elements will appear across multiple pages. After you create the text file with all of the desired style statements and comments, you save the file with the file extension .css to identify it as a CSS file. You then use a <link> tag to link the external style sheet to the Web pages to which you want to apply the style.

Structuring the Web Page

It is useful to understand how you can structure your Web page by dividing it into logical sections. In previous chapters, you used the <div> </div> tags for structure. Specifically, you aligned images by placing the image element within the <div> </div> container. You also set specific styles using the <div> </div> tags. When you use the <div> </div> tags, you are able to design a layout that uses CSS, including inserting images.

When structuring your Web pages, it is also useful to understand the concept of the box model. The **box model** describes the structure of the elements that are displayed on the Web page. Once you have positioned the box on the Web page, you can control its appearance by manipulating its padding, borders, and margins, as shown in Figure 7–10. The **margin** specifies the space between the element and other content on the Web page. The **border** is what surrounds the element content. The **padding** is the space between the content of the element and the box border. These four elements (content, padding, border, and margin) determine how the element content is displayed in the browser.

BTW

The Box Model

The box model is an important concept to understand. Determining an effective structure of your Web pages involves correctly determining the box model controls. Search the Web for the term "box model" to find numerous resources that discuss box model control principles.

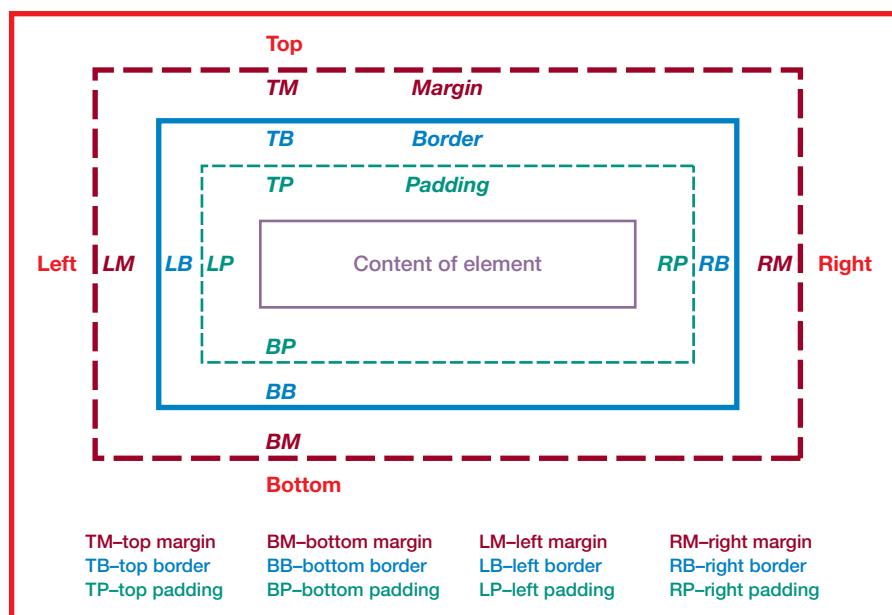


Figure 7–10

You will manipulate some of these characteristics (border, margin-bottom, margin-top) in the external style sheet completed in this section of the chapter. You also use the <div> </div> tags to divide the Web page to allow the pop-up effect.

BTW

Box Model Troubleshooting

You can also find an abundance of information on the Box model troubleshooting tools like those in Firebug or Google Chrome's Developer Tools.

Creating a Pop-up Using Cascading Style Sheets

The CSS code for the external style sheet defines a new style that provides a pop-up capability on a Web page. In this case, you add link and hover functionality that allows a Web site visitor to hover (i.e., move the mouse pointer) over an image and display a larger version of the image.

The two files used for the pop-up function, greece.html and pompeii.html, are stored in the Data Files for Students. These files are all ready to use for the pop-ups but need to have the link statements inserted after the external style sheet is created and saved. It helps to look at the code in one of those files (greece.html) to see how the CSS code you will enter is used in the HTML file. The HTML code from greece.html follows:

```

<div id="sizes">
<p><a>
    
    
</a></p>
<p><a>
    
    
</a></p>
<p><a>
    
    
</a></p>
</div>

```



External Style Sheet Validator

For the external style sheet, be sure to use the CSS validator found at the w3.org validation service at http://jigsaw.w3.org/css-validator/#validate_by_upload.

There are two areas that need to be discussed to understand how the CSS code applies to the HTML code. Table 7–7 contains the CSS code that you will input into an external style sheet named styles7.css. The HTML code above from greece.html shows a division with an id named sizes and three sets of images for the three images, as shown in Figure 7–1b on page HTML 327.

Lines 4 and 7–8 in Table 7–7 define the styles for the id named “sizes” as used in the greece.html code. The `<div>` statement at the start of this code identifies an id named “sizes” at the start of the HTML code. Line 1 in Table 7–7 creates a class named “small” that is used with the image element. Lines 4 and 7 identify a class named “large” to be used as shown in the greece.html code above. Note that line 4 identifies the styles for a link of an image for a class named large (`a img.large`), as used in the division with id sizes. Line 4 also tells the browser to set the large image size to height of zero and width of zero. The net effect of this is that the large image does not display at startup.

Line 7 targets the same elements (`a` and `img`) and the same class (`large`), but this time it sets styles for the hover pseudoclass (`a:hover`). This style statement tells the browser to enlarge the image with class name `large` when a user hovers over the small image.

The “large” version of the image is set to an absolute position 68 pixels from the top of the browser window and 170 pixels from the left of the window. The height of the image when enlarged is 389 pixels, while the width is 500 pixels. This CSS code enlarges an image, but you can also display text or other elements using this same basic code structure.

To Create and Print an External Style Sheet

To create an external style sheet, you open a new text file and enter CSS code shown in Table 7-7. After coding the style statements, you save the file with the file extension .css to identify it as a CSS file.

Table 7-7 Code for an External Style Sheet

Line	CSS Code
1	img.small {border: none; text-decoration: none}
2	/* displays no border around images on the left */
3	
4	div#sizes a img.large {height: 0; width: 0; border-width: 0}
5	/* hide the larger image by setting its height and width to zero */
6	
7	div#sizes a:hover img.large {position: absolute; top: 68px; left: 170px; height: 389px; width: 500px; border:none}
8	
9	/* make the larger image appear in the same space as the frame on the right when a user hovers over
10	one of the smaller images on the left side of the Web page */
11	
12	p {margin-bottom: 26px; margin-top: 26px; font-size: small; font-family: Verdana, Arial, sans-serif; color: #100375}
13	
14	/* add bottom and top margins to p elements so they are somewhat aligned with the large image */

The following step illustrates how to create an external style sheet to define Web page style.

1

- If necessary, click the Notepad++ button on the taskbar. Click File on the menu bar and then click New.

- Enter the CSS code as shown in Table 7-7.

comment lines

- With the USB drive plugged into your computer, click File on the menu bar and then click Save As.

Type styles7.css

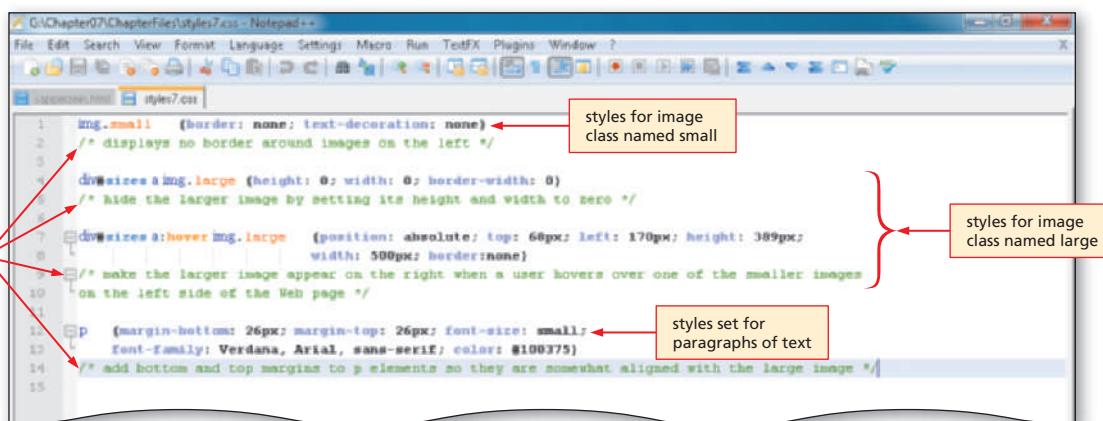


Figure 7-11

in the File name text box. If necessary, navigate to the Chapter07\Chapterfiles folder on your USB drive. Click the Save button in the Save As dialog box to save the file as styles7.css (Figure 7-11).

- Click the File menu, click Print on the File menu, and click the Print button in the Print dialog box.

Linking to an External Style Sheet

You now want to add this pop-up capability to two Web pages in the Sapperzein Galleries Web site: greece.html and pompeii.html. Linking the external style sheet to each of these Web pages gives you the capability to enlarge images in a pop-up format.

To link to the external style sheet, a `<link>` tag must be inserted into each of these two Web pages. The `<link>` tag used to link an external style sheet is added within the `<head>` tag of the Web page HTML. The general format of the `<link>` tag is:

```
<link rel="stylesheet" type="text/css" href="styles7.css" />
```

where `rel="stylesheet"` establishes that the linked document is a style sheet, `type="text/css"` indicates that the CSS language is used in the text file containing the style sheet, and `href="styles7.css"` provides the name and location (URL) of the linked style sheet. To link a style sheet to a Web page, the `<link>` tag must use "stylesheet" as the value for the `rel` property and `text/css` as the value for the `type` property. The URL used as the value for the `href` property varies, based on the name and location of the file used as the external style sheet. The URL used here indicates that the external style sheet, `styles7.css`, is located in the main or root directory of the Web site.

To Link to an External Style Sheet

The following steps illustrate how to add a link to an external style sheet using a `<link>` tag and then save the HTML file.

1

- If necessary, click the Notepad++ button on the taskbar.
- With the USB drive plugged into your computer, click File on the menu bar and then click Open on the File menu.
- If necessary, navigate to the G:\Chapter07\ChapterFiles folder. Click the greece.html file.
- Click the Open button in the Open dialog box.
- Highlight the text, `<!--Insert external style sheet link statement here -->` on line 10.
- Type `<link rel="stylesheet" type="text/css" href="styles7.css" />` to enter the link to the external style sheet (Figure 7-12).

```

<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Greece</title>
<link rel="stylesheet" type="text/css" href="styles7.css" /> ← link statement to external style sheet styles7.css
</head>
<body>
<p>Hover over a photo on the left to enlarge it on the right.</p>
<div id="sizes">
<p><a>

<a href="#" alt="Greece 2" />
<a href="#" alt="Greece 3" />
</a></p>

```

Figure 7-12

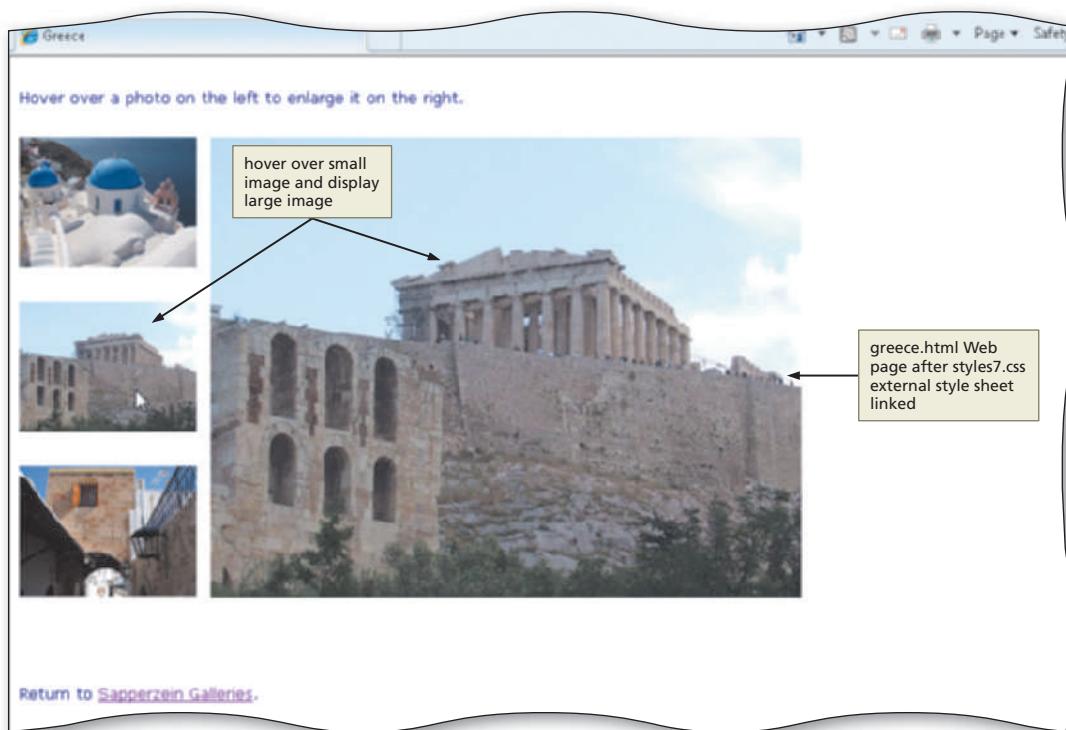
Q&A

Will the styles from the styles7.css take effect for all menus within the Web site?

As long as you insert the style sheet link statement into the Web page, then the menu styles will take effect. Remember that you can override those styles with either an embedded or an inline style sheet. You would do this if there is content that you want to vary from the styles used in the external style sheet.

2

- Click File on the menu bar and then click Save on the File menu.
- Validate the Web page using the W3C service.
- Return to the browser window and use the menu bar to click Travel and then click Greece. Hover over each image shown on the Web page (Figure 7–13) to see the changes in the Web page.

**Figure 7–13**

To Link the Remaining HTML File to an External Style Sheet

You have linked the greece.html page to the external style sheet styles7.css. Now you need to link the pompeii.html Web page to the same style sheet. The following steps show how to add a `<link>` tag to the Pompeii Web page and then save the file.

- 1** If necessary, click the Notepad++ button on the taskbar.
- 2** With the USB drive plugged into your computer, click File on the menu bar and then click Open on the File menu.
- 3** If necessary, navigate to the G:\Chapter07\ChapterFiles folder. Click the pompeii.html file.
- 4** Click the Open button in the Open dialog box.
- 5** Highlight the text, `<!--Insert external style sheet link statement here -->` on line 10.
- 6** Type `<link rel="stylesheet" type="text/css" href="styles7.css" />` to enter the link to the external style sheet.

To Save, Validate, and Test the Web Page

- 1** Click File on the menu bar and then click Save on the File menu.
- 2** Validate the Web page. For the external style sheet, make sure to use the CSS validator found at http://jigsaw.w3.org/css-validator/#validate_by_upload.
- 3** Open pompeii.html in a Web browser and test to see that the pop-up styles are implemented.

Creating an External Style Sheet for Printing

You next create an external style sheet that will be used for printing a Web page. When Web site visitors print a Web page, they generally want to print the content of the Web page, which does not include the navigation menu bar. You can limit what prints by creating an external style sheet that is used specifically to set the styles for a printed Web page.



@page Rule

There are many ways to control the printing of a Web page. Paged media (paper or transparencies) differ from continuous media in that the content of the document is split into one or more discrete pages. Using the @page rule gives you the flexibility that you need.

Understanding the Page Box Model

In this external style sheet, you will add a page box with an @page rule to format the printed page. A **page box** is a rectangular region that contains two areas: the page area and the margin area. The page area includes the elements (or boxes) laid out on the page. The margin area surrounds the page area; the page margin area is transparent. Web developers can specify the margins of a page box inside an @page rule. An **@page rule** contains the keyword @page, followed by an optional page selector, followed by a block of declarations. Page selectors give a Web developer the flexibility to designate the first page, all left pages, or all right pages. For instance, you could specify different margins for the first printed page of a Web site. You could therefore use the first pseudoclass in your @page rule by using the statement:

```
@page :first { }
```

The printing external style sheet does not use any optional page selectors. It does include a declaration in the @page rule with the statement:

```
@page {margin: 0.5in}
```

As with other external style sheets, you must first create the external style sheet, and then you must link that style sheet into the Web page(s) for which you want to use it using the <link /> statement.

To Create and Print an External Style Sheet for Printing

Table 7–8 shows the code for the external style sheet for printing.

Table 7–8 Code for a Print Style Sheet

Line	CSS Code
1	<code>@page {margin: 0.5in}</code>
2	<code>/* set printed page with half-inch margins */</code>
3	
4	<code>#menubar {display: none}</code>
5	<code>/* do not print the menubar that displays at the top of the Web page */</code>
6	
7	<code>.content {text-align: left; font-size: 12pt; font-family: sans-serif}</code>
8	<code>.content img {border-width: 0px; width: 8.778in; height: 2.444in}</code>
9	<code>/* specific styles set for content class */</code>

1

- If necessary, click the Notepad++ button on the taskbar. Click File on the menu bar and then click New.
- Enter the CSS code as shown in Table 7–8.
- With the USB drive plugged into your computer, click File on the menu bar and then click Save As. Type printpage.css in the File name text box. If necessary, navigate to the Chapter07\ChapterFiles folder on your USB drive. Click the Save button in the Save As dialog box to save the file as printpage.css (Figure 7–14).
- Click the File menu, click Print on the File menu, and click the Print button in the Print dialog box to print the CSS code.

```

@page rules setting
style for printed page

File Edit Search View Format Language Settings Macro Run TextFX Plugins Window ?
G:\Chapter07\ChapterFiles\printpage.css - Notepad++
sapperzein.html styles7.css greece.html printpage.css
1 @page {margin: 0.5in}
2 /* set printed page with half inch margins */
3
4 #menubar {display: none}
5 /* do not print the menubar that displays at the top of the Web page */
6
7 .content {text-align: left; font-size: 12pt; font-family: Times New Roman}
8 .content img {border-width: 0px; width: 8.778in; height: 2.444in}
9 /* specific styles set for content class */

```

styles set for class named content on printed page

@page rules setting style for printed page

id named menubar will not print

Figure 7–14

To Link to an External Style Sheet

The following steps illustrate how to add a link to an external style sheet using a `<link>` tag and then save the HTML file.

- Click the sapperzein.html tab in Notepad++.
- Click to the right of the `>` at the end of `</title>` in line 8 and press the ENTER key twice.
- Type `<link rel="stylesheet" type="text/css" href="printpage.css" media="print" />` to enter the link to the external style sheet (Figure 7–15).
- Click File on the menu bar and then click Save on the File menu.

```

<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Sapperzein Galleries</title>
<link rel="stylesheet" type="text/css" href="printpage.css" media="print" />
<style type="text/css">
<!--
body {font-family: Verdana, Arial, sans-serif}

```

link to printpage.css external style sheet

media of style sheet set to "print"

Figure 7–15

To Print an HTML File

- 1 Click Print on the File menu and then click the Print button in the Print dialog box to print the sapperzein.html code.

To Test the External Style Sheet

The following steps illustrate how to test the printpage.css external style sheet.

- 1 Click the Internet Explorer button in the taskbar and then click the Return to Sapperzein Galleries link at the bottom of the Web page.
- 2 Click the Refresh button.
- 3 Click the Print icon arrow in the Command bar and then click Print Preview (Figure 7–16) to verify that the menu bar will not print.

Q&A

What other styles might be appropriate for printing?

As mentioned above, Web site visitors generally want to print the content of a Web page, not necessarily the format. You therefore might want to change the way that headings (h1 through h6) print. You could add a style to the printpage.css that prints all headings in one particular size (perhaps slightly larger than the general content of the Web page). You also might want to vary the margins from the Web page as displayed in the browser versus the margins on a printed page.

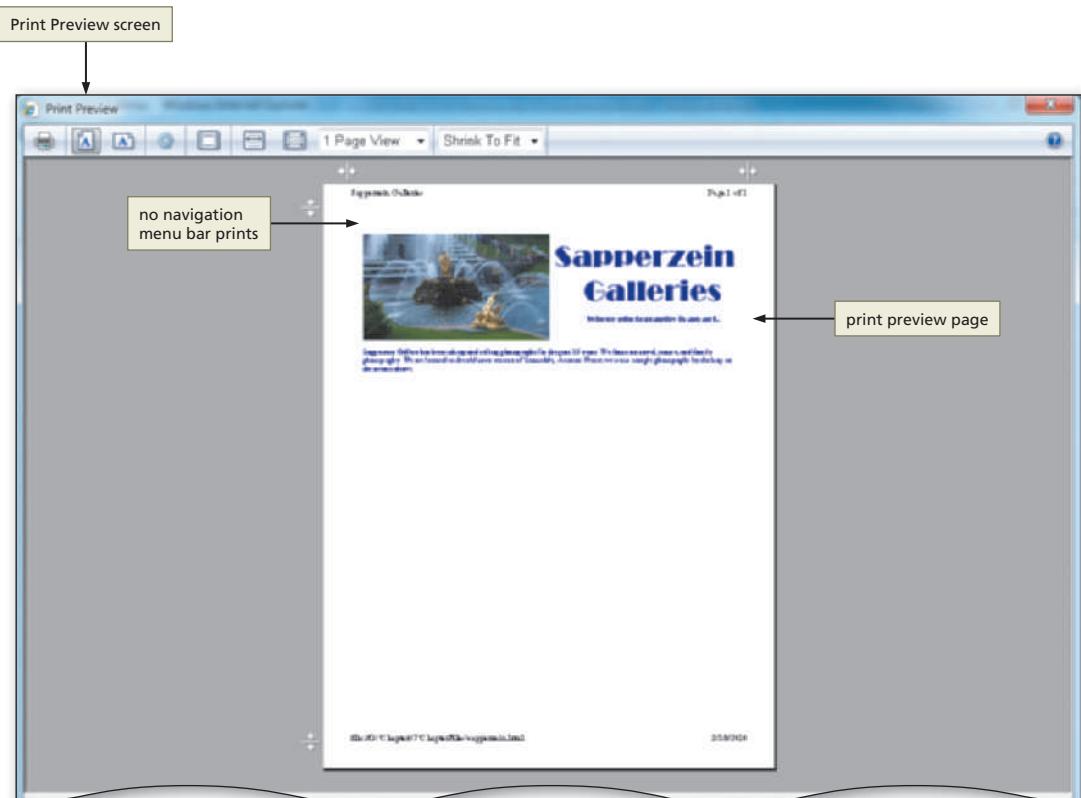


Figure 7–16

To Quit Notepad++ and a Browser

After you have viewed and printed the HTML and CSS files, the project is complete.

- 1** Close all open browser windows.
- 2** Click File on the Notepad++ menu bar and then click Close All.
- 3** Click the Close button on the Notepad++ window title bar.

BTW

Quick Reference

For a list of CSS properties and values, see the CSS Quick Reference (Appendix D) at the back of this book, or visit the CSS Quick Reference Web page (scsite.com/html6e/qr).

Chapter Summary

In this chapter, you have learned how to add advanced CSS features in embedded and external style sheets to give your Web pages a consistent and polished look and to add functionality. The items listed below include all the new HTML and CSS skills you have learned in this chapter.

1. Add an Embedded Style Sheet (HTML 340)
2. Create and Print an External Style Sheet (HTML 345)
3. Link to an External Style Sheet (HTML 346)
4. Create and Print an External Style Sheet for Printing (HTML 348)

Learn It Online

Test your knowledge of chapter content and key terms.

Instructions: To complete the Learn It Online exercises, start your browser, click the Address bar, and then enter the Web address scsite.com/html6e/learn. When the HTML Learn It Online page is displayed, click the link for the exercise you want to complete and read the instructions.

Chapter Reinforcement TF, MC, and SA

A series of true/false, multiple choice, and short answer questions that test your knowledge of the chapter content.

Flash Cards

An interactive learning environment where you identify chapter key terms associated with displayed definitions.

Practice Test

A series of multiple choice questions that test your knowledge of chapter content and key terms.

Who Wants To Be a Computer Genius?

An interactive game that challenges your knowledge of chapter content in the style of a television quiz show.

Wheel of Terms

An interactive game that challenges your knowledge of chapter key terms in the style of the television show, *Wheel of Fortune*.

Crossword Puzzle Challenge

A crossword puzzle that challenges your knowledge of key terms presented in the chapter.

Apply Your Knowledge

Reinforce the skills and apply the concepts you learned in this chapter.

Creating an Egyptian Web Site

Instructions: Start Notepad++ and a browser. Using your browser, open the apply7-1.html file from the Chapter07\Apply folder of the Data Files for Students. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files. The apply7-1.html file is the partially completed HTML file needed for this exercise. Figure 7-17 shows the Apply Your Knowledge Web page as it should appear in the browser after the necessary code is added.

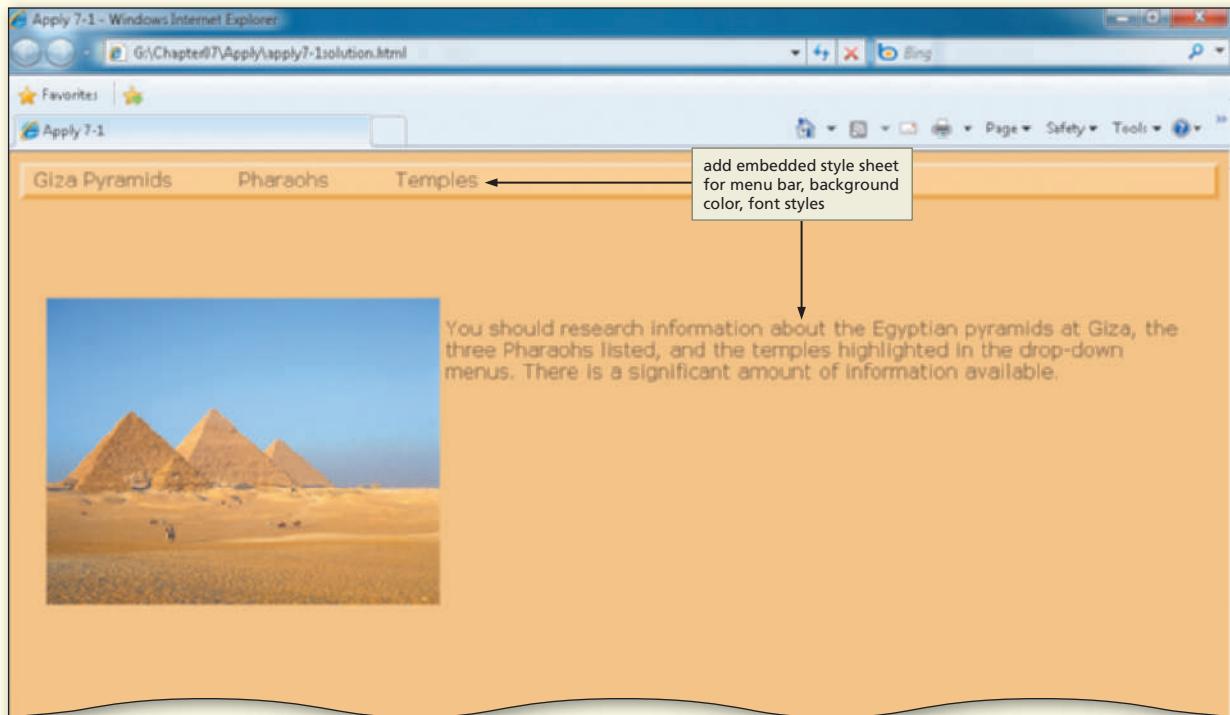


Figure 7-17

Perform the following tasks.

1. Examine the HTML file and its appearance in the browser.
2. Embed the following style sheet into the apply7-1.html file:

```
body {font-family: Verdana, Arial, sans-serif;
background: #edbf79}

a {text-decoration: none; color: #5a3702}

#menubar {border-top: 4px solid #f4dab2; border-right: 4px
solid #e0a140; border-bottom: 4px solid #e0a140;
border-left: 4px solid #f4dab2; background-color:
#f5cb8a; color: #5a3702; height: 23px}

.menu {float: left; padding: 0.1em 3em 0.1em 0.5em;
cursor: default}
```

```
.menu ul {display: none; position: absolute;
background-color: #f4dab2; color: #5a3702;
list-style: none; margin: 0 1em 0 0; padding: 0}
.menu ul li {display: block; font-size: small; padding: 0.2em}
div.menu:hover ul {margin: 0; padding: 0; display: block;
border-bottom: .15em solid #e0a140; border-left: .15em solid #e0a140}
div.menu ul li:hover {background-color: #e0a140}
div.content {margin-left: 20px; margin-right: 20px;
padding-top: 8%; color: #5a3702}
```

3. Insert the necessary HTML in the body of the Web page to utilize the classes within the embedded style sheet. (*Hint:* Refer to the chapter project to help you determine where these classes should be inserted.)
4. Save the file as apply7-1solution.html, validate the code, and print the file.
5. View the Web page in your browser.
6. Print the Web page as laid out on the screen.
7. Submit the solution in the format specified by your instructor.

Extend Your Knowledge

Extend the skills you learned in this chapter and experiment with new skills.

Creating an External Style Sheet

Instructions: You will create and save an external style sheet, and then link it to the file extend7-1.html from the Chapter07\Extend folder of the Data Files for Students. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files. This HTML file contains all of the text for the Web page shown in Figure 7–18.

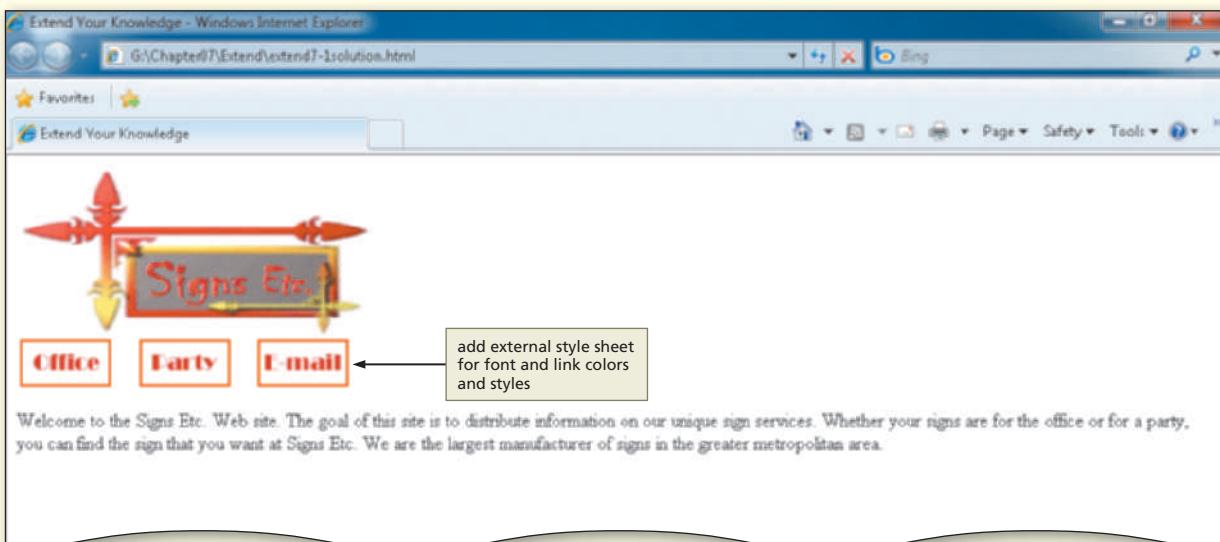


Figure 7–18

Continued >

Extend Your Knowledge *continued*

Perform the following tasks:

1. Start a new file in Notepad++, and add the following CSS code to create a new external style sheet that specifies the following:
 - a. all links red in color, bolder weight, and no text-decoration
 - b. table left margin of 10, color of red, and 14pt font size
 - c. hover color yellow with a red background
2. Save the file as extend7-1styles.css and print the file.
3. Open the extend7-1.html file from the Data Files for Students and add a link statement to the external style sheet extend7-1styles.css.
4. Save the file as extend7-1solution.html, validate the code, and then print the file.
5. Open the extend7-1solution.html file in the browser and print the Web page.
6. Submit the solution in the format specified by your instructor.

Make It Right

Analyze a document and correct all errors and/or improve the design.

Correcting the Valentine's Day Dinner and Dance Web Page

Instructions: Start Notepad++. Open the file makeitright7-1.html from the Chapter07\MakeItRight folder of the Data Files for Students. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files. The Web page is a modified version of what you see in Figure 7–19, but it contains some errors. Although the code in the embedded style sheet has the styles that you want, the CSS code is incorrect. Make the necessary corrections to the Web page to make it look like the figure.

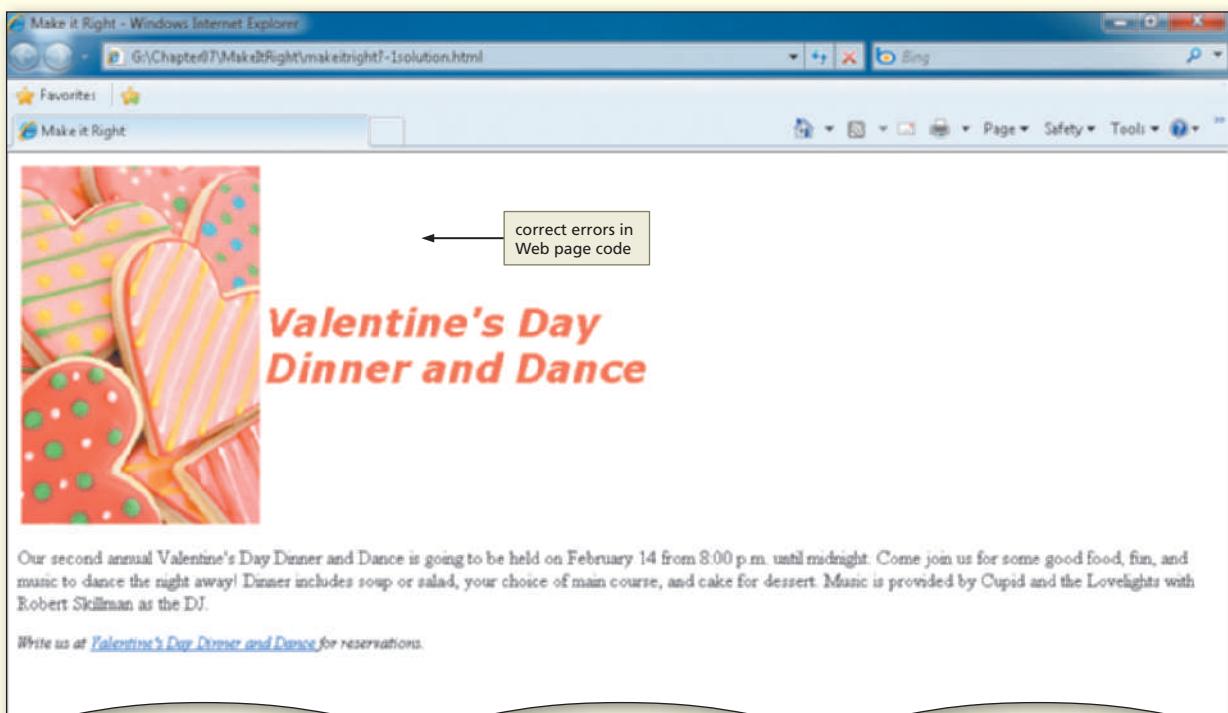


Figure 7–19

In the Lab

Lab 1: Creating an External Style Sheet for Printing

Problem: You have been asked to create a style sheet that can be used for printing across a number of people's personal Web pages. You decide to set the printing styles in an external style sheet so that it can easily be linked to any number of Web page files. You use the lab7-1.html file for this exercise. The HTML code is completed for the Web page, as shown in Figure 7–20a. You should create the external style sheet that results in Figure 7–20b.

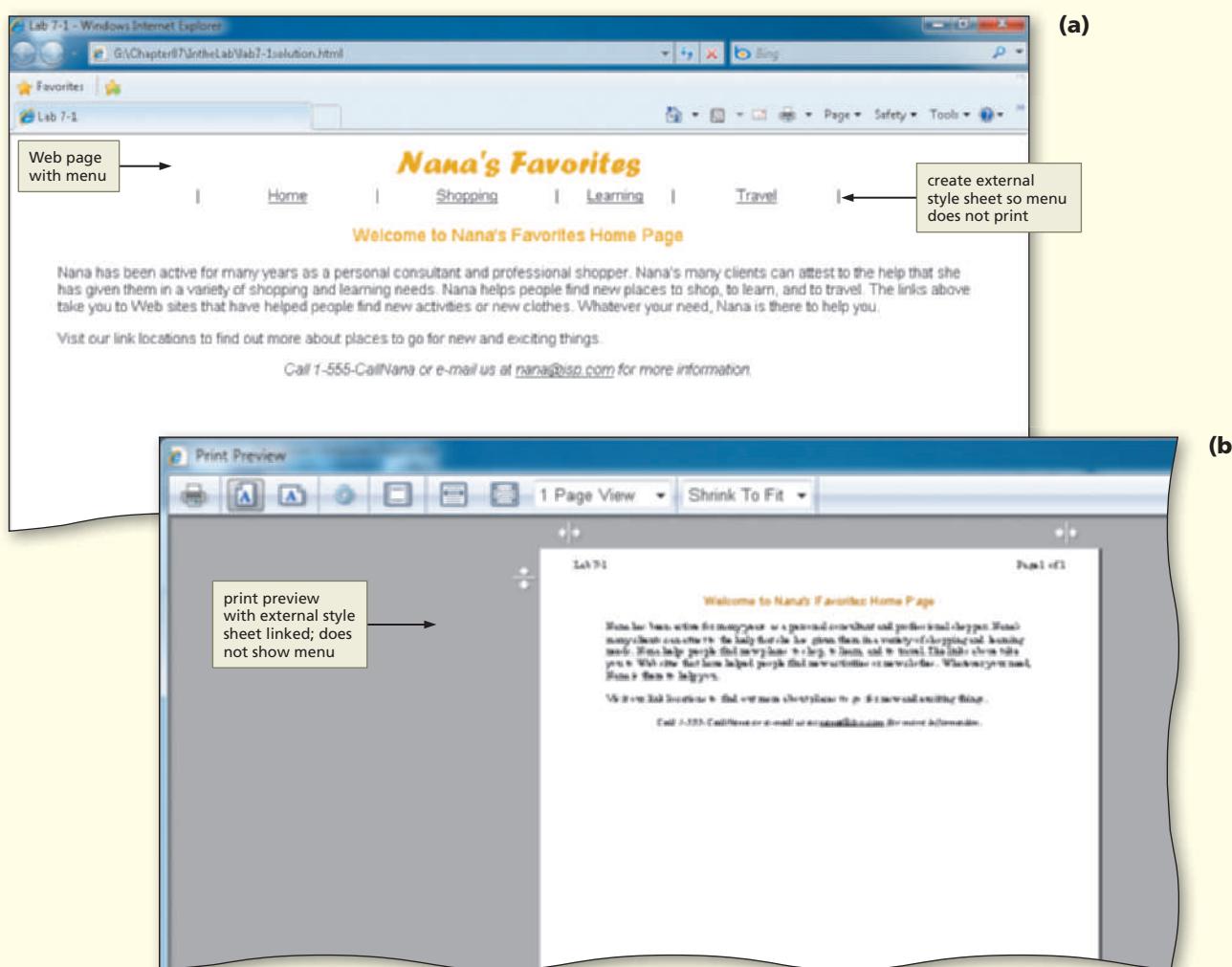


Figure 7-20

Instructions: Perform the following steps to create an external style sheet and link it to an HTML file from the Data Files for Students.

1. Using Notepad++, create a new file for an external style sheet that is used for printing.
2. Using CSS, create the style structure so that the menu bar on the Web page is not printed. (*Hint:* Use the menubar class.) Also, align the content of the Web page on the left with a font size of 12 points in Times New Roman font for the printed page. (*Hint:* Use the content class.)
3. Save the file as lab7-1print.css.

Continued >

In the Lab *continued*

4. Open the HTML file lab7-1.html in the Chapter07\IntheLab folder of the Data Files for Students. Add a link to the external style sheet, lab7-1print.css.
5. Save the HTML file in the Chapter07\IntheLab folder as lab7-1solution.html. Validate the file and then print it.
6. Open the file, lab7-1solution.html in the browser. Use Print Preview to test if your printing external style sheet works correctly.
7. Print the Print Preview view of the Web page.
8. Submit the solution in the format specified by your instructor.

In the Lab

Lab 2: Adding External and Inline Style Sheets

Problem: Your father's business, Bold Ones Painting, is participating in the Home and Garden Show and wants to create a Web page to notify people about the event. The event coordinator asks you to create a Web page that contains information about the business and an e-mail address link, as shown in Figure 7–21. The Web page should have a link to the external style sheet, lab7-2styles.css, which is in the Chapter07\IntheLab folder of the Data Files for Students. The external style sheet is not complete, so you must add some selectors and declarations to complete it.

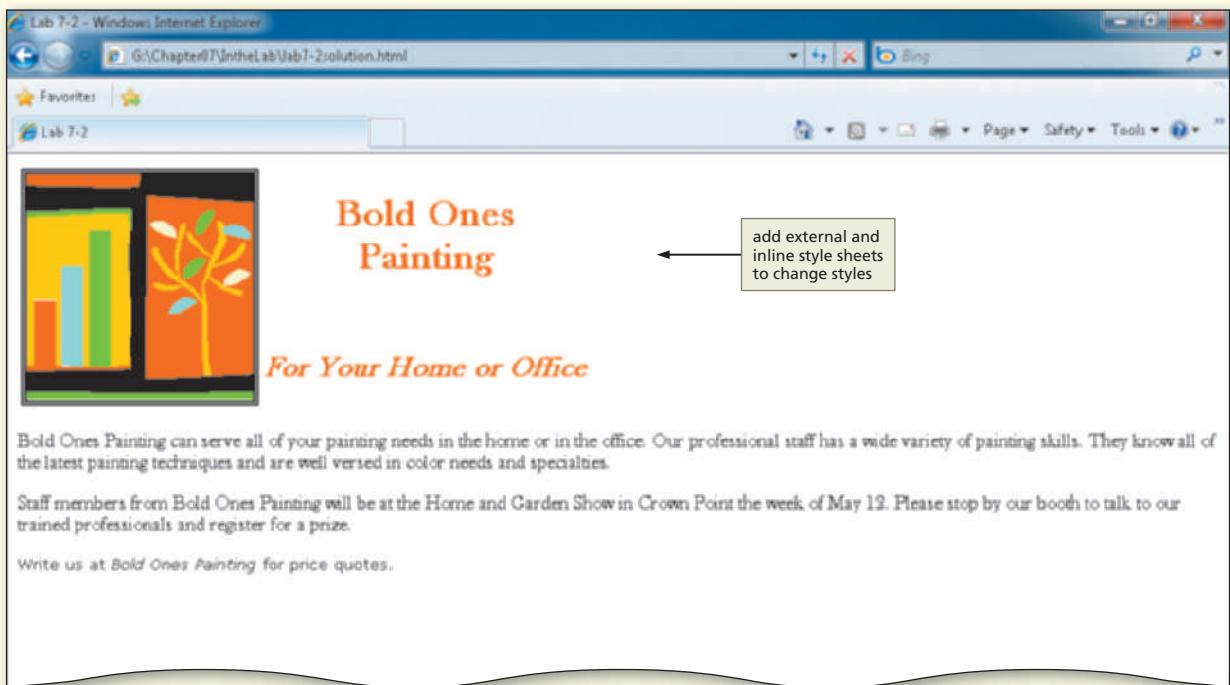


Figure 7–21

Instructions: Perform the following steps.

1. Using Notepad++, open the HTML file lab7-2styles.css in the Chapter07\IntheLab folder of the Data Files for Students.
2. Add the following styles to the external style sheet:
 - a. h1 and h2: "Baskerville Old Face", "Calligrapher", "Arial" font in a bolder weight and color #ff3300
 - b. paragraphs: "Sans Serif", "Baskerville Old Face" font, color of black, with 10 pt left margin
3. Save the file using the same file name.

4. Using Notepad++, open the HTML file Chapter07\IntheLab\lab7-2.html in the Data Files for Students.
5. Add the link necessary to apply the styles from the lab7-2styles.css external style sheet.
6. Add an inline style sheet to the last paragraph of text that sets the font to Verdana and size 10pt.
7. Within the borderless table provided, add the image boldones.jpg in the top data cell of the Web page. Insert an **<h1>** heading tag next to the image that contains the words Bold Ones Painting with a line break as indicated. Also add an **<h2>** heading with the words For Your Home or Office in italic font. (*Hint:* Use an inline style.)
8. Save the file as lab7-2solution.html, validate the file, and print it.
9. Open the Chapter07\IntheLab\lab7-2solution.html file in your browser.
10. Print the Web page.
11. Submit the solution in the format specified by your instructor.

In the Lab

Lab 3: Developing an External Style Sheet for Pop-ups

Problem: The owners of Sabatina's Pizza want to continue enhancing their Web site. They would like to include a pop-up that displays the ingredients and nutrition information for some of their pasta dishes, as shown in Figure 7–22. The file lab7-3.html is an HTML file that contains some of the structure of the Web page. The file, lab7-3.html, is included in the Chapter07\IntheLab folder of the Data Files for Students. In this exercise, you will create an embedded style sheet that pops up the text under the pasta image when a user hovers over the image. Research the three pasta dishes (fettuccine, lasagna, and ravioli) to get an idea of ingredients and nutrition information that you will use in your pop-up text.



Figure 7–22

Continued >

In the Lab *continued*

Instructions: Perform the following steps.

1. Open the lab7-3.html file in Notepad++ and add an embedded style sheet that includes the styles shown in Figure 7–22.
2. Add an embedded style sheet to set the following:
 - a. links should be 1em sans-serif font with padding of 5 and 10 pixels and margins of 0 0 1 pixels
(Hint: Use shorthand properties); the right border should be 5 pixels and color #ff1828
 - b. when a user hovers over a link, the right border should be style double with 5 pixels in white
 - c. at startup, there should be no display of text beneath any of the three images of pasta; use the ` ` tags to control the display of text pop-up upon hovering; position the block of text as absolute with a top margin of 220 pixels and left margin of 550 pixels; the width of the block should be 180 pixels at a minimum, depending on how much text you find in your research; use padding of 5 pixels and margins of 10 pixels; the text should be white on a black background.
3. Add the ingredient/nutrition information to your HTML file.
4. Save the file as lab7-3solution.html, validate it, and then print the HTML code.
5. Open the Web page in your browser to test that the pop-ups display when you hover over each pasta image, as shown in Figure 7–22. Print the Web page.
6. Submit the solution in the format specified by your instructor.

Cases and Places

Apply your creative thinking and problem-solving skills to design and implement a solution.

• EASIER •• MORE DIFFICULT

• 1: Finding CSS Information Online

Browse the Internet to find two Web sites that discuss advanced Cascading Style Sheets (CSS). How else can CSS be used in your Web development that differs from how it has been used in Chapters 2 through 7? What other properties and values can you use to format your Web pages? What are other ways that you can create external style sheets to be used for printing only? Think about these other techniques and be prepared to discuss what other advanced CSS you would like to utilize.

• 2: Trying More Styles

Mr. Sapperzein is very impressed by the use of style sheets in the Sapperzein Galleries Web site and would like to explore additional styles that can be applied. Using Appendix D, find three CSS properties that were not used in Chapter 7 and that are appropriate for this type of Web site. Make sure to do a thorough analysis on this; i.e., do not just find three properties, but determine through your analysis and design process what three properties would be beneficial to the Web site. Then modify the styles7.css style sheet that you used in the chapter project to include these properties and values. Save the style sheet as styles7new.css in the Chapter07\CasesandPlaces folder. Update the link in the Web page, sapperzein.html, and then view the Web page in the browser using the new style sheet. How did the use of these new properties improve the appearance of the Web page?

•• 3: Applying Pop-up Text

Create a Web site that utilizes several pictures. Use the basic pop-up techniques covered in this chapter to pop up text rather than larger images when a Web site visitor hovers over one of the images. Determine how you could use this text pop-up function and why it might be helpful in your own Web development. Write a one-page paper explaining your thoughts on text pop-ups created using CSS.

•• 4: Determining How CSS Could Be Used on a Web Site

Make It Personal

Select a Web site with which you are familiar. Verify that the Web site does not utilize any of the three types of style sheets. Develop a graphic of the Web site hierarchy. Determine how the three types of style sheets could be utilized in this Web site and develop an outline explaining how they would help enhance pages or sections of the site, add style consistency, or make the site easier to maintain. Write a proposal to the owners of the Web site that describes the features you could add with style sheets and the benefits of doing so, relative to the formatting techniques currently used in the Web site. As an example, you might want to address the number of times that a particular `` tag is used in the site and contrast that with the ease of using one external style sheet and a link statement per page. Use other ideas as discussed in the project to stress the other benefits of style sheets. Write the proposal in the form of a bid, giving time estimates and costs associated with the development effort. Include your hierarchy chart and style sheet outline as appendices to the proposal.

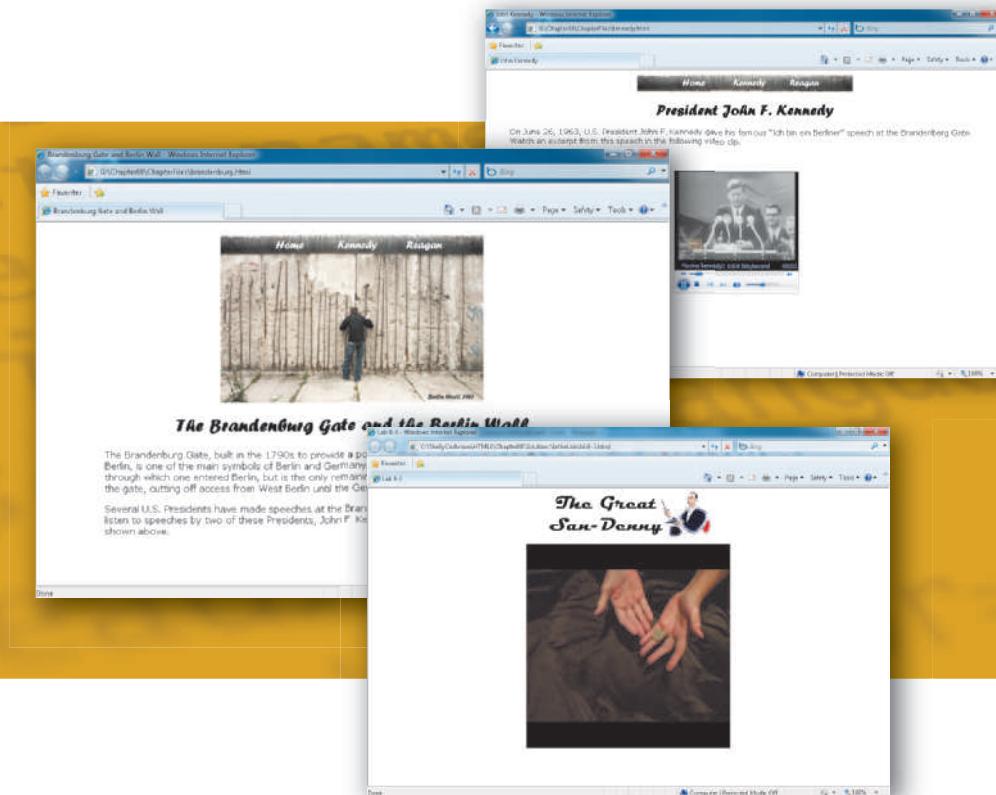
•• 5: Completing the Sapperzein Galleries Web Site

Working Together

In this team exercise, you will complete the Sapperzein Galleries Web site. There are four additional dummy Web pages provided in the Data Files for Students in the Chapter7\ChapterFiles folder. The four remaining Web pages are called `cactus.html` and `flowers.html` that drop down from the Nature menu selection and `group.html` and `individuals.html` that drop down from the Family menu selection. As a first step, determine in your team if the Nature menu item is of interest to the members of the team. If it is not, determine another category to replace Nature and two drop-down submenu categories to replace Cactus and Flowers. All teams should complete this project using the Family category. Once you determine the third category, you need to take the pictures relative to these four final drop-down items and complete the Web pages provided.

This page intentionally left blank

8 Adding Multimedia Content to Web Pages



Objectives

You will have mastered the material in this chapter when you can:

- Describe the benefits and limitations of multimedia in Web sites
- Identify audio and video formats
- Describe parameters for embedded multimedia
- Add an audio clip to a Web page
- Add a video clip to a Web page

8 | Adding Multimedia Content to Web Pages

Introduction

In previous chapters, you used HTML tags and CSS code to change the way a Web page is displayed in a Web browser. You learned how to collect data from Web site visitors using Web page forms. You also learned advanced CSS techniques to create drop-down menus and pop-ups. In this chapter, you learn how to insert multimedia content in the form of audio and video clips.

Project — Adding Multimedia to a History Class Web Site

In your U.S. History class, you are studying the Berlin Wall. Several U.S. Presidents made famous speeches at the Brandenburg Gate of the Berlin Wall. Your class created a Web site that describes the historical significance of two of these speeches, one by President Kennedy, and one by President Reagan. Your instructor, Ms. Amy Sirkо, would like you to enhance that Web site to make these speeches as “real” as possible.

The history class Web site consists of a home page (Figure 8–1a) and two linked Web pages (Figures 8–1b and 8–1c). The home page contains a picture of the Berlin Wall that is used as an image map for navigation. The other two pages use part of that image for navigation to speeches by Presidents Kennedy and Reagan, together with text that explains the most significant line in each speech. You feel that the addition of multimedia files (one audio and one video) might help make Ms. Sirkо’s history class Web site more engaging for the students. It is one thing for a student to read a historically significant line from a speech, but it may have greater impact to see or hear the line being delivered. Multimedia content can provide that valuable experience. Because the use of multimedia content is complex, you must first research how to add multimedia to Web pages. Ms. Sirkо is supportive of the plan and encourages you to start as soon as possible.

Overview

As you read this chapter, you will learn how to add the multimedia content to the two Web pages shown in Figures 8–1b and 8–1c by performing these general tasks:

- Plan the use of multimedia in the Web site
- Enter the HTML code to insert an audio clip
- Enter the HTML code to insert a video clip
- Save the files as HTML files
- View the Web pages and HTML code in your browser
- Validate the Web pages
- Test and print the Web pages

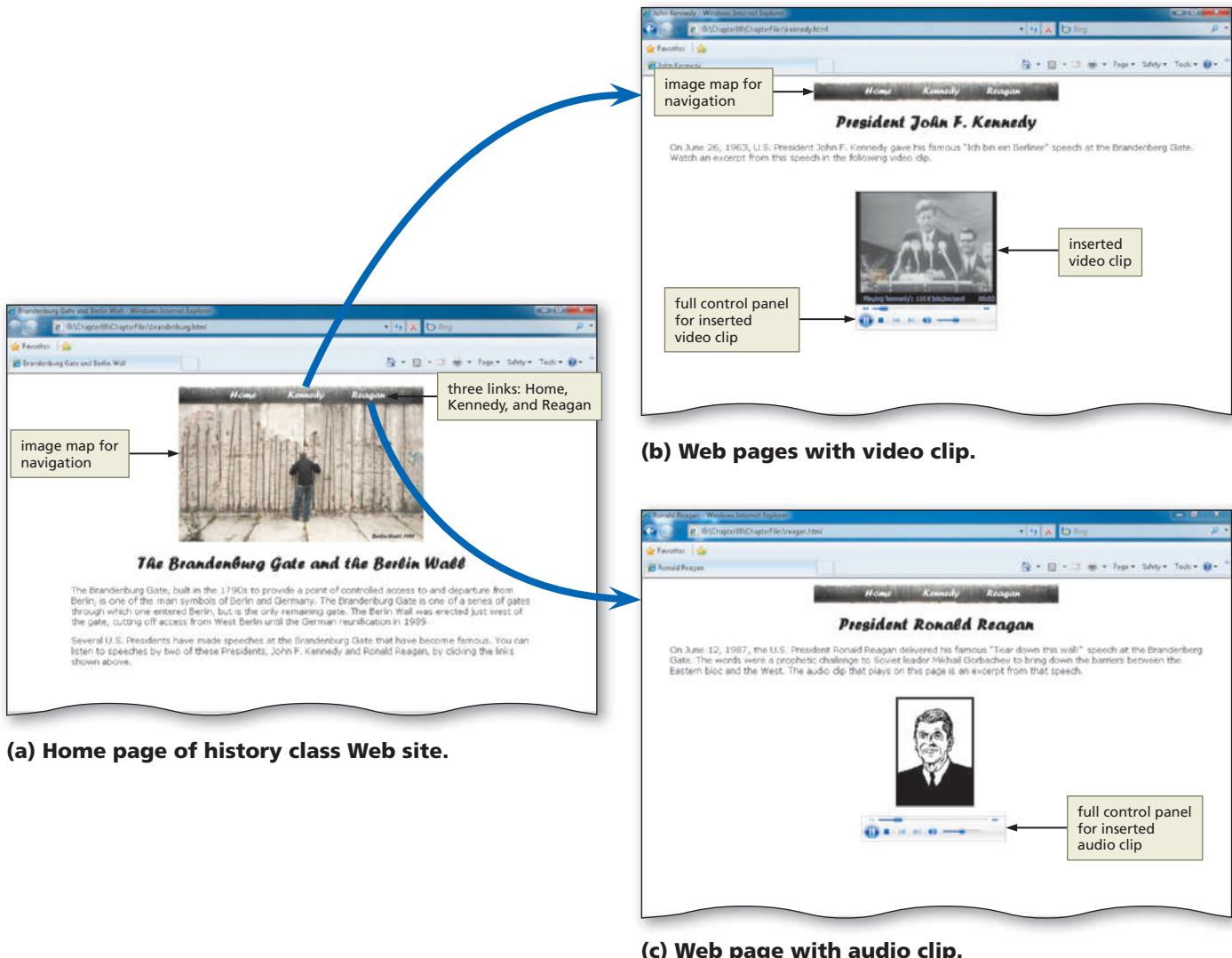


Figure 8–1

General Project Guidelines

As you create Web pages, such as the chapter project shown in Figure 8–1, you should follow these general guidelines:

- 1. Plan the Web site.** First, you should determine if using multimedia content is appropriate for your Web site. If your subject matter is such that audio or video would enhance the visitors' experience, multimedia is a good choice.
- 2. Analyze the need.** In the analysis phase of the Web Development Life Cycle, you should analyze what content to include on the Web page. Chapter 8 introduces a new Web development technique, adding multimedia content to your Web page; this technique can enhance the text and graphics content that you have developed in other chapters.
- 3. Choose the content for the Web page.** You always want your content to add positively to your Web site visitors' experience. You therefore have to assess multimedia content just as you would text or graphics content. Many sources of content are available for Web sites. You can create your own video and audio clips easily and insert those into the appropriate Web pages. You can also take advantage of free use audio and video clips available online.

Plan Ahead

**Plan
Ahead***(continued)*

4. **Determine the type of multimedia to incorporate into your Web pages.** The type of multimedia that you select is based on the purpose of the content. If you want to provide background music, an audio clip is fine; there would be no need for a video clip in this instance. For more complex topics though, a video clip might be more effective. As an example, if you are explaining to someone where middle C is on the piano, a video clip is more appropriate. If you want to give someone the experience of hearing a speech that does not explain things in a "how to" approach, then an audio clip is acceptable.
5. **Create or find the multimedia content and insert it into the Web site.** Once the analysis and design are complete, the Web developer creates or finds the appropriate multimedia content for the Web site. Again, consider free use content sources. Otherwise, developing multimedia content on your own might be a fun experience. Good Web development standard practices should be followed in the steps that you take to insert the multimedia.
6. **Test all Web pages within the Web site.** An important part of Web development is testing to assure that you are following XHTML standards. In this book, you use the World Wide Web Consortium (W3C) validator that allows you to test your Web page and clearly explains any errors you have. When testing, you should check all content for accuracy. Finally, all of the Web pages with multimedia content should be validated per the standard set throughout this book.

When necessary, more specific details concerning the above guidelines are presented at appropriate points in the chapter. The chapter also will identify the actions performed and decisions made regarding these guidelines during the creation of the Web page shown in Figure 8–1 on the previous page.

Using Multimedia

The popularity of the World Wide Web (the Web) is due in part to the ability to view Web pages that include graphic images, audio, and video. These additions can be wonderful for a Web site and give the users more enhanced and effective browsing experiences. Sometimes, however, the addition of multimedia can distract from the Web site message. Always remember the purpose of the Web site. If the multimedia content enhances that purpose, it should be included. If the multimedia content distracts from the purpose of the Web site, then you should reconsider using it.

Multimedia is defined as the combination of text, sound, and video to express an idea or convey a message. In the past, it was somewhat prohibitive to use multimedia on the Web. Most people dialed up to connect to the Internet, so the bandwidth (i.e., the capacity for data transfer) was not there to accommodate the large files required for audio, video, and even some graphic images. The Web of yesterday was mostly text-based with a few background colors and small graphic images. Today, most people have broadband Internet connections with a much greater capacity for data transfer. Multimedia Web pages that include large graphics, audio, and video are therefore common today.

Multimedia is used widely on Web pages. Many companies utilize videos to show potential customers new products or how to use their existing products. You may be able to review a medical procedure online before having the procedure done yourself. You can view clips of movies or hear segments of audio recordings from Web pages that provide content in those formats. **Podcasts**, a series of audio or video clips that are released in a sequence, are becoming more prevalent in both academic and corporate settings. Additionally, some instructors rely on multimedia content to stress important aspects of a class. It is important to determine where multimedia content may enhance the learning or viewing experience. You do not want to divert your Web site visitors with distracting

multimedia. The Web site enhanced in this chapter is one based on historical information. To improve the user experience, you insert an audio clip and a video clip on two different Web pages. These clips enhance the content of each Web page nicely; they do not distract from the message.

This chapter provides an introduction to the use of multimedia in Web development. The chapter focuses on two different forms of multimedia: audio and video. Other multimedia formats exist, but are not covered in this chapter, including Java applets, Flash, and automated slideshows. The finished Web pages (Figure 8–2a and 8–2b) are more useful for the Web site visitors because they contain relevant audio and video clips that provide valuable multimedia content.

BTW

Slideshows, Java, and Flash

You can easily create a slideshow from your pictures using most movie editing software. A Java applet is a program written in the Java programming language that can be included in a Web page. Adobe Flash is a popular multimedia platform not discussed in this chapter. Search the Web for more information on these formats.

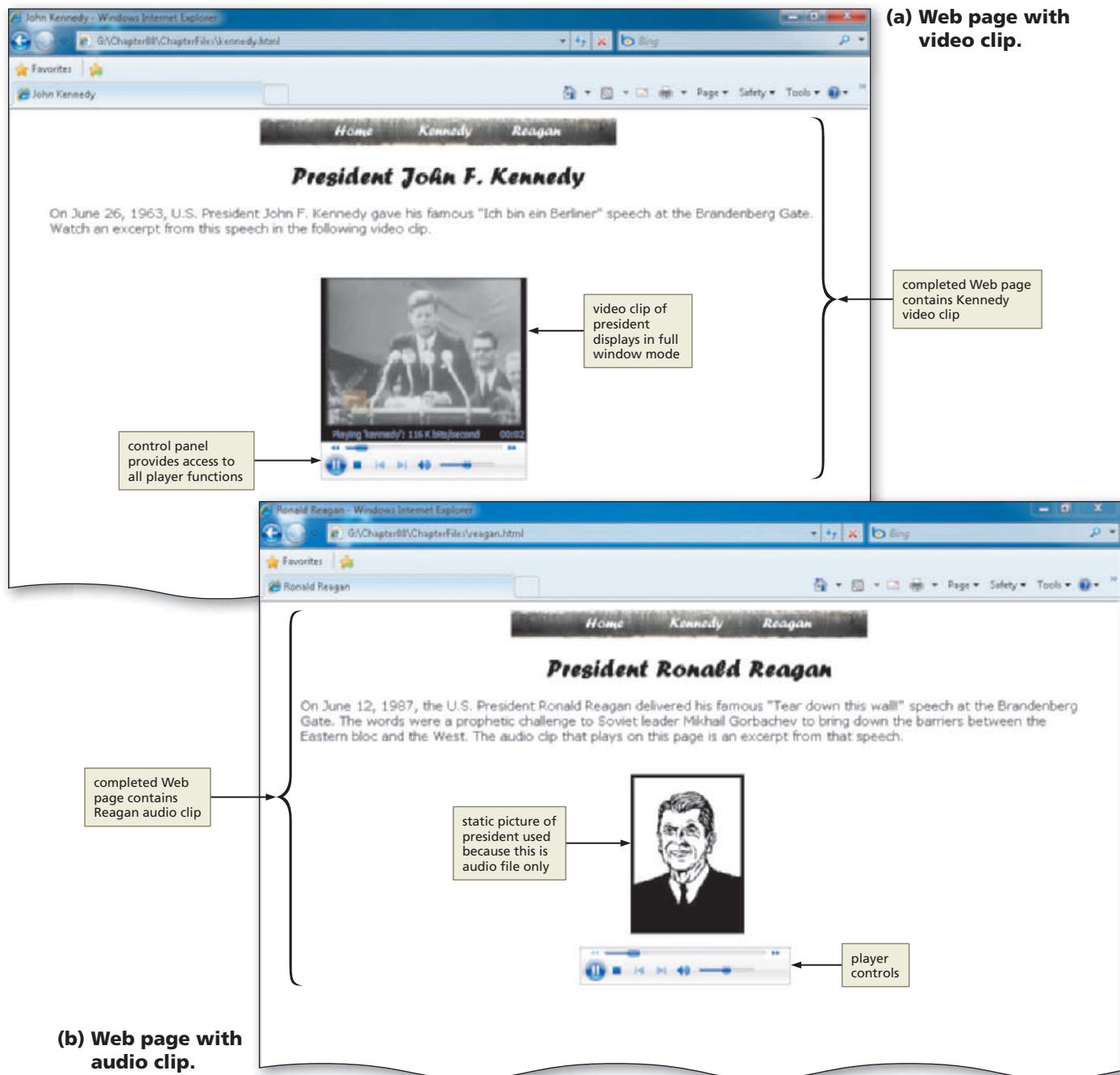


Figure 8-2

BTW**Multimedia Sources**

Search the Web for sources that you can use to download free use audio and video clips. Determine a way to keep track of the site, the specific licensing language, and the clips that you download. This is a great way to make sure that you follow the guidelines required by that Web site.

BTW**Multimedia Software**

Many software products are available to make and edit audio and video content. Search for the names of products that you can use. Price levels vary, but you may be able to find free trials that you can download.

Creating or Finding Multimedia Files

You can obtain multimedia files by creating them yourself or finding files that are already available. The good thing about creating your own multimedia files is that you do not have to be concerned with licensing agreements. The bad thing about creating your own multimedia files is that they may not be the same quality as those that are created professionally. You can create your own audio files using a microphone and software designed to edit digital files. Many software options are available today for audio creation and editing. If you do use any portion of files that have been professionally developed, be certain that you understand and apply all licensing requirements.

For video files, a digital camcorder allows you to create clips that can be used on a Web page. This is what has made video Web sites such as YouTube so popular. For the Windows environment, Windows Movie Maker is available on many new computers. MAGIX Movie Edit Pro software is another option that provides online support, blogs, newsletters, and access to free downloads of a variety of software. Corel VideoStudio has simple and more advanced menu system options that novice movie editors and professionals alike can utilize comfortably. Roxio Creator is another software option that allows you to create professional-quality videos. Many other software options for Windows offer free trial versions that can be used for several days or weeks.

For Mac users, iMovie is installed on most new computers and is an excellent option. DVD Studio Pro is another Mac option that provides support for all professional audio formats. Final Cut Studio gives you the flexibility that you need for movie project work that you do individually or in collaboration with a team. Final Cut Studio allows you to have technical control together with creative options. Many other multimedia software products are available for the Mac operating system as well.

Adobe Premiere Elements is another highly rated software solution for movie editing that can be used on either the Mac or Windows operating systems. Adobe provides a frequently updated online library of templates and effects. As with many multimedia software products, Premier Elements provides the capability to edit as a novice or as a professional.

You might try downloading a free trial of the software, if available, and using it to edit a movie that you have made. You could even try a few different programs to see how they work. Most multimedia software operates in a similar fashion and has a user-friendly help utility that you can use to guide you through the process of editing. Most products also provide templates and effects that you can use to enhance the creative aspects of presenting multimedia.

To find multimedia resources on the Web, search for “free audio or video.” Many Web sites contain these types of multimedia files. Again, be certain to understand and apply the license agreements that accompany any multimedia content that you find on the Web.

Embedded vs. External Multimedia

Embedding media is a similar in concept to inserting inline images. The embedded media files appear within the Web page and users have access to the audio or video player controls right on the page. Because the media file is embedded directly into the Web page, you can complement the audio or video clip resource with surrounding text or graphical images. The `<object>` tag is used to insert embedded content. Figure 8–3 shows how the embedded video clip of President Kennedy’s speech is supplemented with a heading, text, and a menu bar.

External media files are accessed through a link that your Web site visitor clicks. This gives your visitors the option of downloading or not downloading the file. Unlike embedded media, the external media is displayed out of context with the Web page

that calls it. The `<embed>` tag is used to define external interactive content or plug-ins. Using external links is a Web development practice that is used often in Web sites such as YouTube. Although external media files are frequently used, the `<embed>` tag is not XHTML compliant, so this text will not cover that tag.

BTW**The `<embed>` tag**

The `<embed>` tag is not XHTML compliant, so it is not used in this chapter. However, many people do use the `<embed>` tag. Search the Web for more information on this tag.

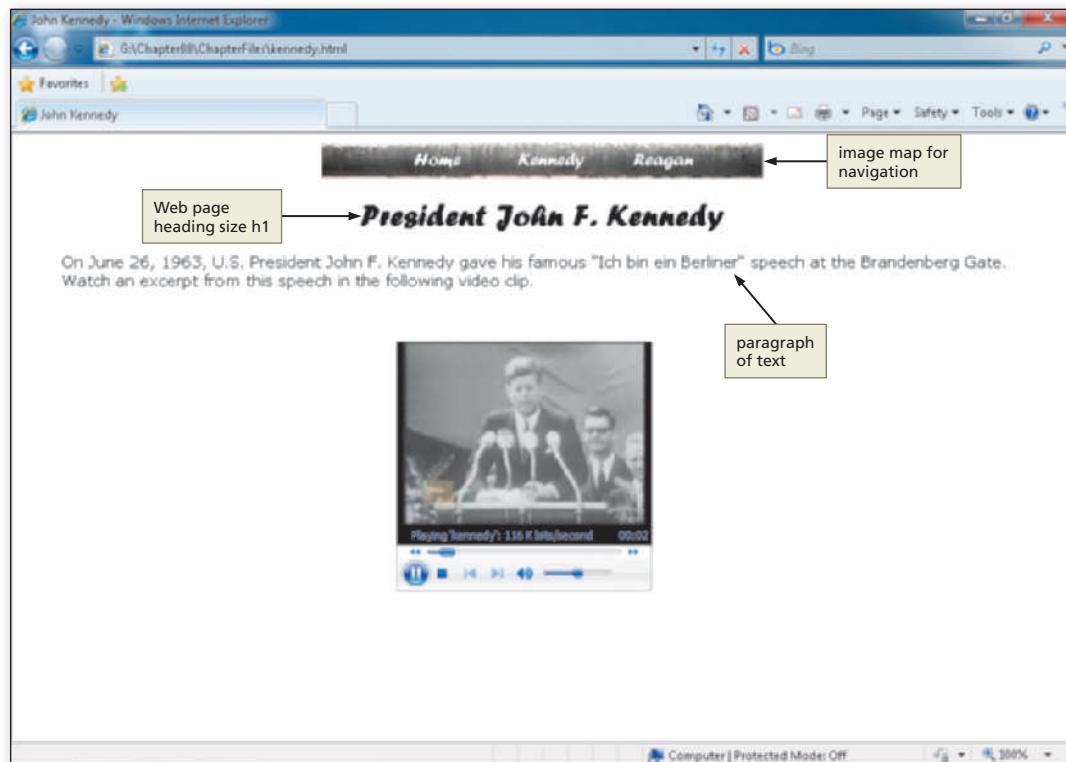


Figure 8–3 Completed Web page at startup with video clip.

Media Players and Plug-Ins

The functionality of a browser includes the ability to display text and the graphics formats discussed in earlier chapters. In order to play an audio or video file, a browser needs the help of an application called a player or a plug-in. A **media player** is computer software that is used to play multimedia files. Most software media players support an array of media formats, including both audio and video files. Windows Media Player, used in this chapter, plays both audio and video. Some media players work only with audio files, known as audio players, while others work only with video files, known as video players. The producers of these players focus on providing a better user experience as they are specifically tailored toward the media type. Windows Media Player is included with Microsoft Windows. Mac operating systems are preloaded with QuickTime Player for playing movies and iTunes for playing a variety of media formats.

A **plug-in** (also called an add-in or add-on) is extra software that is added to the browser (or other program) to provide a capability that is not inherent to the program (the browser in this case). In other words, for an embedded media file to work in a browser, the Web site visitor needs to have the correct plug-in. Most browsers do have a variety of plug-ins installed, but the Web site visitor can also download and install necessary plug-ins from the browsers' manufacturers. Common plug-ins are Windows Media Player, Apple QuickTime Player, Adobe Flash and Shockwave, Microsoft Silverlight, and Adobe Acrobat.

BTW**Players and Plug-ins**

Investigate what plug-ins are available for the version of browser that you use. (In Internet Explorer, click on the Tools menu, then select Manage Add-Ons.) Search the Web for information about the listed plug-ins and determine if there are other plug-ins that would be helpful that you can download.

BTW

Updating or Installing Plug-ins

Your Web site visitors may not have the plug-ins that are required to play the multimedia content on your Web site. Many Web sites discuss how to best provide users with access to such software.

Internet Explorer (used in this book) utilizes proprietary ActiveX controls. When IE encounters a multimedia file, it searches for the appropriate ActiveX control. If it cannot find it, users are most often asked if they want to install that control. Some browsers direct users to the Web site from which they can download the required plug-in. A suggestion to Web developers is to utilize common formats, such as .wmv, .mp4, .m4u, .flv, and .mp3. It is also a good idea to let your users know what format is being used. If the plug-in is not installed, IE generally provides the capability to install it. Many Web sites today also provide a link to the appropriate manufacturer for an ActiveX control needed to play the file.

As you will see in the following section, the various audio and video formats can be played on a variety of players. For embedded multimedia, it is good to use a format supported by multiple players.

Audio and Video File Formats

A variety of audio and video formats can be used on the Web. Table 8–1 lists the most commonly used audio file formats, and Table 8–2 lists common video file formats. Audio files that are used on the Web often utilize file compression techniques. This reduces the size of the file, but it can also diminish the sound quality. Uncompressed audio formats included in Table 8–1 are: AIFF, AU, and WAV. In this project, you insert an .mp3 file.

Table 8–1 Common Audio File Formats

Format	File Extension	Description
AIFF	.aiff	<ul style="list-style-type: none"> The standard audio file format used by Apple Limited to synthesizers and music files Can be much smaller in size than other formats
AU	.au	<ul style="list-style-type: none"> The standard audio file format used by Sun, UNIX, and Java Can be compressed
MIDI	.mid .rmi	<ul style="list-style-type: none"> Musical Instrument Digital Interface (MIDI) Limited to synthesizers and music files Can be much smaller in size than other formats
MP3	.mp3	<ul style="list-style-type: none"> One of the most popular formats for music storage Compresses files to approximately 1/10 the size of uncompressed files
MP4	.mp4	<ul style="list-style-type: none"> Created on basis of the QuickTime format; used for audio and video Is a quicker, faster, high-quality media Can be used with QuickTime Player, Adobe Flash Player, or RealPlayer
RealAudio	.ra .ram	<ul style="list-style-type: none"> Designed for streaming audio over the Internet Sound quality not as good as other formats
WAV	.wav	<ul style="list-style-type: none"> Standard audio format for Windows Commonly used for storing uncompressed CD-quality sound files Compression is available to reduce file size

The video format that you choose depends on the visitors you expect to visit your Web site. Are most of your Web site visitors users of Macs or PCs? What is the level of computer (e.g., amount of RAM or cache) and connection speed (e.g., broadband) for your average user? In this project, you insert a .wmv file.

Table 8–2 Common Video File Formats

Format	File Extension	Description
AVI	.avi	<ul style="list-style-type: none"> • Audio/Video Interleaved • Developed by Microsoft to use with Windows • Can contain both audio and video data • Requires Windows Media Player
Flash Video	.flv	<ul style="list-style-type: none"> • Developed by Adobe • Format of choice for embedded video on the Web • Used by Google and YouTube
MPEG	.mpg	<ul style="list-style-type: none"> • Moving Pictures Group • Can be highly compressed resulting in small file size • Requires QuickTime Player, RealPlayer, or Windows Media Player, which are easily downloaded
	.mpeg	
	.mp3	
MP4	.mp4	<ul style="list-style-type: none"> • Created on basis of QuickTime format; used for audio and video • Is a quicker, faster, high-quality media • Can be used with QuickTime Player, Adobe Flash Player, or RealPlayer
QuickTime	.mov	<ul style="list-style-type: none"> • Developed by Apple Computer for both Windows and Apple computers • File compression can result in smaller file size • Requires QuickTime Player or Adobe Flash Player, which are easily downloaded
RealVideo	.rm .rv	<ul style="list-style-type: none"> • Proprietary video format developed by RealNetworks • Requires RealPlayer
Shockwave Flash	.swf	<ul style="list-style-type: none"> • Small Web Format • Can contain audio, video, or animations • Requires Adobe Flash Player
Windows Media	.wmv	<ul style="list-style-type: none"> • Developed by Microsoft • Originally designed for Internet streaming applications • Requires Windows Media Player

Adding an Audio File to a Web Page

Ms. Amy Sirk's history class Web site consists of three files. The first Web page, brandenburg.html, is the home page, which contains an image map for Web site navigation together with two paragraphs of text. This Web page will not change. The next Web page, reagan.html, is the file into which you will insert an audio file, reagan-audio.mp3. The third page, kennedy.html, will contain a video clip. All Web pages in this Web site are linked to the external style sheet, styles8.css. The external style sheet file is also not changed. Please review the file, however, so that you are familiar with the styles used in the Web site. Pay special attention to the classes that are named in the external style sheet and review how those classes are used in the kennedy.html and reagan.html files.

BTW**Audio Clips**

Many good Web design sites discuss the use of audio clips in Web development. Search for ideas of how you can most effectively utilize an audio clip.

BTW**Multimedia
<object> Tag**

The World Wide Web Consortium (W3C) has a wealth of information about the use of the `<object>` tag. For more information about attributes that can be used with this tag, visit the W3C Web site and search for `<object>` tag.

The Object Tag

The object element supports many different media types, including: pictures, sounds, videos, as well as other objects. The term **object** is used to describe the things that people want to place in HTML documents. Appendix A lists the attributes that can be used with the `<object>` tag.

To insert the audio clip, you will use the same classid (the Windows Media Player) that you will use for the video clip. The `<object>` statement used for the audio clip insertion is as follows:

```
<object classid="clsid:6BF52A52-394A-11D3-B153-00C04F79FAA6"
       height="45" width="250">
    <param name="URL" value="reagan-audio.mp3" />
</object>
```

The **classid** is an attribute that specifies which ActiveX control is being used. Table 8–3 shows the classids for the available ActiveX controls. For the insertion of an audio clip, you use the classid for Windows Media Player. It is imperative that you specify the classid correctly, or the multimedia content may not work correctly or at all. The classid is not case sensitive, so you can type the classid string as shown in Table 8–3 or use all lowercase letters. This information can also be found at each manufacturer’s Web site.

Table 8–3 Class IDs for ActiveX Controls

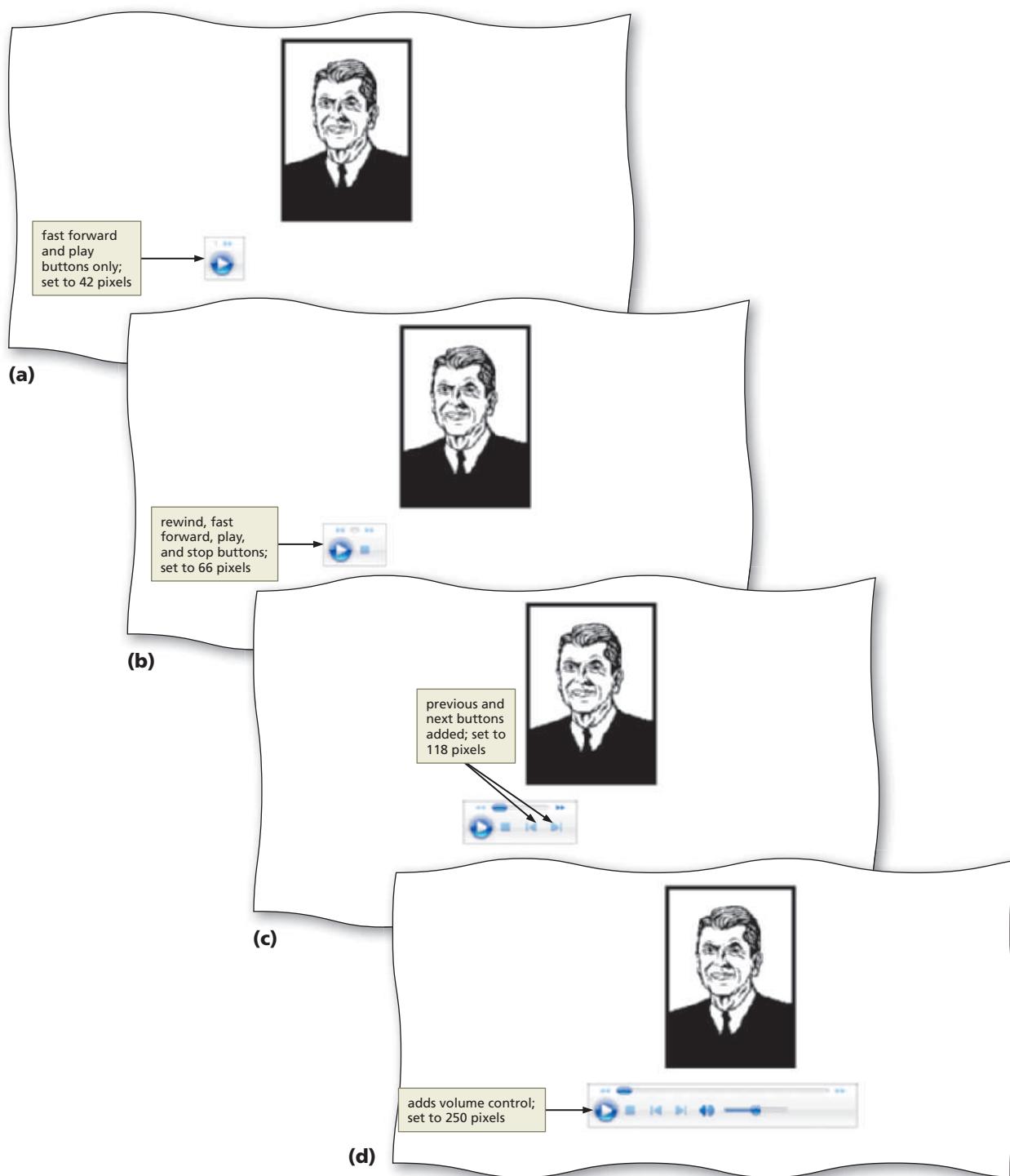
ActiveX Control	Class ID
Flash Shockwave Player	D27CDB6E-AE6D-11cf-96B8-444553540000
Java Applet	8AD9C840-044E-11D1-B3E9-00805F499D93
QuickTime Player	02BF25D5-8C17-4B23-BC80-D3488ABDDC6B
RealAudio Player	CFCDA03-8BE4-11cf-B84B-0020AFBBCCFA
Windows Media Player	6BF52A52-394A-11d3-B153-00C04F79FAA6

The object statement also includes the **height** and **width** attributes — that is, the height and width of the object in pixels. You include the height and width for the audio file to identify the view of the player control panel that you want to display. Table 8–4 shows the resulting control panel when you specify different widths. In all cases in Table 8–4, the height of the object is 45 pixels, which provides enough room to view the player control horizontally. These are not exact numbers, but suggestions for a width that displays the image in the right column. You can experiment with the width and height to see what might best fit your application.

Table 8–4 Control Panel Views for Windows Media Player

Width Setting	Resulting Control Display	Image
42 pixels	Displays the fast forward and play/pause buttons	
66 pixels	Displays the rewind, fast forward, play/pause, and stop buttons	
118 pixels	Adds the previous and next buttons	
250 pixels	Adds a volume control (full control panel)	

Figure 8–4 shows how the various width dimensions display the player control relative to the Web page. In the styles8.css external style sheet, the left margin of the class="audio" is set to 375 pixels. That means the object displays 375 pixels from the left margin, regardless of what control width is set. Figures 8–4a through 8–4d illustrate how the controls shown in Table 8–4 display, with each control starting 375 pixels from the left. Certainly, you would want to adjust the left margin in the external style sheet if you decided to use various widths of the control. As for most Web development decisions, the application purpose would dictate why you would select various widths. For instance, if you wanted to allow Web site visitors to only play and stop the audio clip and not allow them to vary the volume, you would set width to 42 pixels.

**Figure 8-4**

Parameters

Many other object parameters are available that are not discussed in this chapter. The various ActiveX controls have different parameters that can be used with that control. Review online what other parameters might be effective for you to use in your Web development.

Object Tag Parameters

Each type of player has various `<object>` tag parameters that can be used. A **parameter** identifies the behavior and appearance of the object to which you assign the parameters. Table 8–5 lists the most commonly used parameters for Windows Media Player.

Table 8–5 Commonly Used Parameters for Windows Media Player

Parameter	Default	Description
autoplay	true	Specifies whether the current media item begins playing automatically
balance	0	Specifies the current stereo balance; values range from –100 to 100
baseURL	[no default]	Specifies the base URL used for relative path resolution with URL script commands that are embedded in media items
enabled	false	Specifies whether the Windows Media Player control is enabled
fullscreen	false	Specifies whether video content is played back in full-screen mode
mute	false	Specifies if audio is muted
playcount	1	Specifies the number of times a media item will play; minimum value of one
rate	1.0	Specifies the playback rate; 0.5 equates to half the normal playback speed, 2 equates to twice
showaudiocontrols	true	Sets if the audio controls should show
showcontrols	true	Sets if the player controls should show
showdisplay	false	Sets if the display should show
showstatusbar	false	Sets if status bar should show
stretchtofit	false	Specifies whether video displayed by the control automatically sizes to fit the video window, when the video window is larger than the dimensions of the video image
uimode	full	Specifies which controls are shown in the user interface; possible values: invisible, none, mini, full
URL		Specifies the name of the media item to play; you can specify a local filename or a URL
volume	[last setting; 0–100]	Zero specifies no volume and 100 specifies full volume
windowlessvideo	false	Specifies or retrieves a value indicating whether the Windows Media Player control renders video in windowless mode; when windowlessvideo is set to true, the player control renders video directly in the client area, so you can apply special effects or layer the video with text

Although you do not add any QuickTime (QT) video clips to any Web pages in this chapter, Table 8–6 lists the most commonly used parameters for another popular player, Apple QuickTime. To see complete parameters for any player, review them online at the manufacturers' Web sites.

Table 8–6 Commonly Used Parameters for Apple QuickTime

Parameter	Default	Description
autoplay	true	Specifies whether the current media item begins playing automatically
bgcolor	[no default]	Sets the background color for the object
controller	true	Specifies if the player controls should show
endtime	[no default]	Specifies the time in the clip at which the video ends

Table 8–6 Commonly Used Parameters for Apple QuickTime (continued)

Parameter	Default	Description
loop	true	Plays the clip in a continuous loop
mute	false	Specifies if audio is muted
src	[no default]	Specifies the source of the clip
starttime	[no default]	Specifies the time in the clip at which the video begins
volume	[last setting]	Sets the initial audio volume

You add one parameter in the `<object></object>` container for the audio clip, as shown in the code below.

```
<object classid="clsid:6BF52A52-394A-11D3-B153-00C04F79FAA6"
       height="45" width="250">
    <param name="URL" value="reagan-audio.mp3" />
</object>
```

The URL parameter, as shown in Table 8–5, identifies what audio clip is to be played. In this case, you are inserting an MPEG (.mp3) file. This is a short audio clip of President Reagan speaking his famous line at the Brandenburg Gate. As soon as you open the Web page with this `<object>` statement, the `reagan-audio.mp3` audio clip starts immediately (note the `autoplay` default in Table 8–5). This Web page includes a picture of President Reagan, so it could be appropriate to start the audio immediately. As stated before, the purpose of the Web site dictates what parameters you use. Think about yourself as a Web site visitor. How do you feel when a loud audio clip plays as soon as you enter a Web site? What do you think if you have no option to pause or stop the clip? Always consider your own browsing habits, likes, and dislikes when you design Web sites. If you did not want to start the .mp3 file automatically, you would add a `<param name="autoplay" value="false" />` statement within the `<object></object>` container. With this audio clip insertion, you do provide the player control panel that gives the user the opportunity to pause or stop the audio clip.

Inserting an audio file.

You would insert an audio file into a Web page if you want your Web site visitors to be able to hear content or background music. In the case of this Web site, you will insert an audio clip of President Reagan's famous line, "Mr. Gorbachev, tear down this wall!"

- **Determine the area on the Web page into which you want to insert the audio clip.** This was a simple task for the Web site under development. There is a `reagan.html` Web page that discusses the speech made by that president. A picture of President Reagan is already inserted into the Web page. Because this is an audio clip, and only the player control bar will display, you insert the audio clip directly under the picture of the president.
- **Decide which parameters (if any) that you should use.** For this audio clip, you just want to play it with the default settings, so you will not set any parameters. The audio clip will start automatically when the Web page is opened. The Web site visitors will have the ability to pause, stop, or replay the audio clip using the default player control panel provided.

Plan Ahead

To Start Notepad++ and Open an HTML File

- 1** Start Notepad++ and, if necessary, maximize the window.
- 2** With the USB drive plugged into your computer, navigate to the Chapter08\ChapterFiles folder.
- 3** Double-click reagan.html in the list of files to open the file shown in Figure 8–5.

Q&A

What is the purpose for the “audio” class on line 28?

This is a class identified in the external style sheet, styles8.css, to which you link on line 10. If you have not already done so, you should review that external style sheet. All classes (on lines 21, 25, and 28) are created in that external style sheet.

Q&A

Why is the map container (lines 32 through 36) in this file?

Notice that line 16 contains an image file that includes the `usemap="#menubar"` statement. Line 32 identifies that map as the menubar, which is an image map used for navigation.

The screenshot shows the Notepad++ editor with the file "reagan.html" open. The code is as follows:

```

10 <link rel="stylesheet" type="text/css" href="styles8.css" />
11
12 </head>
13 <body>
14
15 <div style="text-align: center">
16   
17
18 <h1>President Ronald Reagan</h1>
19 </div>
20
21 <div class="content">
22   On June 12, 1987, the U.S. President Ronald Reagan delivered his famous "Tear down this wall!" speech at the Brandenburg Gate. The words were a challenge to Soviet leader Mikhail Gorbachev to bring down the barriers between the Eastern and Western blocs. The audio clip that plays on this page is an excerpt from that speech.
23 </div>
24
25 <div class="picture">
26   </div>
27
28 <div class="audio">
29   <!-- Insert audio statements here -->
30 </div>
31
32 <map name="menubar" id="menubar">
33   <area shape="rect" coords="83,8,137,28" href="brandenburg.html" alt="Home" />
34   <area shape="rect" coords="184,8,254,28" href="kennedy.html" alt="Kennedy" />
35   <area shape="rect" coords="295,8,360,28" href="reagan.html" alt="Reagan" />
36 </map>
37
38 </body>
39 </html>

```

Annotations in the screenshot:

- An arrow points to the line `<link rel="stylesheet" type="text/css" href="styles8.css" />` with the text "link to external style sheet".
- An arrow points to the line `<div class="content">` with the text "classes from external style sheet".
- An arrow points to the line `<!-- Insert audio statements here -->` with the text "comment statement on line 29".
- A brace groups the three `<area>` tags under the `<map>` tag, with the text "image map statements" pointing to it.

Figure 8–5

To Add an Audio Clip to a Web Page

The reagan.html Web page file is complete except for the <object> statement which will be added next. Table 8–7 shows the HTML code for this statement.

Table 8–7 HTML Code to Add Audio Clip Object Statement

Line	HTML Code
29	<object classid="clsid:6BF52A52-394A-11D3-B153-00C04F79FAA6" height="45" width="250">
30	
31	<param name="URL" value="reagan-audio.mp3" />
32	
33	</object>

The following step shows how to add an audio clip to the President Reagan Web page.

1

- Highlight the words
`<!-- Insert audio

statements here -->
on line 29.
- Enter the HTML
code in Table 8–7,
indenting as shown.
- Click File on the
menu bar and then
click Save to save
the reagan.html file
(Figure 8–6).

Q&A

How will I know
what classid to use
for various ActiveX
controls?

Table 8–3 on page
HTML 370 lists the
classids for the most
commonly used
ActiveX controls. If
there are other con-
trols that you use, you
can find their classids
online on the manufac-
turer’s Web site or by doing a Web search for that information.

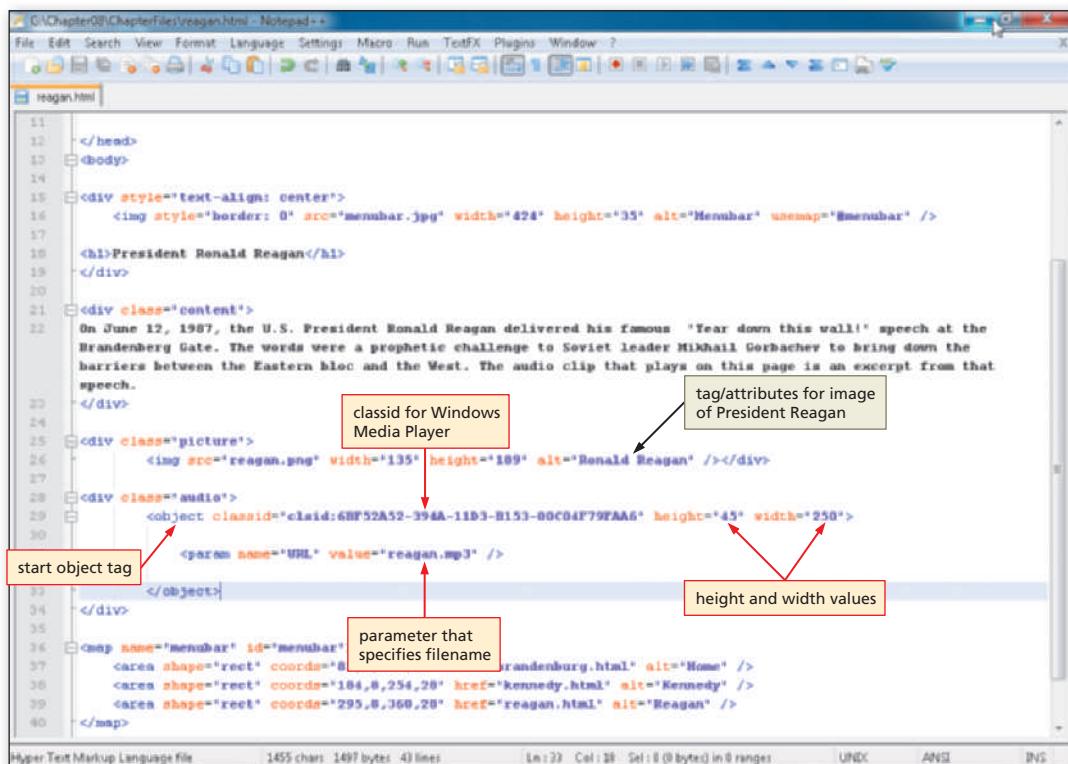


Figure 8–6

Experiment

- Change the width and height attribute values and save the file with each change. Open the Web page in the browser to see how the changes affect the display of the control panel.

To Validate and View a Web Page Using ActiveX Controls

As always, you should validate and then view your pages. When you try to link to the Reagan Web page, it is likely that your browser will block the page's ActiveX content and display a security notification. Internet Explorer sets security to high by default. This prohibits ActiveX controls from running without your intervention. The following steps illustrate how to validate the Web page, and then respond to the security notification in Internet Explorer to view the Web page.

1

- Validate the `reagan.html` Web page using the W3C validation service.
- With the USB drive plugged in to your computer, navigate to the `Chapter08\ChapterFiles` folder and open the `brandenburg.html` file in the Web browser to show the home page of this Web site. Click the `Reagan` link to navigate to the newly saved file.
- Because of Internet Explorer's high level of security, you get a notification in the gold bar at the top of the browser, as shown in Figure 8–7.

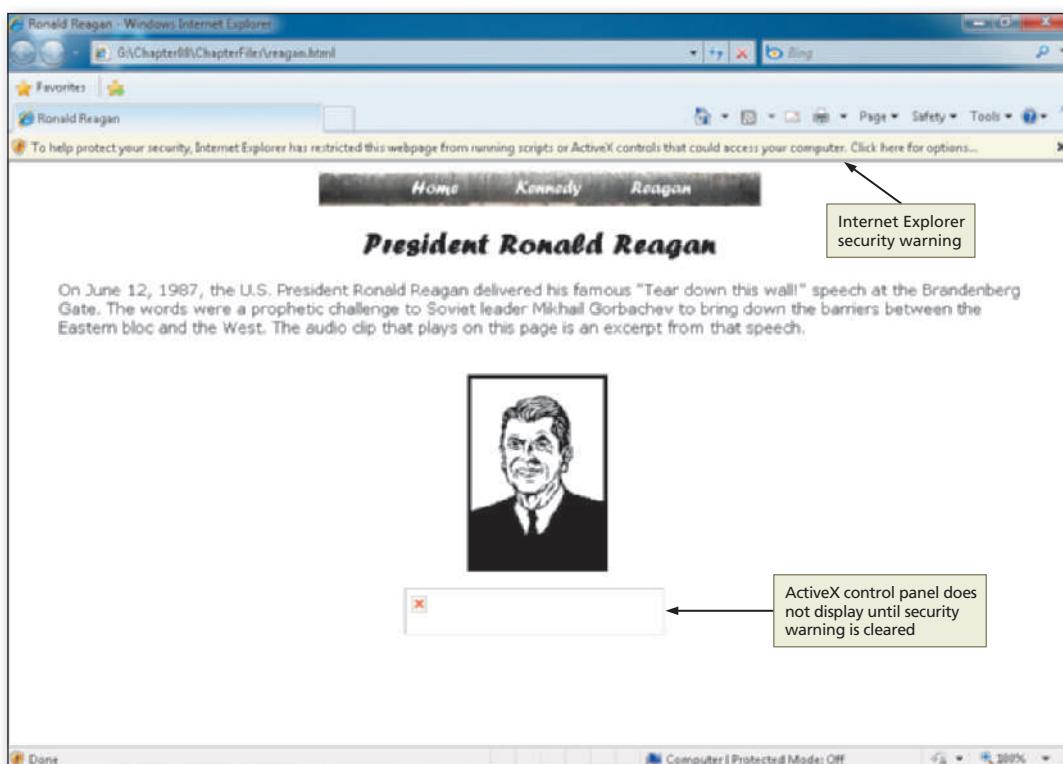


Figure 8–7

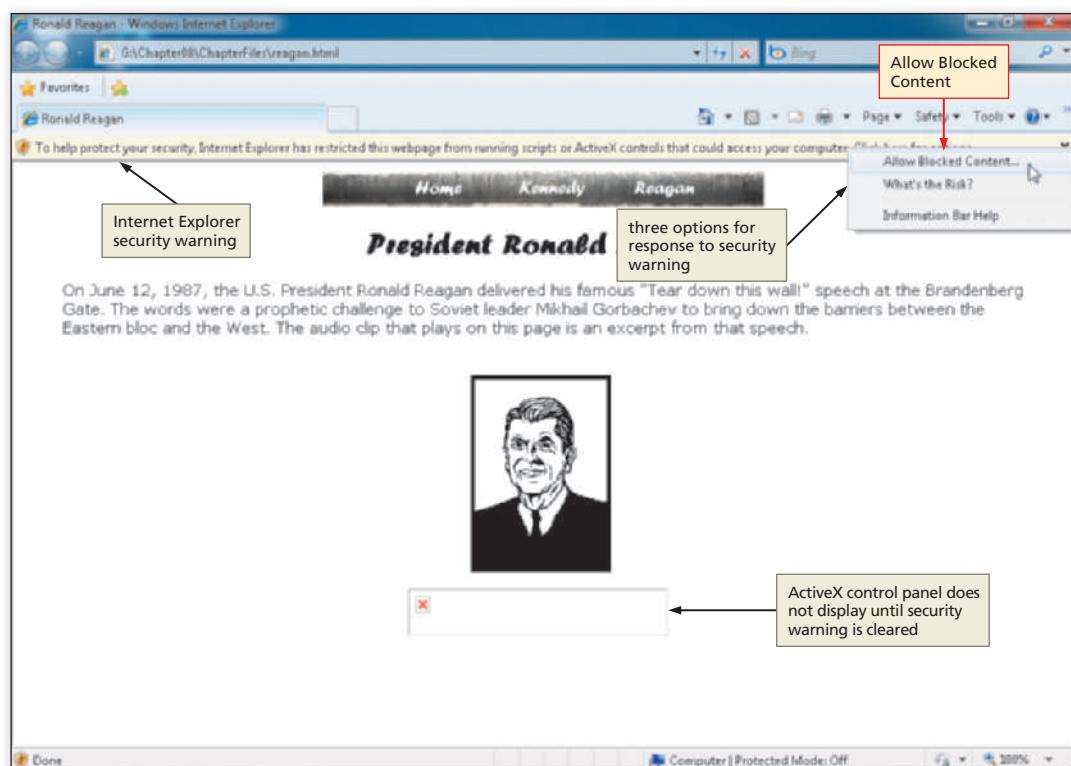
Q&A

Will the ActiveX controls and security notification appear if I use a browser other than Internet Explorer?

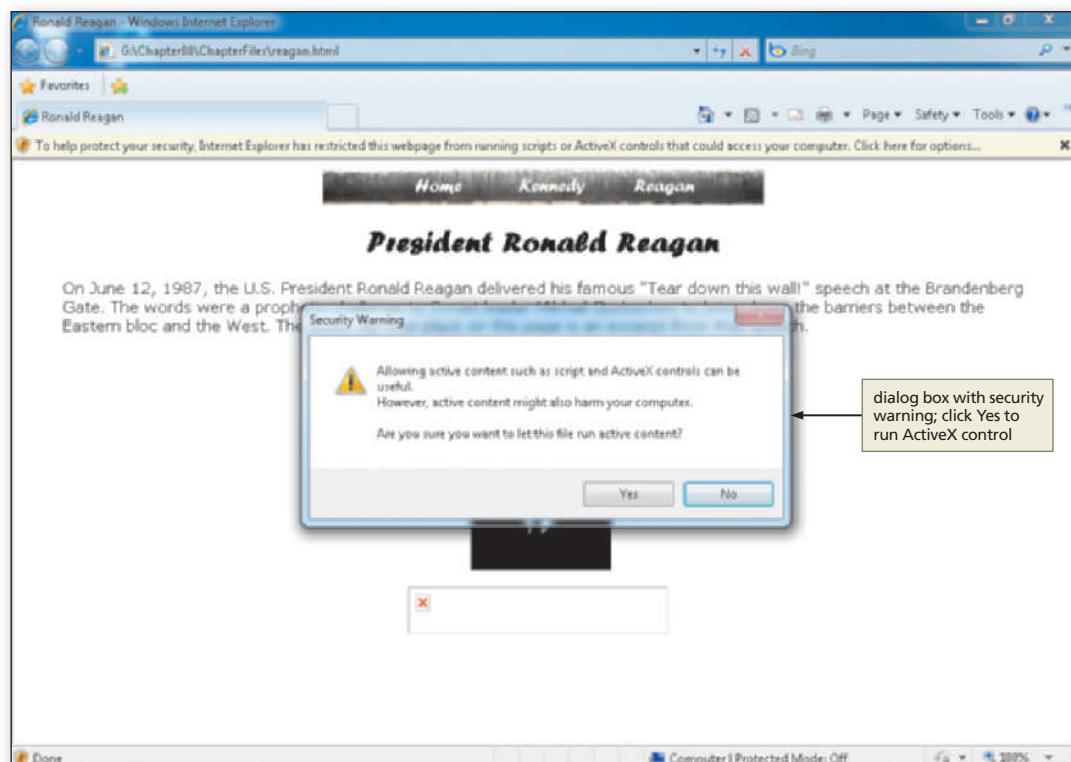
No. ActiveX controls only display in the Internet Explorer Web browser. If you display this page in another browser, the multimedia control panel will not appear and the audio file will not play.

2

- Click anywhere in the gold bar at the top of the browser window to view the options to view the options (Figure 8–8).

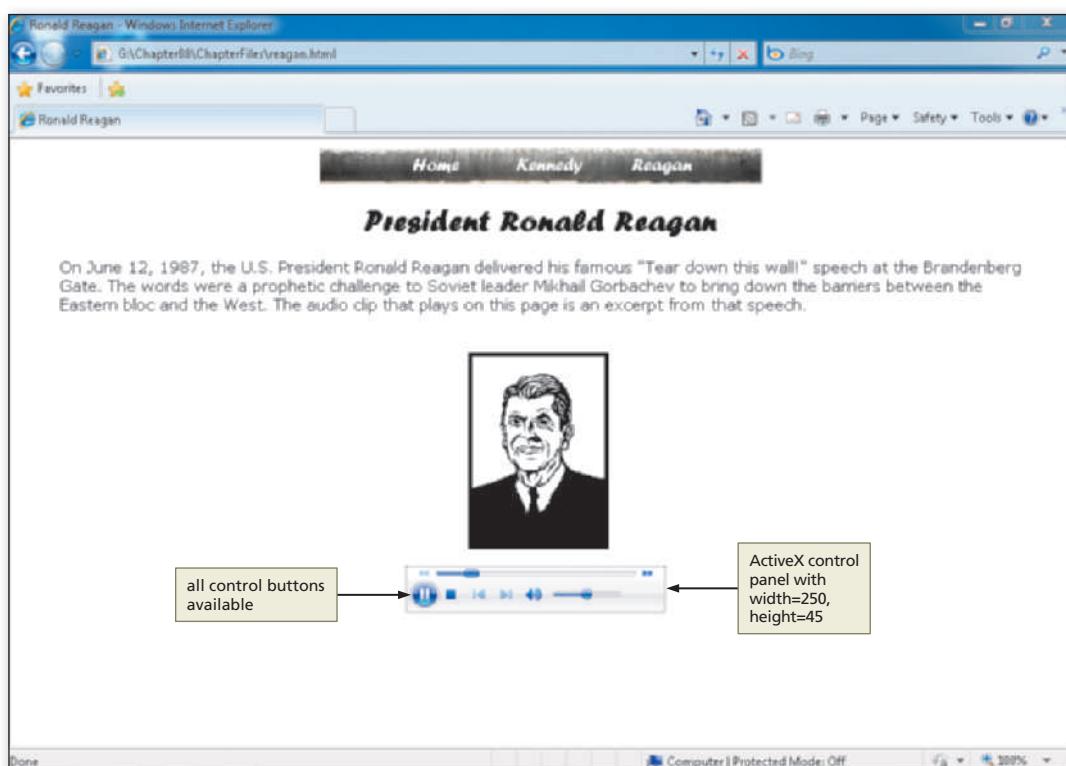
**Figure 8–8****3**

- Click the Allow Blocked Content option to display the dialog box shown in Figure 8–9.

**Figure 8–9**

4

- Click Yes in the Security Warning dialog box to display the completed Web page with audio controls shown in Figure 8–10.

**Figure 8–10**

Q&A Do I have to respond to the security questions for each page in a Web site?

No, you are asked to respond just once within a given Web site. For example, if you were to open the kennedy.html Web page next (once the video clip is inserted), you would not be asked those security questions.

Q&A Can I change the security level in IE?

Yes, but you need to consider what that means. If you disable the high security in IE, then you are allowing all ActiveX controls to run automatically. It might be best to leave the security as it is and respond to the prompts as shown here.

To Print a Web Page and an HTML File

- Print the Reagan Web page from the browser.
- Click the Notepad++ button on the taskbar.
- Print the Reagan.html file from Notepad++.

Creating a video clip.

Video clips can add valuable content to a Web page. There are many uses for video clips. Companies use clips for training or to convey messages from the CEO. Instructors use videos to help students better understand complex topics in online classes. Many people use video clips just for fun on their personal Web sites or on YouTube.

- **Utilize video clips that enhance the content of the Web page.** In the chapter project, a video clip of a live speech adds value to the Web page.
- **Determine whether to utilize the autostart parameter.** Sometimes you want to force the user to read some material on the Web page first and then view the corresponding video clip. In this case, you want to turn autostart off and force the user to click play in order to watch the video.
- **Use uimode and screen stretching parameters as needed.** There are many ways to show a video. Sometimes you want the video to be a small part of a Web page; at other times, you want it to be the entire message or full screen. Sometimes you want a control panel to display; at other times you need no control panel. You need to determine what values to use with the various parameters that add to the viewing experience.

Plan Ahead

Adding a Video Clip to a Web Page

The next step is to add a video file, kennedy-video.wmv, to the kennedy.html file. You use the `<object></object>` container for this, just as you did for the audio clip. You continue to use the Windows Media Player (WMP) ActiveX control, so the classid is the same. You add different parameters for this video insertion.

The following section of HTML code shows the statements that you add for this video clip addition.

```
<object classid="clsid:6BF52A52-394A-11D3-B153-00C04F79FAA6">
    <param name="URL" value="kennedy-video.wmv" />
    <param name="autostart" value="false" />
    <param name="uimode" value="full" />
    <param name="stretchtofit" value="true" />
</object>
```

Note that the classid is specific to the Windows Media Player ActiveX control (see Table 8–3 on page HTML 370). The parameters can be found in Table 8–5 on page HTML 372. You do not enter the width and height attributes this time so that you can see what this movie clip looks like at its default size. You can, however, add width and height attributes in the `<object>` tag. This is something that you can experiment with. Because you insert the stretchtofit parameter within the `<object></object>` container, the movie clip will stretch regardless of the sizes that you specify in the width and height attributes. Remember that you have a link to an external style sheet, styles8.css, so if you do change the width and height, you will probably need to change the left margin dimension as well.

The four parameters used for the video clip insertion are: URL, autostart, uimode, and stretchtofit. The URL parameter is used in the same way for a video clip as it was for an audio clip. It tells the name of the object that you want to insert. The default for the autostart parameter is "true", which means that a clip will automatically start unless you set autostart to "false" by entering `<param name="autostart" value="false" />` as the command. With autostart turned off, your Web site visitor has to click the play button on the ActiveX control panel to start the video clip. Just as with other decisions that are made

BTW Video Clips

Video clips can have a large file size depending on the length of the clip together with the quality of the clip. Search for information about rules-of-thumb for file sizes when video clips are incorporated into a Web site.

during the analysis phase of the Web development life cycle, you have to decide what is best for your application. Do you want the audio or video clip to start as soon as the Web page is opened, or is it best to force your user to start them?

The next parameter statement <param name="uimode" value="full" /> specifies which controls are shown. For this project, the uimode is set to full. Figure 8–11a shows the media player window when the uimode is set to full. Figure 8–11b shows the Web page when uimode is set to none. Note that this is an unusable situation. You have set the uimode to none together with no autoplay. Your Web site visitor therefore has no access to the controls to start the video. The video is also not going to autoplay because of the autoplay="false" statement. Figure 8–11c shows the Web page with uimode set to mini. Compare the Web page in Figure 8–11c to Figure 8–11a. Although users have access to some controls, they do not have access to all of the controls. Finally, Figure 8–11d is the Web page displayed when uimode is set to invisible. Again, because you have the autoplay set to false, you display nothing and you also see and hear nothing. For video clips, this is unusable. For audio clips, this might be effective. If you wanted to hear background sounds but not see the controls, you could set an invisible uimode along with autoplay set to true (the default).

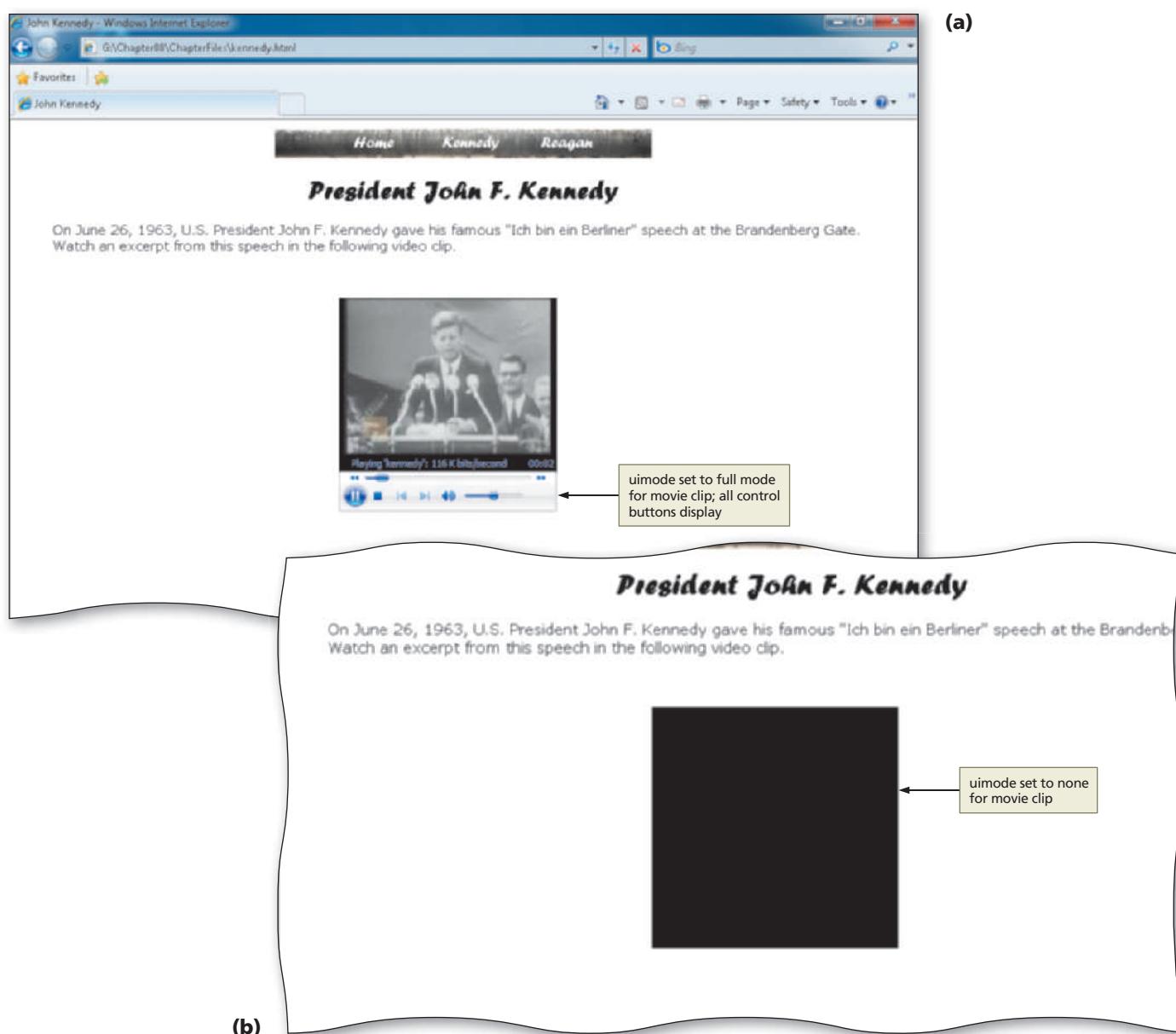
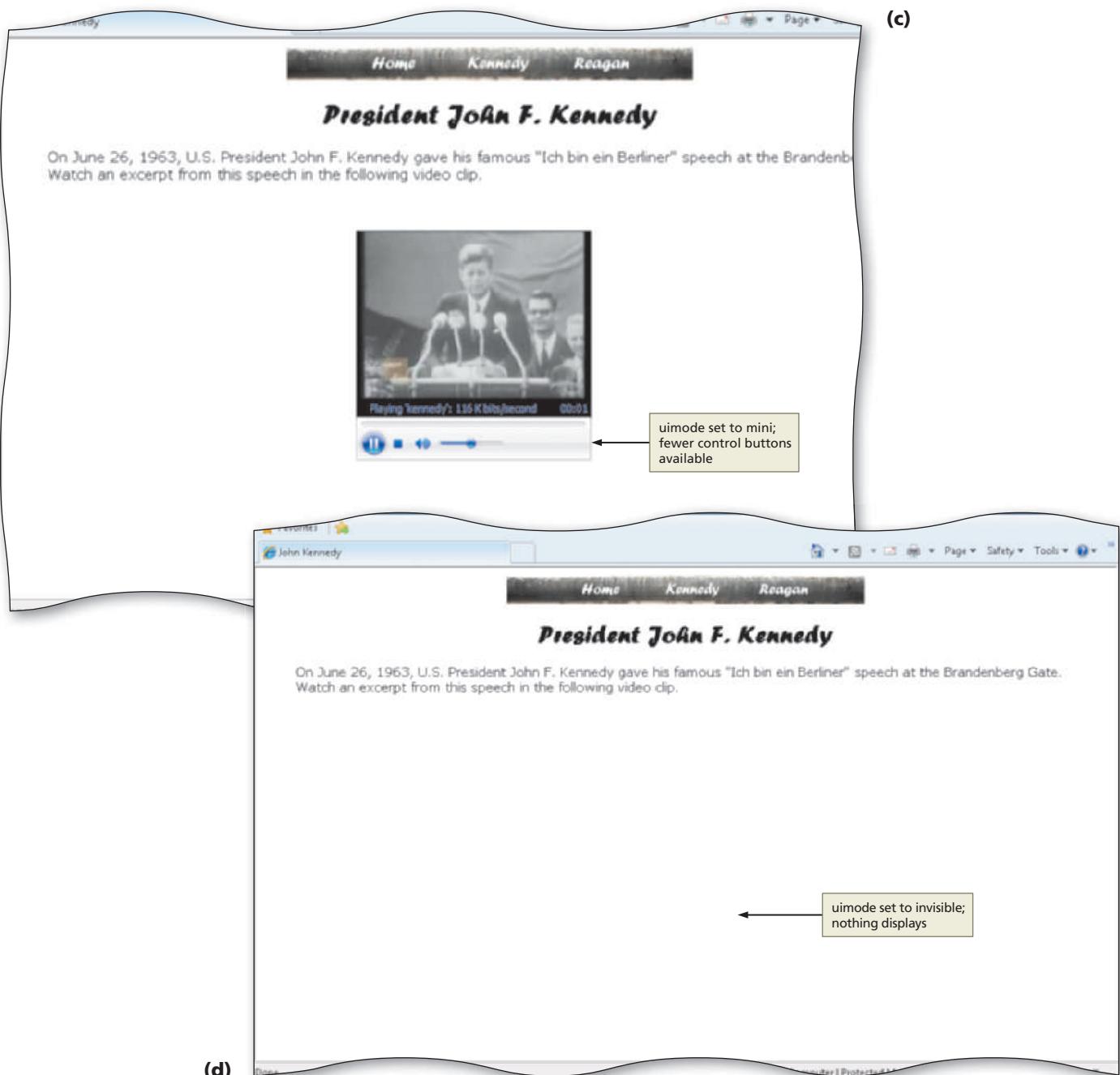
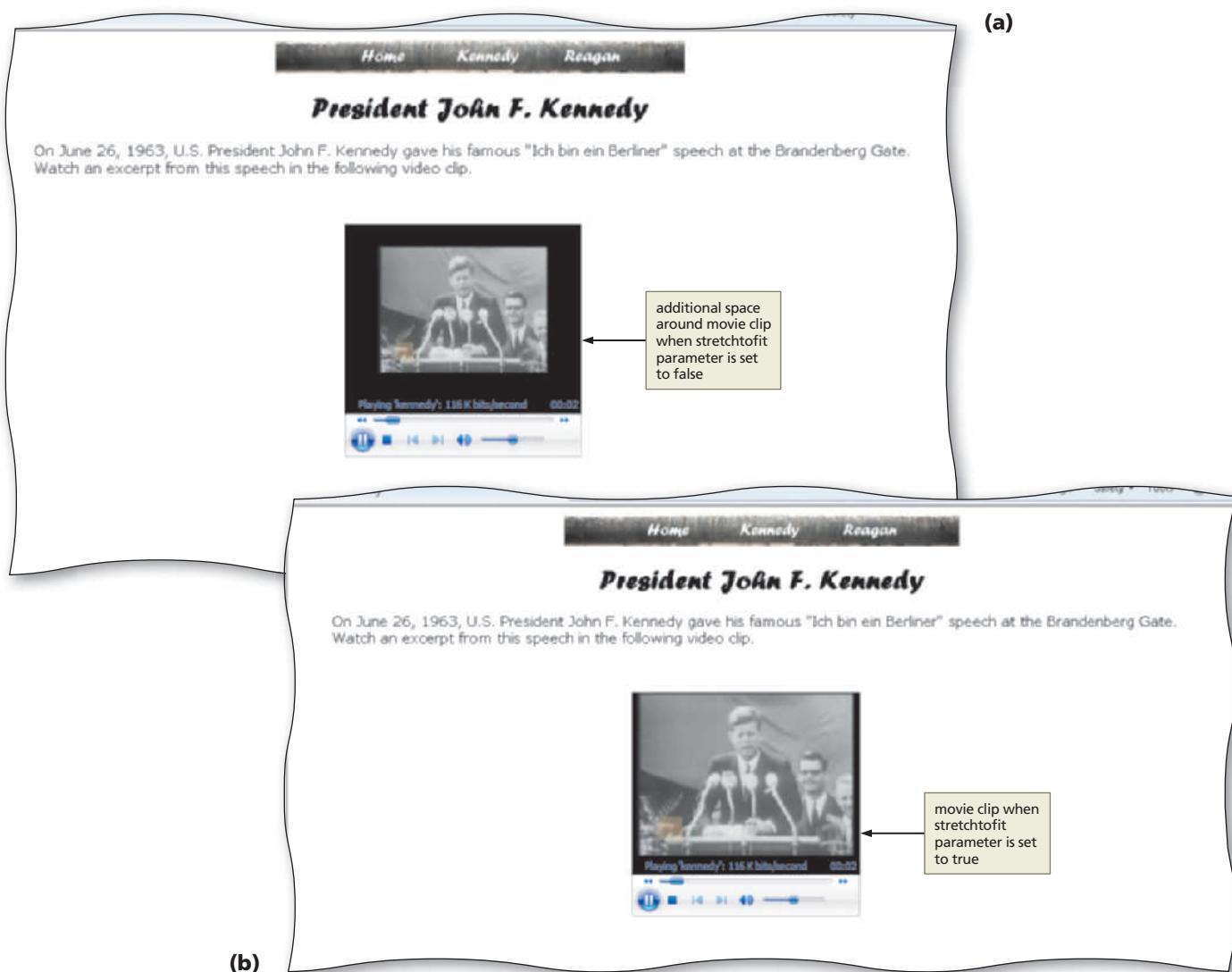


Figure 8-11

**Figure 8-11 (continued)**

The last parameter used in this container is `stretchtofit`. The `stretchtofit` parameter sizes the video to fit the video window, when the video window is larger than the dimensions of the video image. In the case of this video, it is smaller than the video window in Windows Media Player. The default for this option is false, so the effect on the Web page looks like the video clip shown in Figure 8–12a on the next page. Figure 8–12b shows the Web page with `stretchtofit` set to true.

**Figure 8-12**

To Add a Video Clip to a Web Page

To add the video file, kennedy-video.wmv, to the kennedy.html file, you use the `<object></object>` container, just as you did for the audio clip, but with different parameters. Table 8-8 shows the code to add the video clip.

Table 8-8 HTML Code to Add Video Clip

Line	HTML Code
26	<code><object classid="clsid:6BF52A52-394A-11D3-B153-00C04F79FAA6"></code>
27	
28	<code><param name="URL" value="kennedy-video.wmv" /></code>
29	<code><param name="autostart" value="false" /></code>
30	<code><param name="uimode" value="full" /></code>
31	<code><param name="stretchtofit" value="true" /></code>
32	
33	<code></object></code>

1

- If necessary, click the Notepad++ button on the taskbar.
- With the USB drive plugged in to your computer, open the kennedy.html file from the Chapter08\ChapterFiles folder (Figure 8–13).

```

5 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
6   <head>
7     <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
8     <title>John Kennedy</title>
9
10    <link rel="stylesheet" type="text/css" href="styles8.css" />
11
12  </head>
13  <body>
14
15    <div style="text-align: center">
16      
17
18      <h1>President John F. Kennedy</h1>
19    </div>
20
21    <div class="content">
22      On June 26, 1963, U.S. President John F. Kennedy gave his famous "Ich bin ein Berliner" speech at the Brandenburg Gate. Watch an excerpt from this speech in the following video clip.
23    </div>
24
25    <div class="movie">
26      <!-- Insert video statements here -->
27    </div>
28
29    <map name="menubar" id="menubar">
30      <area shape="rect" coords="83,0,137,28" href="brandenburg.html" alt="Home" />
31      <area shape="rect" coords="184,0,254,28" href="kennedy.html" alt="Kennedy" />
32      <area shape="rect" coords="293,0,360,28" href="reagan.html" alt="Reagan" />
33    </map>
34
35  </body>
36 </html>

```

comment on line 26
where object tag is inserted

Figure 8–13**2**

- Highlight the words <!—Insert the video statements here --> on line 26.
- Enter the HTML code shown in Table 8–8 (Figure 8–14).
- Save the file.

Q&A

What determines whether or not you should automatically start the video clip when the Web page is opened?

Your specific Web page purpose determines this option. If you want your Web site visitors to read some text beforehand, as is the case here, you should set `autoplay` to false, forcing visitors to click the play button when they want to view the video.

Q&A

I like leaving a lot of space around the video clip. Do I have to stretch the video to fit the screen?

No. The default is to not stretch the video; that is why you had to turn that off with the `stretchtofit="true"` parameter. This is something with which you should experiment.

**Experiment**

- Add the `width` and `height` attributes to the `<object>` statement. Change the values of the `width` and `height` and save the file with each change. Open the Web page in the browser to see how the changes affect the display of the video. Remove the `width` and `height` attributes.

```

21 <div class="content">
22   On June 26, 1963, U.S. President John F. Kennedy gave his famous "Ich bin ein Berliner" speech at the Brandenburg Gate. Watch an excerpt from this speech in the following video clip.
23 </div>
24
25 <div class="movie">
26   <object classid="clsid:6BF52A52-394A-11D3-B153-00C04F79FAA6">
27     <param name="URL" value="kennedy-video.wmv" />
28     <param name="autoplay" value="false" />
29     <param name="uiMode" value="full" />
30     <param name="stretchtofit" value="true" />
31   </object>
32 </div>
33
34 <map name="menubar" id="menubar">
35   <area shape="rect" coords="83,0,137,28" href="brandenburg.html" alt="Home" />
36   <area shape="rect" coords="184,0,254,28" href="kennedy.html" alt="Kennedy" />
37   <area shape="rect" coords="293,0,360,28" href="reagan.html" alt="Reagan" />
38 </map>
39
40 </body>
41 </html>

```

classid for Windows Media Player

URL parameter identifies file name to insert

uiMode parameter set to full; full set of video controls will be shown

autoplay parameter set to false; user must click play button to start movie clip

stretchtofit parameter set to true; video will be stretched to fit player

Figure 8–14

To Validate and View a Web Page

After you have added the video clip to the kennedy.html Web page file, you need to validate the file and view the Web page to review the style changes.

- 1** Validate the Web page using the W3C validation service.
- 2** Return to the Web site home page and click the navigation link for the kennedy.html file in the browser to view the changes (Figure 8–15).

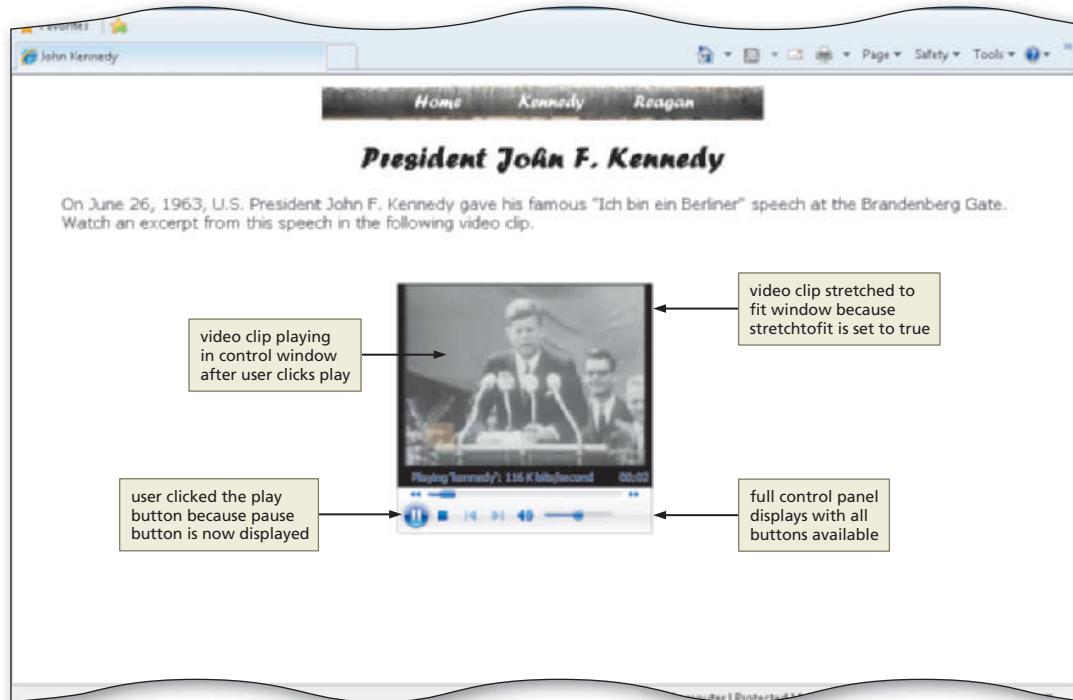


Figure 8–15

BTW

Quick Reference

For a list of HTML tags and attributes, see the HTML Quick Reference (Appendix A) at the back of this book, or visit the HTML Quick Reference Web page (scsite.com/HTML6e/qr).

To Print a Web Page and HTML File

- 1** Print the Kennedy Web page from the browser.
- 2** Print the Notepad++ kennedy.html file.

Chapter Summary

In this chapter, you have learned how to embed audio and video files in your Web pages. The items listed below include all the new HTML skills you have learned in this chapter.

- | | |
|---|--|
| 1. Add an Audio Clip to a Web Page (HTML 375) | 3. Add a Video Clip to a Web Page (HTML 382) |
| 2. Validate and View a Web Page Using ActiveX Controls (HTML 376) | |

Learn It Online

Test your knowledge of chapter content and key terms.

Instructions: To complete the Learn It Online exercises, start your browser, click the Address bar, and then enter the Web address scsite.com/html6e/learn. When the HTML Learn It Online page is displayed, click the link for the exercise you want to complete and read the instructions.

Chapter Reinforcement TF, MC, and SA

A series of true/false, multiple choice, and short answer questions that test your knowledge of the chapter content.

Flash Cards

An interactive learning environment where you identify chapter key terms associated with displayed definitions.

Practice Test

A series of multiple choice questions that test your knowledge of chapter content and key terms.

Who Wants To Be a Computer Genius?

An interactive game that challenges your knowledge of chapter content in the style of a television quiz show.

Wheel of Terms

An interactive game that challenges your knowledge of chapter key terms in the style of the television show *Wheel of Fortune*.

Crossword Puzzle Challenge

A crossword puzzle that challenges your knowledge of key terms presented in the chapter.

Apply Your Knowledge

Reinforce the skills and apply the concepts you learned in this chapter.

Adding a Background Sound

Instructions: Start Notepad++ and a browser. Using your browser, open the apply8-1.html file from the Chapter08\Apply folder of the Data Files for Students. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files. Figure 8–16 shows the Apply Your Knowledge Web page as it should appear in the browser after the necessary code is added.

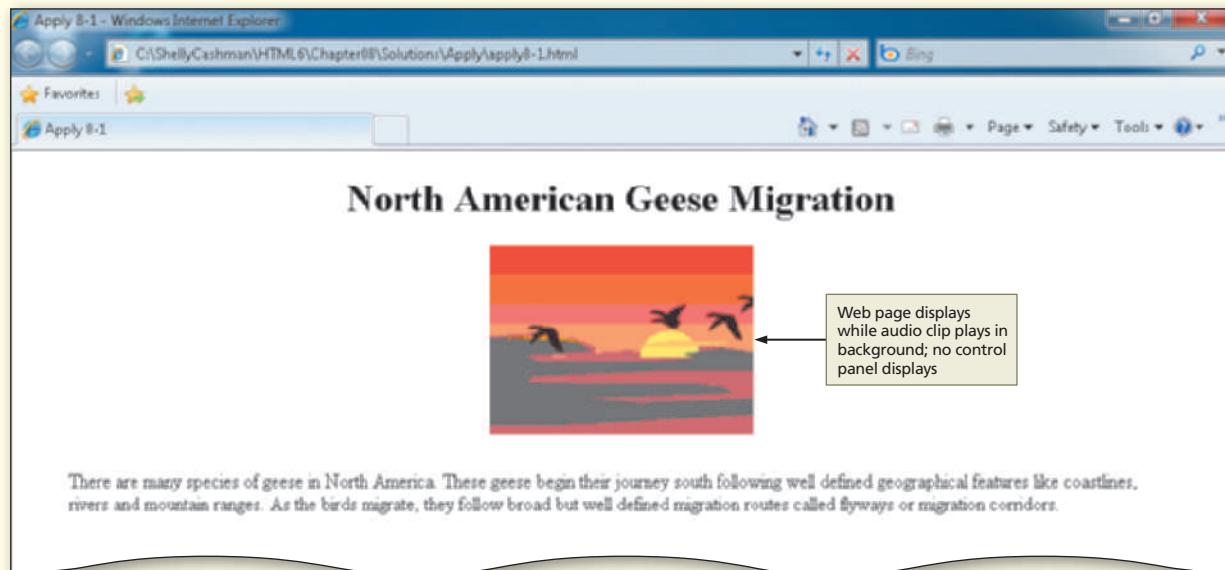


Figure 8–16

Continued >

Apply Your Knowledge *continued*

Perform the following tasks:

1. With the apply8-1.html file open in Notepad++, add the 04geeseflyhonk.mp3 audio clip with the following options:
 - a. Insert the audio clip in such a way that no control panel displays. (*Hint:* Refer to Table 8–4 on page HTML 370.)
 - b. Repeat the audio clip three times. (*Hint:* Refer to Table 8–5 on page HTML 372.)
2. Save the file as apply8-1solution.html. Validate the code and print the file.
3. Submit the solution in the format specified by your instructor.

Extend Your Knowledge

Extend the skills you learned in this chapter and experiment with new skills.

Combining Audio and Video Files

Instructions: Start Notepad++. Open the file extend8-1.html from the Chapter08\Extend folder of the Data Files for Students. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files. Save the file as extend8-1solution.html. This file contains the Web page shown in Figure 8–17.

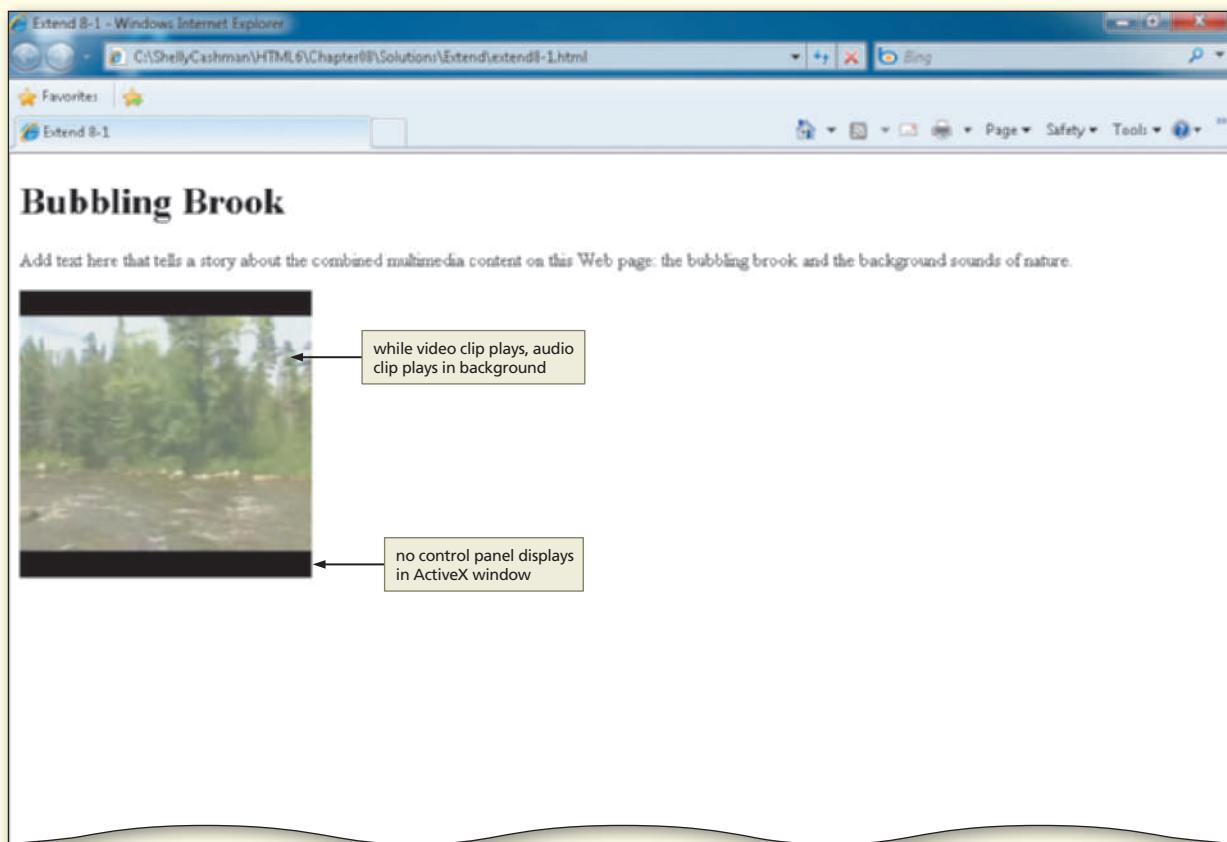


Figure 8–17

Perform the following tasks:

1. With the extend8-1solution.html file open in Notepad++, add the bkgrndnature.mp3 audio clip with the following options:
 - a. Insert the audio clip in such a way that no control panel displays.
 - b. Do not display the audio clip.
2. Add the video clip brook.wmv with the following options:
 - a. Stretch the video to fit the window.
 - b. Do not display the control panel.
3. Save the file; validate the code; print the file.
4. Submit the solution in the format specified by your instructor.

Make It Right

Analyze a document, and correct all errors and/or improve the design.

Correcting the Dog Tricks Web Page

Instructions: Start Notepad++. Open the file makeitright8-1.html from the Chapter08\MakeItRight folder of the Data Files for Students and save it as makeitright8-1solution.html. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files. The Web page is a modified version of what you see in Figure 8–18, but it contains some errors. The dogtricks.wmv video should not autostart and should display in the default size with the control panel. Make the necessary corrections to the Web page to make it look like the figure.

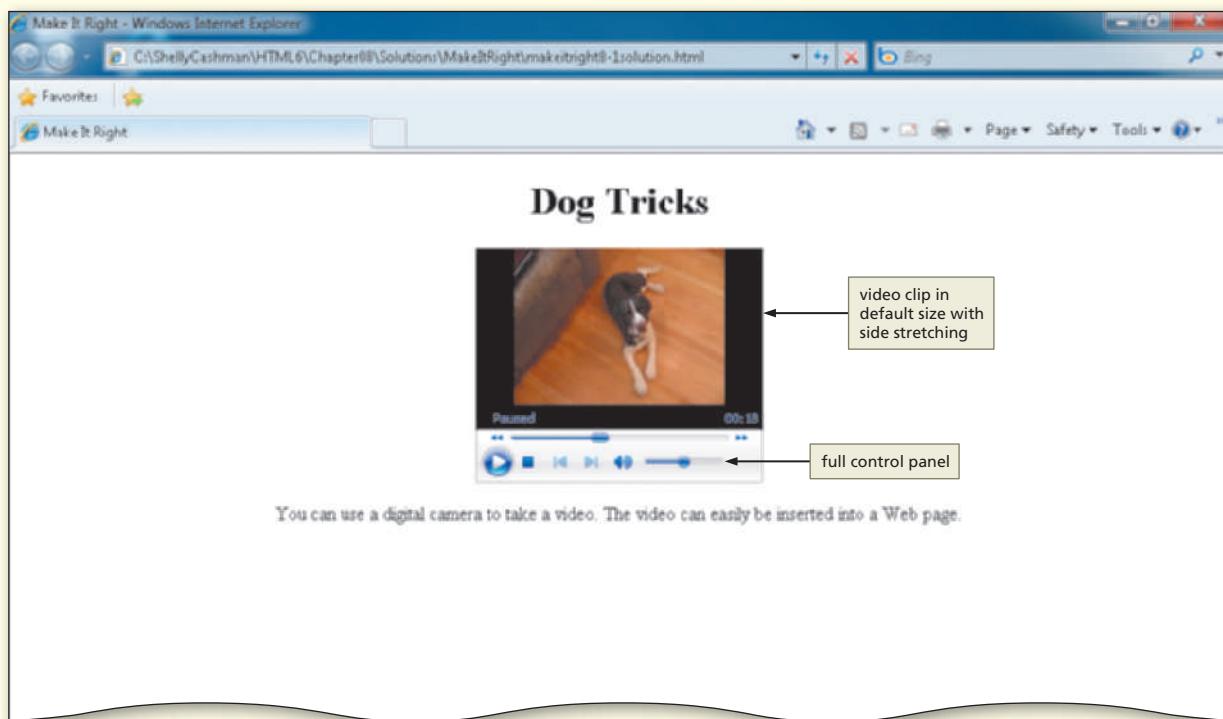


Figure 8–18

In the Lab

Lab 1: Creating Beautiful Music

Problem: Your piano teacher would like to see an example of how an audio clip of piano playing would look on a Web page. She wants to be able to pause, stop, and replay the audio clip. She also wants to see visual images of the sound waves as the clip is playing, as shown in Figure 8–19.

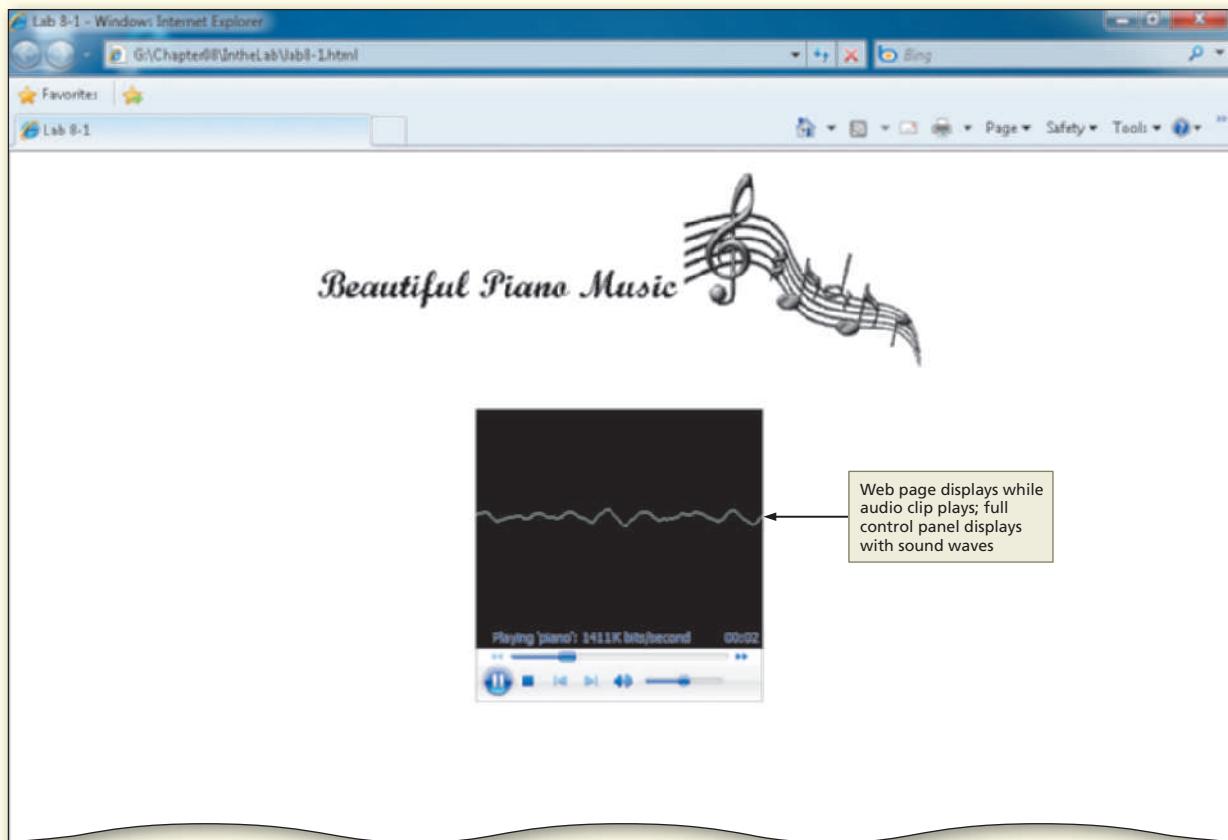


Figure 8–19

Instructions: Perform the following steps.

1. Using Notepad++, open the HTML file lab8-1.html in the Chapter08\IntheLab folder of the Data Files for Students.
2. Add the audio clip, piano.wav, to the Web page.
3. Do not allow this video to start automatically.
4. Add the parameter needed to display this audio clip so that you can see the sound waves as the music plays, as shown in Figure 8–19. (*Hint:* Review the uimode parameter.)
5. Add the parameter needed to display the control panel for the ActiveX control.
6. Save the HTML file in the Chapter08\IntheLab folder as lab8-1solution.html. Validate the file and then print it.
7. Test the Web page in Internet Explorer.
8. Submit the solution in the format specified by your instructor.

In the Lab

Lab 2: Video on a Full Screen

Problem: Your biology class is learning about bird migration. You have read about how some researchers are attempting to reintroduce whooping cranes in the U.S. and help whooping cranes raised in captivity participate in migration. You have found a video clip that shows these efforts in progress. You decide to create a Web page that will show this video. You want the video to display on the full screen of the monitor, as shown in Figure 8–20.



Figure 8–20

Instructions: Perform the following steps.

1. Using Notepad++, open the HTML file lab8-2.html in the Chapter08\IntheLab folder of the Data Files for Students.
2. Add the video clip whooper.wmv to the Web page.
3. Do not allow this video to start automatically.
4. Add the parameters needed to display this video on a full screen with the full control panel, as shown in Figure 8–20.
5. Save the HTML file in the Chapter08\IntheLab folder as lab8-2solution.html. Validate the file and then print it.
6. Test the Web page in Internet Explorer.
7. Submit the solution in the format specified by your instructor.

In the Lab

Lab 3: Adding Audio and Video Clips That Play Simultaneously

Problem: You want to create a Web page for your magic show that uses audio and video that come from two different sources, as shown in Figure 8–21. With the Kennedy video clip, the audio and video are together in one file. Using two media files (magicshow.wmv and magicmusic.mp3) gives the same effect (audio + video) with two separate files. As an example, you might use this technique when you shoot a video yourself but want to combine it with appropriate audio that you either create or find. This lab shows you how to combine separate audio and video files to play at the same time. The file lab8-3.html is a file that is complete other than the statements needed to insert the two multimedia objects. The file lab8-3.html is included in the Chapter08\IntheLab folder of the Data Files for Students.

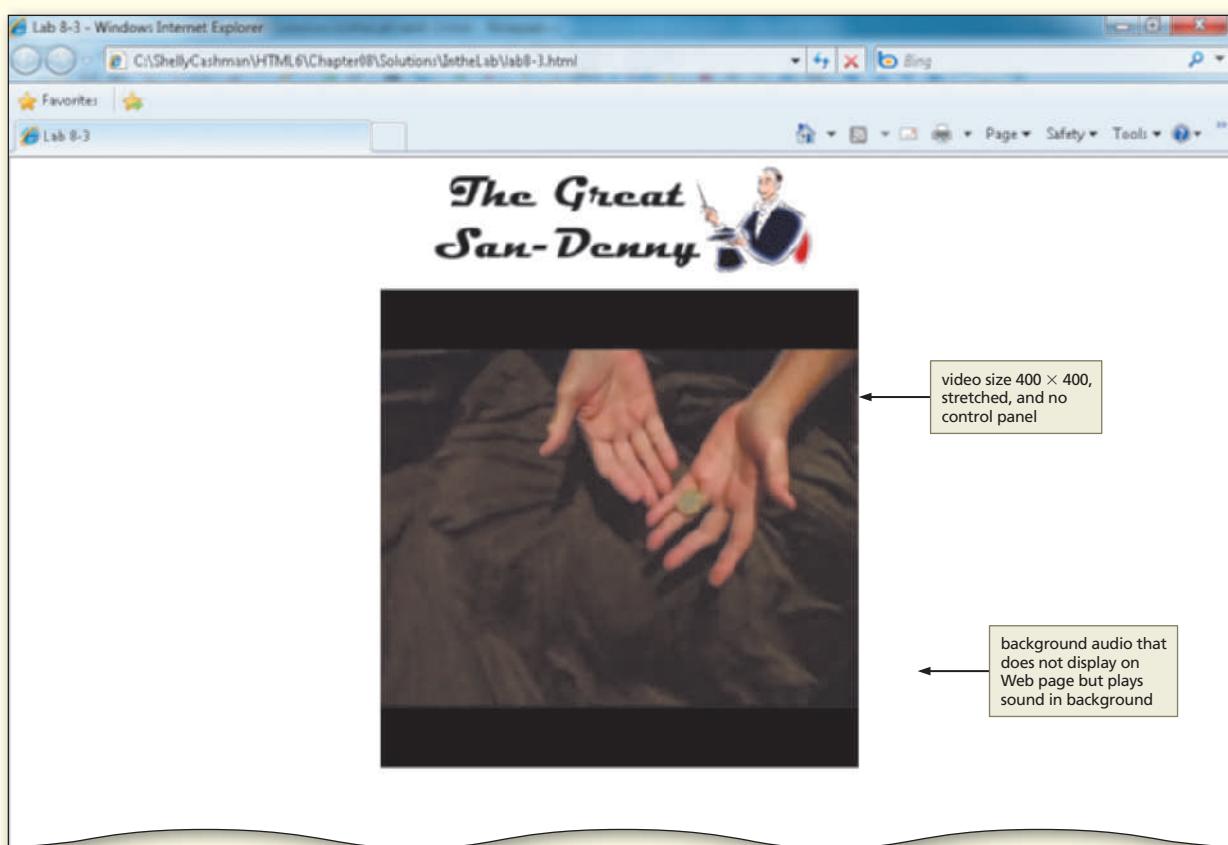


Figure 8–21

Instructions: Perform the following steps.

1. Open the file lab8-3.html in Notepad++. Save the file as lab8-3solution.html.
2. Insert the magicshow.wmv file as the video. Make sure it fills the Windows Media Player window as much as possible, but not the full monitor screen. The video should start automatically. There should not be any control panel on the WMP window, as shown in Figure 8–21.
3. Insert the magicmusic.mp3 file. This audio clip should not display at all, but it should start automatically.
4. Save the file, validate it, and then print it.

5. Test the Web page in Internet Explorer.
6. Submit the solution in the format specified by your instructor.
7. Bonus: If you have downloaded video editing software, try to combine these two files into one video file. How would this change the code that you inserted in Steps 2 and 3?

Cases and Places

Apply your creative thinking and problem-solving skills to design and implement a solution.

• EASIER •• MORE DIFFICULT

• 1: Finding Multimedia Information Online

Search on the Internet for an ActiveX control other than those discussed in this chapter (see Table 8–3 on page HTML 370 for ideas). Why would you use Flash in lieu of Windows Media Player or QuickTime? Is the functionality of Java Applets different from or similar to the functionality of Windows Media Player or QuickTime? How does RealAudio differ from its competition? Determine when you would use ActiveX controls other than the ones used in the exercises you have completed in this chapter.

• 2: Other Multimedia Formats

Review the audio and video formats listed in Tables 8–1 and 8–2 on pages HTML 368 and HTML 369. In this chapter, you used only two types of files: .mp3 and .wmv. What are the benefits of these other formats? How are these formats better or worse than the formats used in the chapter? Why would you use one format over the other? How prevalent is one media format over the others? Search online for free audio or video clips in at least three of the other formats.

•• 3: Movie-Making Software

Search for software that you can use to make or edit movie and audio clips. Most software that can be used to edit or make movies can also be used with audio files. Find software that has a free trial period. Download that software and play with it, using the video and audio clips provided in the Data Files for Students. Make sure to copy the file that you plan to use and save the original, just in case something happens. As you work with the software, use the Help utility if you have any problems. Demonstrate the software to the class.

•• 4: Adding to the Chapter 8 Web Site

Make It Personal

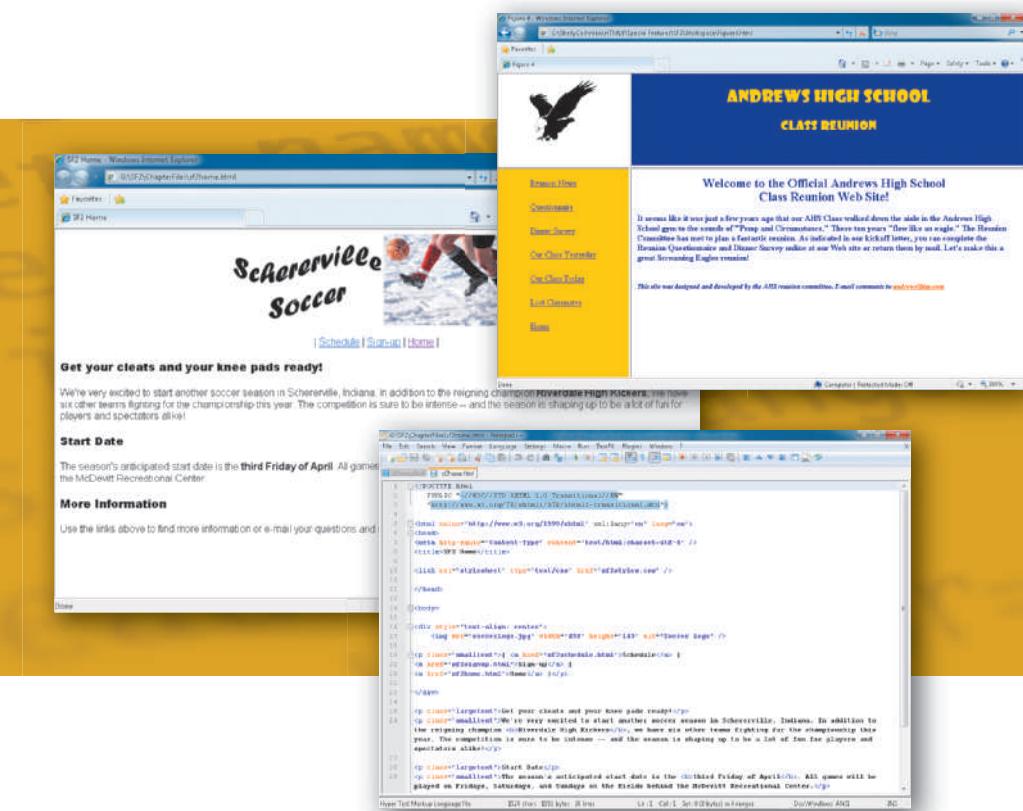
Who is your favorite U.S. President? Has that president ever given a speech or visited the Berlin Wall? If you cannot find information on that president speaking there, what speech is that president famous for? Find an audio or video clip of that U.S. President and add the clip to the Chapter 8 Web site. If the president served at a time prior to the availability of audio or video clips, what else can you look for? Is there a “reenactment” clip of someone dressed as that president speaking? Can you record the audio for a famous speech and use a picture of the president? What will you have to do to the image map in order to accommodate a new clip? How can you restructure the Web site if you use a speech other than one at the Berlin Wall?

Cases and Places *continued***•• 5: Creating a Video****Working Together**

Work with your team to analyze and design a video that you can use in a Web site of your creation. You might want to storyboard the scenes (i.e., to graphically organize a process with a series of illustrations or images displayed in sequence for the purpose of visualizing the motion) so that it is fully designed. Use a digital camera to film the video. Using the software evaluated in Cases and Places 3, edit the video so that it is a reasonable file size (if necessary). Create a Web site in which this video can be inserted. Share your Web site and video with the class.

Special Feature 2

Converting Frames on Your Web Site



Objectives

You will have mastered the material in this special feature when you can:

- Define terms related to frames
- Identify all parts of a framed Web site structure
- Discuss the purpose and design of a frame definition file
- Determine a structure to replace frames on a Web site
- Develop a Web site to replace frames

Introduction

A **frame** is a rectangular area of a Web page — essentially, a window — in which a separate Web page can be displayed. Frames allow a user to display several Web pages at one time in a single browser window. Each frame displays a different, individual Web page, each of which is capable of interacting with other Web pages. Web pages that include frames look and act differently from Web pages created in previous projects. Frames are not used in the projects in this book because many screen readers, such as those used by visually impaired people, have difficulty displaying a Web site that uses frames. The Americans with Disabilities Act (ADA) standards recommend that frames not be used for Web sites. In addition to problems with devices for the disabled, there are other potential problems with the use of frames for Web development. Frames can cause problems when people bookmark or add the Web page as a favorite. It is the frame definition that is stored, and that may not be the page that the user wants to save. Search engines also may have problems indexing a Web site that uses frames. Finally, when users print a Web page with frames, they may not get what they see on the screen. However, many Web sites do utilize a frame structure, and as a Web developer, you may be responsible for maintaining Web sites based on frames. This Special Feature explains frames and how they are used, and shows you how to convert Web sites from a frame structure to better accommodate ADA standards and to address other frame-related issues.

BTW

Americans with Disabilities Act

Review the Web site dedicated to this act, www.ada.org. It contains a wealth of information on the standards related to this act, including information about Web site functionality.

Project — Converting Frames

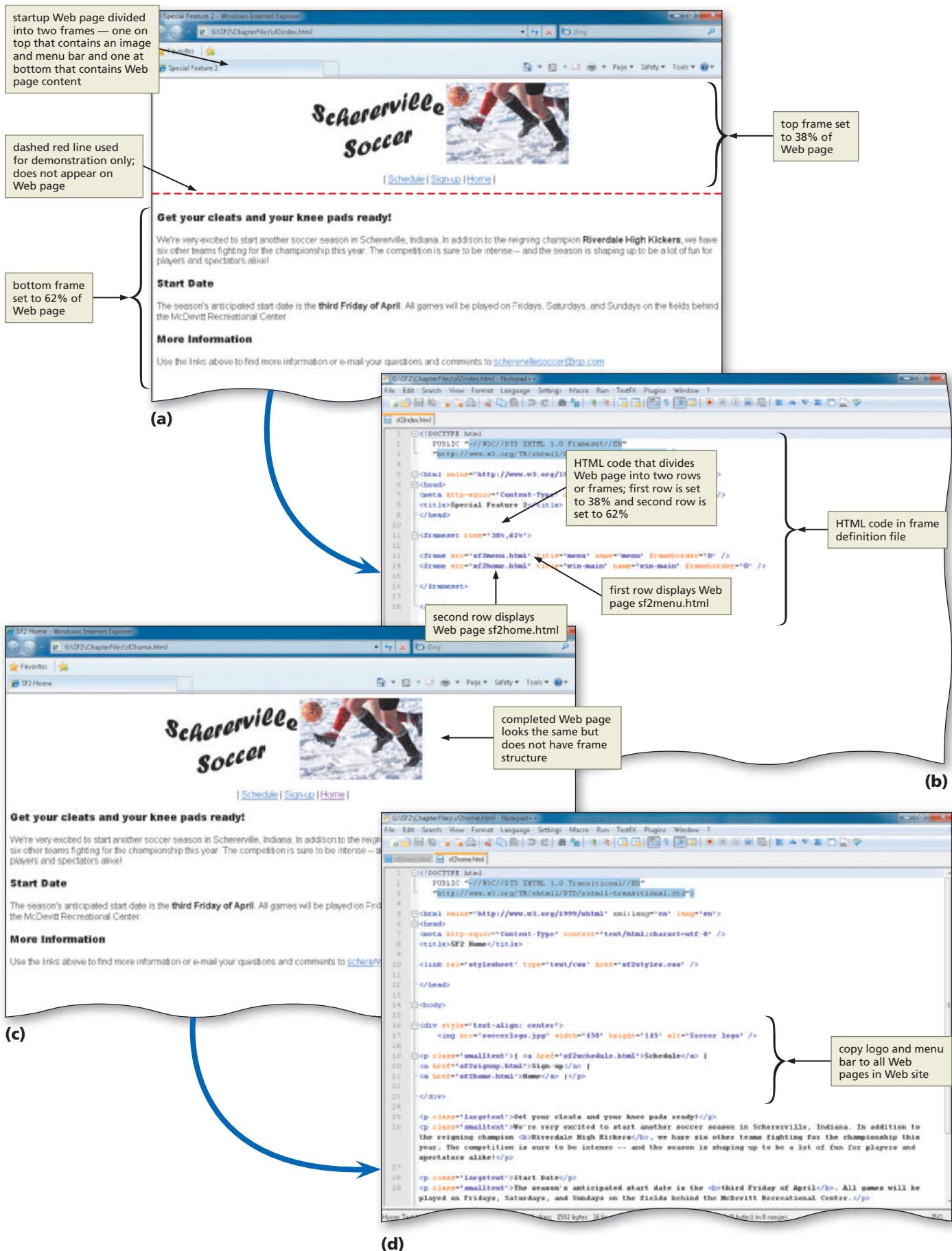
The Web site presented in this Special Feature is based on a frame structure. In order to understand the frame structure, you have to know what the frame definition is and how it works. You also have to understand how the other Web pages in a Web site are related to the frame definition file.

In this Special Feature, you review frame-based Web pages provided in the Data Files for Students. See the inside back cover of this book for instructions on downloading the Data Files for Students, or contact your instructor for information about accessing the required files. Figure 1a shows the initial Web site using frames. A red dashed line indicates where the Web page is divided into frames. You see that the top frame (the section of the Web page above the red dashed line) contains an image of soccer players and a menu bar that is used for navigation. The bottom frame (the section of the Web page beneath the red dashed line) contains the Web page content. You look at the frame definition file (Figure 1b) and see how it works with the other files in the Web site. You are then introduced to options that can take the place of a frame structure. Finally, you redesign the Web site to look similar but remove the frames (Figures 1c and 1d).

Overview

As you read through this feature, you will learn how to assess the frame definition file that creates the Web site shown in Figure 1a. You then convert the Web site structured with frames (Figure 1b) to a Web site structured without frames, using techniques that you have previously studied (Figures 1c and 1d). You complete this by performing these general tasks:

- Determine the use of a frame definition file to structure a Web site.
- Identify what other Web design methods could be used to restructure the Web site.
- Make the changes necessary to the Web pages provided to restructure the Web site.
- Validate and test the Web pages.

**Figure 1**

**Plan
Ahead****General Project Guidelines**

The Web pages used in this project are already completed with a frame structure. In order to change the structure, you first need to understand how the current frame structure works. Then you can determine techniques that can be used to change the structure. In preparation for this project, you should follow these general guidelines:

- 1. Review the layout of the frame structure.** Frames can assume different layouts. You could have a two-frame layout in which there is a menu bar across the top or left side of the Web page or a three- or four-frame layout. You must first assess the existing frame structure.
- 2. Identify the purpose and structure of the frame definition file.** Once you have determined the structure, you need to review the frame definition file to see how it applies to the layout. This must be done so that you can convert the Web site appropriately.
- 3. Determine a conversion strategy that can be used to restructure the Web site.** A Web site needs to be converted effectively. There are different techniques that you can use, but each technique needs to be assessed for effectiveness and efficiency.
- 4. Make the changes necessary.** Once your technique is determined, you need to make the necessary changes to the Web site to convert it from a frame structure to a new structure (as developed in all chapters of this book).

BTW**Frames**

Frames represent an older technology that is still in use today. Research the terms “frames” or “HTML frames” on the Internet to determine how prevalent they are in current Web development.

The Frame Definition File

The frame definition file tells the browser what frames to display and structures the frame layout. This is the file that you open in the browser in order to view the Web site. The frame definition file contains the names of the two, three, or four Web pages that are opened in frames at startup.

Table 1 shows the frame tags and attributes that are used to create a frame structure.

Table 1 Frame Tag Attributes

Tag	Attribute	Function
<frameset>	cols rows	<ul style="list-style-type: none"> Indicates the number of columns Indicates the number of rows
<frame>	frameborder marginwidth marginheight noresize name scrolling src	<ul style="list-style-type: none"> Turns frame borders on or off (1 or 0) Adjusts the margin on the left and right of a frame Adjusts the margin above and below a document within a frame Locks the borders of a frame to prohibit resizing Defines the name of a frame that is used as a target Indicates whether a scroll bar is present Indicates the Web page or other file to be displayed in the frame

The HTML code that creates this frame structure for the Web page shown in Figure 1a on page HTML 395 is contained in the sf2index.html frame definition file, and is as follows (line numbers have been added):

```

1  <!DOCTYPE html
2      PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
3          "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
4  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
5      lang="en">
6  <head>
7      <meta http-equiv="Content-Type" content="text/
6      html; charset=utf-8" />
8  </head>
9  <frameset rows="38%,62%">
10 <frame src="sf2menu.html" title="menu" name="menu"
10     frameborder="0" />
11 <frame src="sf2home.html" title="win-main" name="win-main"
11     frameborder="0" />
12 </frameset>
13 </html>
```

Note that in lines 2 and 3 of the code, the word “Transitional” that you have used in all chapter projects is replaced with “Frameset.” This is because the document type to use frames is Frameset.

In the code above, also notice that one frameset (line 9) is inserted that contains two rows (top and bottom) that create two sections (or frames) of 38% and 62% of the Web page, respectively. (If you wanted to structure the frames with the menu frame on the left and the content frame on the right, you would change the attribute “rows” in the frameset line of code to “cols” instead and adjust the percentages as needed.) There will always be one frame for each `<frame />` tag that is used. That is, the `<frame />` tag defines one particular window (or frame) within a frameset. Each frame in a frameset can have different attributes, such as border, scrolling, the ability to resize, etc., but each frame contains only one Web page at a time as described below.

At startup (on lines 10 and 11 of the code), you see that the `sf2menu.html` file opens in the first (top) frame, and the `sf2home.html` file opens in the second (bottom) frame (or row). In other words, the top frame (`sf2menu.html`, that contains the logo and menu bar) displays in the top 38% of the Web page, while the content frame (or section) named `sf2home.html` displays in the bottom 62% of the Web page. Also note that line 11 gives that frame the target name “`win-main`” as an identifier. The name “`win-main`” is used by the `sf2menu.html` file (shown later) as the target frame in which all content is displayed.

BTW

Framesets

A frameset can be thought of as a window with various windowpanes.

Within each pane is a separate Web page. The frame definition file is the HTML file that defines the Web pages that are displayed in the individual panes. Every Web page used in a frameset can be viewed independently in the browser as well as within the frameset.

To View a Web Site with Frames

To view the Web site with frames used in this project, you open the sf2index.html frame definition file in the browser. Then you open the HTML file and compare this Web site to the code shown in the frame definition file.

1

- With a USB drive plugged into your computer, start your browser.
- Open the sf2index.html file in the SF2\ChapterFiles folder of the Data Files for Students (Figure 2a).
- Click the menu options (Schedule, Sign-up, and Home) to view the entire Web site.



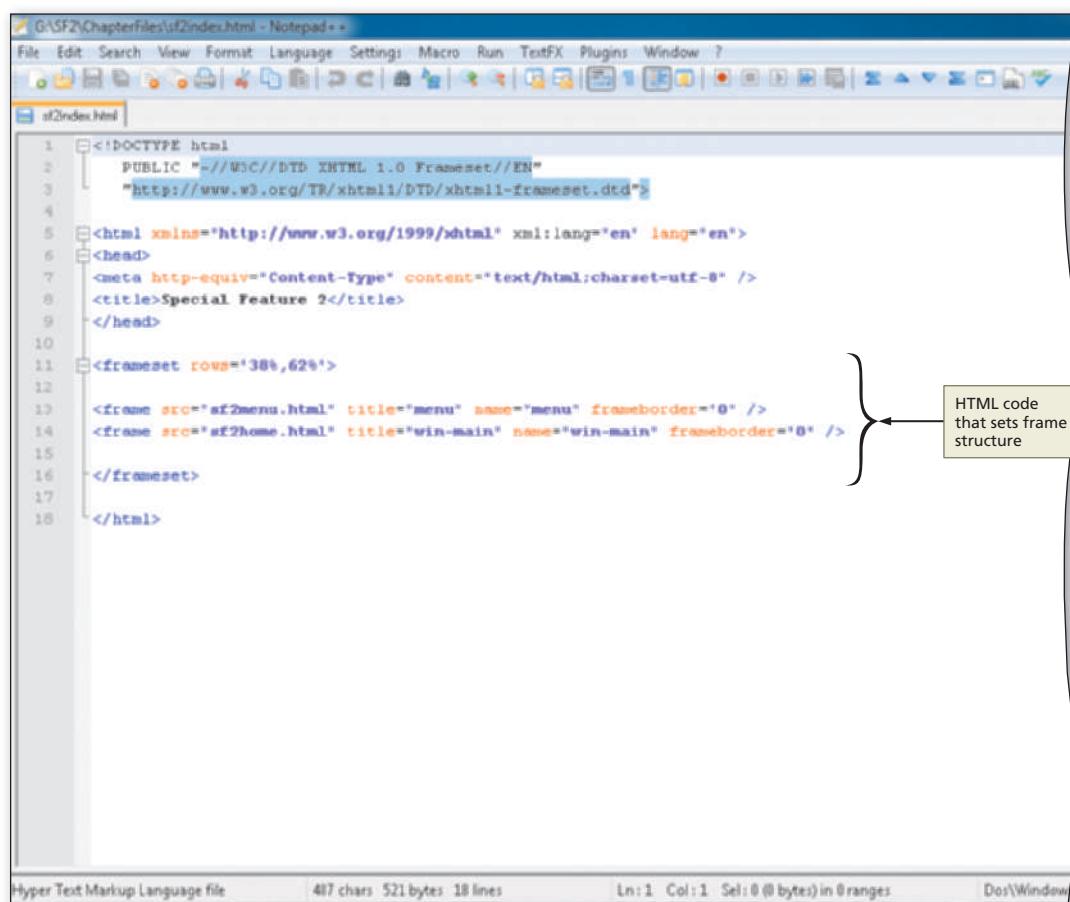
Figure 2 (a)

2

- Start Notepad++ and open the sf2index.html frame definition file in the SF2\ChapterFiles folder of the Data Files for Students (Figure 2b).

Q&A How can you tell that this Web site uses frames?

The only way to be completely sure that the structure uses frames is to open the Web page source code. You immediately see that the structure of the Web page (shown in Figure 2b) is developed in frames. In this Web site, the soccer logo and the menu bar are always present on the top section (or frame) of the Web page, and this section is static (i.e., it remains the same always). The content from the sf2home.html file displays in the bottom frame (the target named “win-main”). A Web site visitor can click any of the three menu items (Schedule, Sign-up, Home), and the content from those Web pages displays in the dynamic (i.e., changing) target “win-main” (the bottom frame) of the Web page.



```

1 <!DOCTYPE html
2   PUBLIC "-//W3C//DTD XHTML 1.0 Frameset//EN"
3   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd">
4
5 <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
6 <head>
7 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
8 <title>Special Feature 2</title>
9 </head>
10
11 <frameset rows="38%,62%">
12
13 <frame src="sf2menu.html" title="menu" name="menu" frameborder="0" />
14 <frame src="sf2home.html" title="win-main" name="win-main" frameborder="0" />
15
16 </frameset>
17
18 </html>

```

HTML code that sets frame structure

Figure 2 (b)

Frame Layouts

Frame layouts can be designed in a variety of ways. The goal and purpose of the Web site determine which layout is appropriate. For example, the Schererville Soccer Web site uses a basic two-frame structure, as shown in Figures 2a and 2b. The menu on the top remains constant, and the content frame on the bottom changes.

BTW**Web Page Design**

Whether you are using frames or any other navigation technique, design is very important. Many Web sites are available that provide information about the use of frames and other Web design techniques. These sites include links to sample Web sites that demonstrate advanced Web design topics. Many design tips also are available that can help you create a Web site that is informative as well as attractive.

Figure 3 shows a three-frame structure, often used to display a company logo (top) in the third frame. To create a three-frame structure as shown in Figure 3, the HTML code is as follows. This time, you identify two rows first. Within the second row, you identify two columns. There are two frameset tags used in the code because these tags define the structure of the frames within a window.

```
<frameset cols="25%, 75%">
<frameset rows="20%, 80%">
<frame src="header.html" title="header" name="header"
scrolling="no" />
<frame src="menu.html" title="menu" name="menu" />
</frameset>
<frame src="home.html" title="win-main" name="win-main" />
</frameset>
```

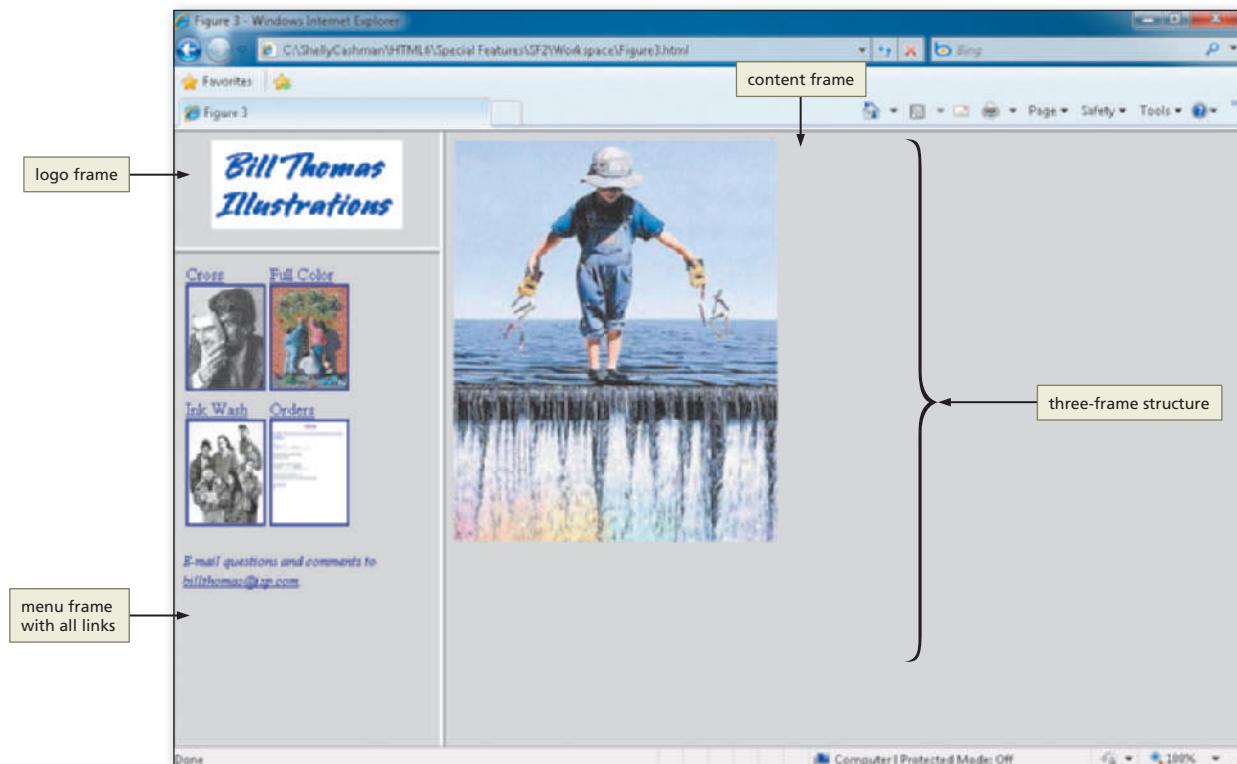
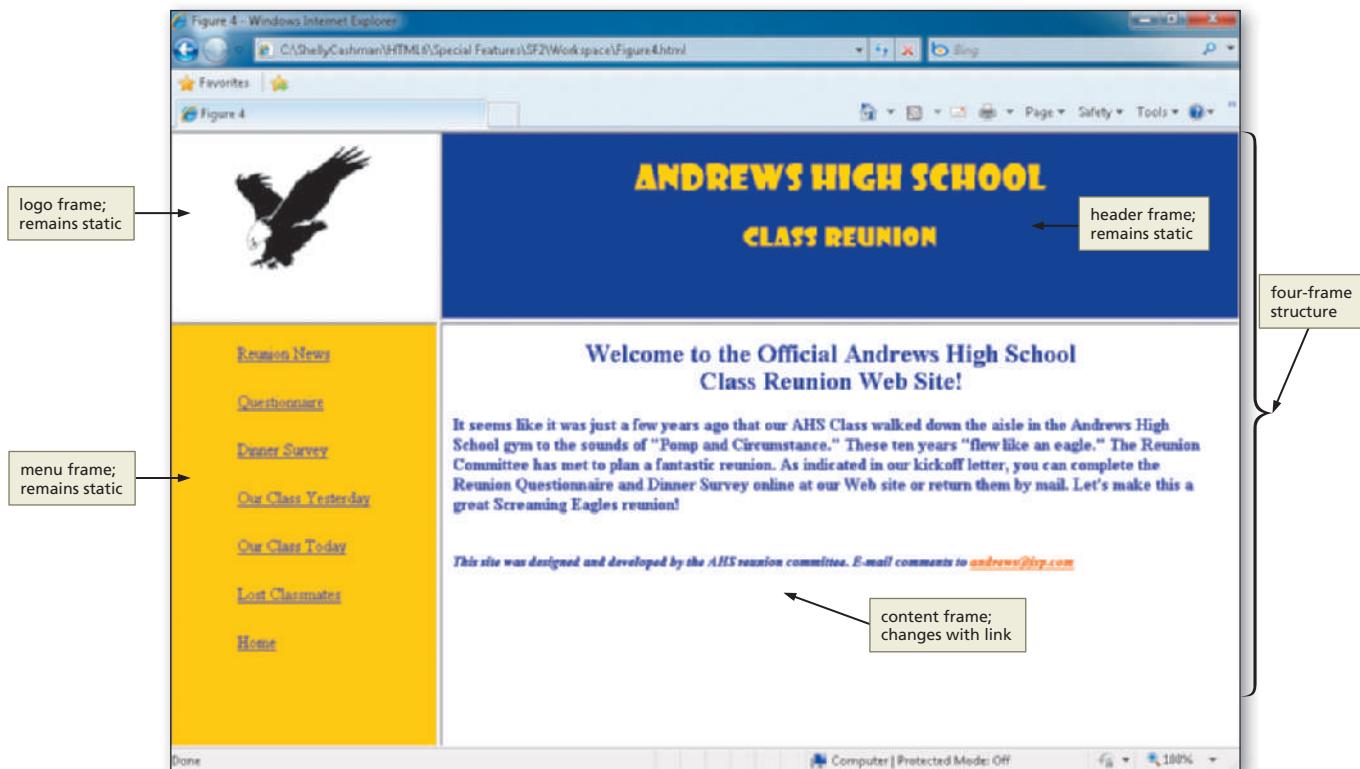


Figure 3

A four-frame structure, as shown in Figure 4, can be used to split a header image from the header text. The HTML code needed to create the four-frame structure shown in Figure 4 is as follows. This time, you identify two rows first with two columns in each row.

```
<frameset rows="30%,70%">
<frameset cols="25%,75%">
<frame src="logo.html" title="logo" name="logo" />
<frame src="header.html" title="header" name="header" />
</frameset>
<frameset cols="25%,75%">
<frame src="menu.html" title="menu" name="menu" />
<frame src="home.html" title="win-main" name="win-main"/>
</frameset>
</frameset>
```

**Figure 4**

Determining a Conversion Strategy

Now that you understand the purpose and organization of the frame definition file, you must decide how to convert the Web site from a frame structure to an ADA-compliant structure. You have several methods to choose from to accomplish this conversion. One way is similar to what you did in the Chapter 7 project, using the `<div> </div>` tags to provide a pop-up window when a user hovers over an image. Here, however, you want the content of a second Web page to remain visible all the time, not just when a user hovers over it. Using the division tag, you can lay out a Web page so that a menu remains constant in one division (or section), and the content displays in a second division (or

section) of the Web page. That is fine, but it means that you must have all content in one Web page. That is not an efficient way to convert from frames for this project.

A second way to convert from a frame structure is to copy the static portion of the Web page (the image and the menu bar) to all pages in the Web site. The sf2menu.html file contains the logo and the menu bar for this Web site. You will open this file and copy the logo and menu bar code and paste it into the sf2home.html, sf2schedule.html, and sf2signup.html Web pages.

To Copy HTML Code for a Menu Bar

The following step illustrates how to copy the necessary menu bar code from the sf2menu.html file.

1

- If necessary, start Notepad++.
- With a USB drive plugged into your computer, open the sf2menu.html file in the SF2\ChapterFiles folder.

Q&A

Why open this file and not the frame definition file, sf2index.html?

The frame definition file, sf2index.html, only contains the structure for the frames. The sf2menu.html contains the logo and menu bar that you want to copy to the other Web pages in the Web site.

- Highlight lines 16 through 23 (Figure 5).
- Click Edit on the menu bar and then click Copy.

```

<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Schererville Soccer</title>
<link rel="stylesheet" type="text/css" href="sf2styles.css" />
</head>
<body>
<div style="text-align: center">

<p style="font-face: broadway">| <a href="sf2schedule.html" target="win-main">Schedule</a> |<br/>
<a href="sf2signup.html" target="win-main">Sign-up</a> |<br/>
<a href="sf2home.html" target="win-main">Home</a> |</p>
</div>
</body>
</html>

```

Figure 5

To Paste Code into the Home Page and Edit the Code

The following step illustrates how to paste the code copied above into the home page and edit the code to remove the target property.

1

- In Notepad++, open the sf2home.html file in the SF2\ChapterFiles folder on the USB drive.
- Click after the > in the <body> tag on line 14 and press the ENTER key twice.
- Click Edit on the menu bar and then click Paste.
- Delete the code target="win-main" on lines 19, 20, and 21 (Figure 6).
- Click the Save icon on the toolbar to save the sf2home.html file with the new code.

```

1  <!DOCTYPE html
2      PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
3      "http://www.w3.org/TR/xhtml1/DTD/xhtml-transitional.dtd">
4
5  <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
6      <head>
7          <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
8          <title>SF2 Home</title>
9
10         <link rel="stylesheet" type="text/css" href="sf2styles.css" />
11
12     </head>
13
14     <body>
15
16         <div style="text-align: center">
17             
18
19             <p class="smalltext">| <a href="sf2schedule.html">Schedule</a> |<br/>
20             <a href="sf2signup.html">Sign-up</a> |<br/>
21             <a href="sf2home.html">Home</a> |</p>
22
23         </div>
24
25
26         <p class="largetext">Get your knee pads ready!</p>
27         <p class="smalltext">We're very excited to start another soccer season in Schererville, Indiana. In addition to the reigning champion <b>Riverdale High Kickers</b>, we have six other teams fighting for the championship year. The competition is sure to be intense -- and the season is shaping up to be a lot of fun for play spectators alike!</p>
28
29         <p class="largetext">Start Date</p>
30         <p class="smalltext">The season's anticipated start date is the <b>third Friday of April</b>. All games played on Fridays, Saturdays, and Sundays on the fields behind the McDevitt Recreational Center.</p>

```

target="win-main" code deleted in lines 19 through 21

lines 16 through 23 pasted into home page

Figure 6

Q&A By copying/pasting this code, I can provide the same look to my Web site visitors as a frame structure, right?

Yes, your Web site will operate in the same manner without the frame structure, with the image and the menu bar across the top of the Web page and the content changing on the bottom part.

Q&A Why do I have to remove the target="win-main" code?

The target is named to support the dynamic (i.e., changing) frame within the frame structure. Because there is now only one Web page and no frame structure, you do not need to identify the target frame.

To Copy and Paste Code into Other Web Pages

The following steps illustrate how to copy and paste the menu bar code into the Schedule and Signup Web pages.

1

- Copy lines 16 through 23 in the newly saved sf2home.html file (the code target = "win-main" on lines 19 through 21 should be deleted).
- In Notepad ++, open the sf2schedule.html file. Click the beginning of line 16 and press ENTER once. Move the insertion point back to line 16 and paste the copied eight lines of code.
- Click the Save icon on the toolbar to save the sf2schedule.html file with the new code (Figure 7).

```

<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml/DTD/xhtml-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>SF2 Schedule</title>
<link rel="stylesheet" type="text/css" href="sf2styles.css" />
</head>
<body>
<div style="text-align: center">

<p style="font-family: broadway">| <a href="sf2schedule.html">Schedule</a> |<br/>
| <a href="sf2signup.html">Sign-up</a> |<br/>
| <a href="sf2home.html">Home</a> |</p>
</div>
<table border="1" cellpadding="5" cellspacing="10">
<tr>
<th colspan="3">April Games</th>
</tr>
<tr>
<th>Teams</th>
<th>Day</th>
<th>Time</th>
</tr>

```

Figure 7

2

- Open the sf2signup.html file. Click the beginning of line 16 and press ENTER once. Move the insertion point back to line 16 and paste the same eight lines of code.
- Click the Save icon on the toolbar to save the sf2signup.html file with the new code.

To Validate and Print a Document

You are finished entering the lines of code needed for the logo and menu bar. As with all Web development projects, you now need to save, validate, and print the files.

- Validate sf2home.html, sf2schedule.html, and sf2signup.html using the w3.org validation service.
- Once the files are all successfully validated, print the files.

To View the Web Site in the Browser

As with all other projects, it is important to view the Web pages in the browser and test all links. After converting the Web site structure, you open the sf2home.html file as your starting point rather than the frame definition file sf2index.html.

- 1** If necessary, start your browser. Open the file sf2home.html in the browser (Figure 8).
- 2** Test all links by clicking the menu bar items Schedule, Sign-up, and Home.

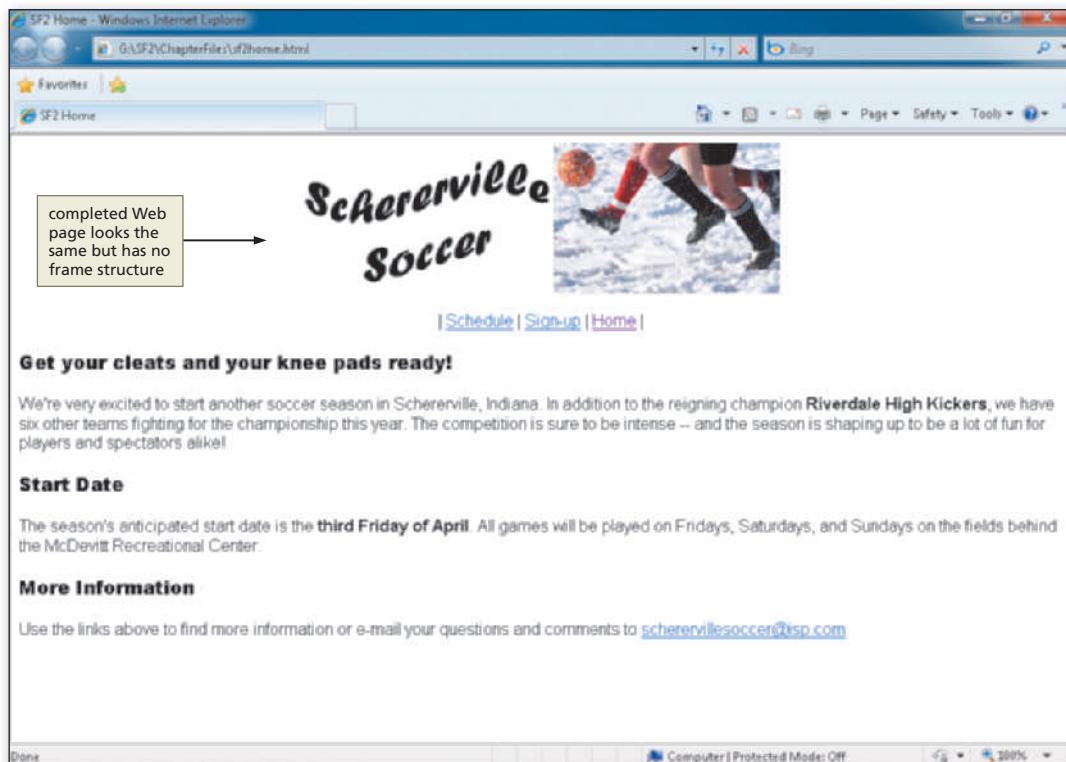


Figure 8

Feature Summary

In this feature, you have learned how to convert a frame-structured Web site to one in which frames are not used. This is an important aspect of Web development because of the ADA compliance standards that do not fully support frames. The items listed below include all the new Web development skills you have learned in this feature.

- 1. View a Web Site with Frames (HTML 398)
- 2. Copy HTML Code for a Menu Bar (HTML 402)
- 3. Paste Code into the Home Page and Edit the Code (HTML 403)
- 4. Copy and Paste Code into Other Web Pages (HTML 404)

In the Lab

Design and/or create a document using the guidelines, concepts, and skills presented in this chapter. Labs are listed in order of increasing difficulty.

Lab 1: Determining Another Way to Convert from Frames

Problem: Your assignment is to research frame conversion strategies and determine another way to convert a Web site from a frame structure to a non-frame structure.

Instructions:

1. Search online for additional information about ADA compliance issues related to frames.
2. Review other techniques that can be used to convert a Web site structured with frames.
3. Write a paper that addresses the topics discussed in this Special Feature. Your paper should:
 - a. Specify reasons frames may not be an effective Web site structure
 - b. Identify techniques that could be used to restructure a Web site
 - c. Determine how you could utilize these techniques in your own Web development.
4. Save the paper with the name Lab SF2-1 Conversion Strategies. Submit the file in the format specified by your instructor.

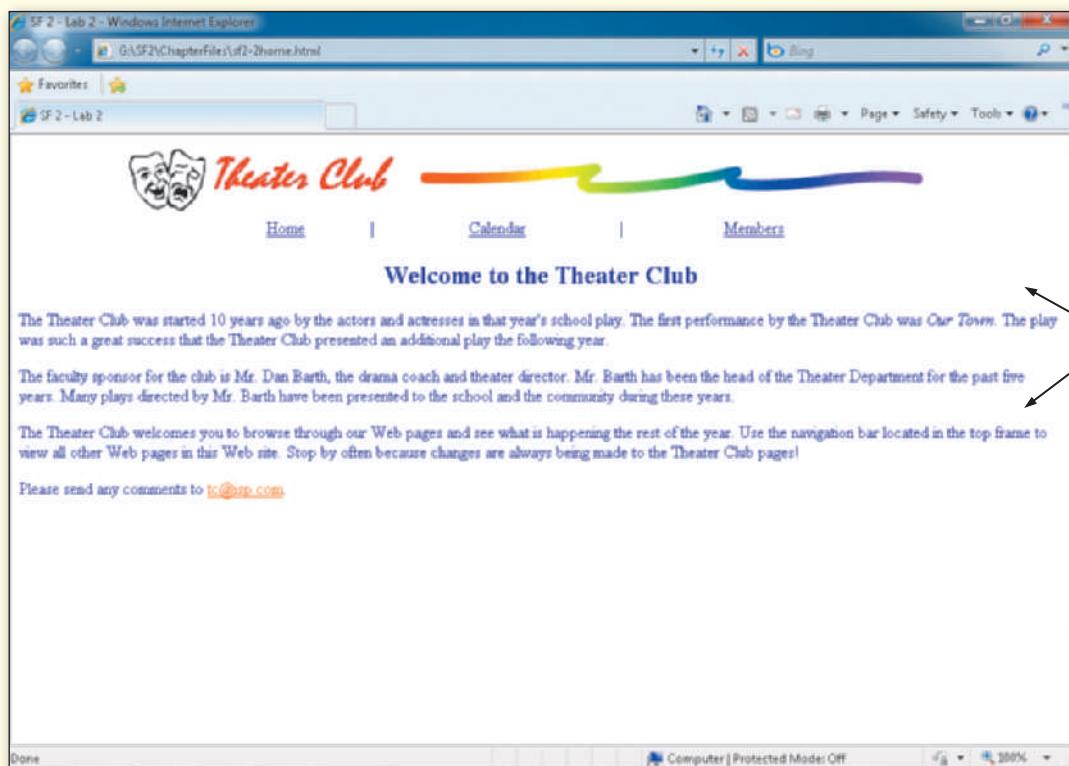
In the Lab

Lab 2: Convert a Second Web Site

Problem: In this assignment, you convert a Web site that is based on the frame structure to a non-frame structure, as shown in Figure 9.

Instructions:

1. In your browser, open the LabSF2-2.html file found in the Data Files for Students.
2. Review the structure and functionality of the Web site highlighted in this file.
3. Determine how you can restructure the Web site by converting from its current frame structure. Your conversion strategy can be that which is used in this Special Feature or another strategy that you found in your research in Lab 1.
4. Make the changes to the Web pages in the Web site. Save the new file as LabSF2-2solution.html, validate all files, test the site, and print the pages of the modified site.
5. Submit the files in the format specified by your instructor.

**Figure 9**

This page intentionally left blank

Appendix A

HTML Quick Reference

HTML and XHTML Coding Standards

HTML is the original language used for publishing hypertext on the World Wide Web. It is a nonproprietary format based on Standard Generalized Markup Language (SGML). HTML documents can be created with a wide variety of tools, from simple text editors such as Notepad and Notepad++, to sophisticated WYSIWYG authoring tools such as Adobe Dreamweaver. Extensible Markup Language (XML) is a markup language that uses tags to describe the structure and content of a document, not just the format.

Extensible Hypertext Markup Language (XHTML) is a reformulation of HTML so it conforms to XML structure and content rules. By combining HTML and XML, XHTML provides the display features of HTML and the stricter coding standards required by XML.

Table A-1 lists some important XHTML coding practices that Web developers should follow to ensure that their HTML code conforms to XHTML standards as defined by the World Wide Web Consortium (W3). The projects in this book follow XHTML standards and adhere to the rules outlined in Table A-1. This information is also shown in Table 1-3 in Chapter 1.

Table A-1 XHTML Coding Practices

Practice	Invalid Example	Valid Example
HTML file must include a DOCTYPE statement	<html> <head><title>sample Web page</title>	<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1- transitional.dtd"> <html> <head><title>sample Web page</title></head>
All tags and attributes must be written in lowercase	<TABLE WIDTH=" 100% ">	<table width=" 100% ">
All attribute values must be enclosed by single or double quotation marks	<table width=100%>	<table width=" 100% ">
All tags must be closed, including tags such as img, hr, and br, which do not have end tags, but which must be closed as a matter of practice	 <hr> <p>This is another paragraph	 <hr /> <p>This is another paragraph</p>
All elements must be nested properly	<p>This is a bold paragraph</p>	<p>This is a bold paragraph</p>

HTML Tags and Attributes

HTML uses tags such as `<h1>` and `<p>` to structure text into headings, paragraphs, lists, hypertext links, and so on. Many HTML tags have attributes that can be defined in different ways to further modify the look of the Web page. Table A–2 lists HTML tags and their associated attributes. The list provides a brief description of each tag and its attributes. The default value for each attribute is indicated by bold text in the Description column. For a comprehensive list, more thorough descriptions, examples of all HTML tags, and XHTML coding standards, visit the World Wide Web Consortium Web site at www.w3.org.

As the World Wide Web Consortium continually updates the HTML specifications, HTML tags are added to, deleted, and replaced by newer tags. In the list in Table A–2, deprecated elements—tags that can be replaced with newer elements—are indicated with an asterisk. Deprecated elements still are available for use, and most browsers still support them. Browsers will probably continue to support deprecated tags and attributes in the near future, but eventually these tags may become obsolete, so future support cannot be guaranteed. It is therefore a best practice to not use deprecated tags. Obsolete elements are no longer in use and are not supported by common browsers. This appendix does not list obsolete elements.

Table A–2 HTML Tags and Attributes

HTML Tag and Attributes	Description
<code><!DOCTYPE></code>	Indicates the version of XHTML used
<code><!--Text here--></code>	Inserts invisible comments
<code><a>....</code>	Anchor; creates a hyperlink or fragment identifier
<code>charset=character set</code>	Specifies the character encoding of the linked resource
<code>href=url</code>	Hyperlink reference that specifies the target URL
<code>name=text</code>	Specifies a name for enclosed text, allowing it to be the target of a hyperlink
<code>rel=relationship</code>	Indicates the relationship going from the current page to the target
<code>rev=relationship</code>	Indicates the relationship going from the target to the current page
<code>target=name *</code>	Defines the name of the window or frame in which the linked resource will appear
<code><address>....</address></code>	Used for information such as authorship, e-mail addresses, or addresses; enclosed text appears italicized and indented in some browsers
No attributes	
<code><area>....</area></code>	Creates a clickable area, or hotspot, on a client-side image map
<code>coords=value1, value2</code>	Specifies the coordinates that define the edges of the hotspot; a comma-delimited list of values
<code>href=url</code>	Hyperlink reference that specifies the target URL
<code>Nohref</code>	Indicates that no link is associated with the area
<code>shape=shape</code>	Identifies the shape of the area (poly, rect, circle)
<code>target=name *</code>	Defines the name of the window or frame in which the linked resource will appear
<code>....</code>	Specifies text to appear in bold
No attributes	
<code><base /></code>	Identifies the base in all relative URLs in the document
<code>href=url</code>	Specifies the absolute URL used to resolve all relative URLs in the document
<code>target=name *</code>	Defines the name for the default window or frame in which the hyperlinked pages are displayed

Table A-2 HTML Tags and Attributes (continued)

HTML Tag and Attributes	Description
<big>....</big>	Increases the size of the enclosed text to a type size bigger than the surrounding text; exact display size depends on the browser and default font
No attributes	
<blockquote>....</blockquote>	Sets enclosed text to appear as a quotation, indented on the right and left
No attributes	
<body>....</body>	Defines the start and end of a Web page's content
alink=color *	Defines the color of an active link
background=url *	Identifies the image to be used as a background
bgcolor=color *	Sets the document's background color
link=color *	Defines the color of links not yet visited
vlink=color *	Defines the color of visited links
....	Sets enclosed text to appear in bold
No attributes	
 	Inserts a line break
clear=margin *	Sets the next line to start in a spot where the requested margin is clear (left, right, all, none); used to stop text wrap
<caption>....</caption>	Creates a caption for a table
align=position *	Sets caption position (top, bottom, left, right)
<center>....</center> *	Centers the enclosed text horizontally on the page
No attributes	
<cite>....</cite>	Indicates that the enclosed text is a citation; text usually is displayed in italics
No attributes	
<code>....</code>	Indicates that the enclosed text is a code sample from a program; text usually is displayed in fixed width font such as Courier
No attributes	
<col>....</col>	Organizes columns in a table into column groups to share attribute values
align=position	Sets horizontal alignment of text within the column (char, center, top, bottom, left, right)
span=value	Sets the number of columns that span the <col> element
valign=position	Specifies vertical alignment of text within the column (top, middle, bottom)
width=value	Sets the width of each column in the column group
<colgroup>....</colgroup>	Encloses a group of <col> tags and groups the columns to set properties
align=position	Specifies horizontal alignment of text within the column (char, center, top, bottom, left, right)
char=character	Specifies a character on which to align column values (for example, a period is used to align monetary values)
charoff=value	Specifies a number of characters to offset data aligned with the character specified in the char property
span=number	Sets the number of columns the <col> element spans
valign=position	Specifies vertical alignment of text within the column (top, middle, bottom)
width=value	Sets the width of each column spanned by the colgroup statement

Table A–2 HTML Tags and Attributes (continued)

HTML Tag and Attributes	Description
<dd>....</dd> No attributes	Indicates that the enclosed text is a definition in the definition list
<div>....</div> <i>align=position *</i> <i>class=name</i> <i>id=name</i>	Defines block-level structure or division in the HTML document Specifies alignment of the content block (center, left, right) Assigns the class name to each class of divisions Assigns a unique name to a specific content block
<dl>....</dl> No attributes	Creates a definition list
<dt>....</dt> No attributes	Indicates that the enclosed text is a term in the definition list
.... No attributes	Indicates that the enclosed text should be emphasized; usually appears in italics
<fieldset>....</fieldset> <i>align=position</i>	Groups related form controls and labels Specifies alignment of a legend as related to the fieldset (top, bottom, middle, left, right)
.... * <i>size=value *</i> <i>color=color *</i> <i>face=list *</i> <i>point-size=value *</i> <i>weight=value *</i>	Defines the appearance of enclosed text Sets the font size in absolute terms (1 through 7) or as a relative value (for example, +2) Sets the font color; can be a hexadecimal value (#rrggbb) or a word for a predefined color value (for example, navy) Identifies the font face; multiple entries should be separated by commas Sets the point size of text for downloaded fonts Sets the weight of the font, ranging from 100 (lightest) to 900 (heaviest)
<form>....</form> <i>action=url</i> <i>enctype=encoding</i> <i>method=method</i> <i>target=text</i>	Marks the start and end of a Web page form Specifies the URL of the application that will process the form; required attribute Specifies how the form element values will be encoded Specifies the method used to pass form parameters (data) to the server Specifies the frame or window that displays the form's results
<frame>....</frame> <i>frameborder=option</i> <i>marginheight=value</i> <i>marginwidth=value</i> <i>name=text</i> <i>noresize</i> <i>scrolling=option</i> <i>src=url</i>	Delimits a frame within a frameset Specifies whether the frame border is displayed (yes, no) Adds <i>n</i> pixels of space above and below the frame contents Adds <i>n</i> pixels of space to the left and the right of the frame contents Specifies the name of the frame Prevents the user from resizing the frame Adds scroll bars or not—always (yes), never (no), or add when needed (auto) Defines the URL of the source document that is displayed in the frame
<frameset>....</frameset> <i>cols=value1, value2,...</i> <i>rows= value1, value2,...</i> <i>frameborder=option</i>	Defines a collection of frames in a frameset Defines the number and width of frames within a frameset Defines the number and height of frames within a frameset Specifies whether the frame border is displayed (yes, no)

Table A-2 HTML Tags and Attributes (continued)

HTML Tag and Attributes	Description
<hn>....</hn>	Defines a header level <i>n</i> , ranging from the largest (h1) to the smallest (h6)
<td>Specifies the header alignment (left, center, right)</td>	Specifies the header alignment (left , center, right)
<head>....</head>	Delimits the start and end of the HTML document's head
No attributes	
<hr />	Inserts a horizontal rule
<td>Specifies the alignment of the horizontal rule (left, center, right)</td>	Specifies the alignment of the horizontal rule (left, center , right)
<td>Specifies to not use 3D shading and to round the ends of the rule</td>	Specifies to not use 3D shading and to round the ends of the rule
<td>Sets the thickness of the rule to a value in pixels</td>	Sets the thickness of the rule to a value in pixels
<td>Sets the width of the rule to a value in pixels or a percentage of the page width; percentage is preferred</td>	Sets the width of the rule to a value in pixels or a percentage of the page width; percentage is preferred
<html>....</html>	Indicates the start and the end of the HTML document
version=data	Indicates the HTML version used; not usually used
<i>....</i>	Sets enclosed text to appear in italics
No attributes	
<iframe>....</iframe> *	Creates an inline frame, also called a floating frame or subwindow, within an HTML document
<td>Aligns the frame with respect to context (top, middle, bottom, left, right)</td>	Aligns the frame with respect to context (top, middle, bottom , left, right)
<td>Specifies whether a frame border is displayed (1=yes; 0=no)</td>	Specifies whether a frame border is displayed (1=yes; 0=no)
<td>Sets the frame height to a value in pixels</td>	Sets the frame height to a value in pixels
<td>Sets the margin between the contents of the frame and its top and bottom borders to a value in pixels</td>	Sets the margin between the contents of the frame and its top and bottom borders to a value in pixels
<td>Sets the margin between the contents of the frame and its left and right borders to a value in pixels</td>	Sets the margin between the contents of the frame and its left and right borders to a value in pixels
<td>Assigns a name to the current frame</td>	Assigns a name to the current frame
<td>Prevents the user from resizing the frame</td>	Prevents the user from resizing the frame
<td>Defines the URL of the source document that is displayed in the frame</td>	Defines the URL of the source document that is displayed in the frame
<td>Sets the frame width to a value in pixels</td>	Sets the frame width to a value in pixels
<td>Adds scroll bars or not—always (yes), never (no), or add when needed (auto)</td>	Adds scroll bars or not—always (yes), never (no), or add when needed (auto)
....	Inserts an image into the current Web page
<td>Defines image alignment in relation to the text or the page margin (top, middle, bottom, right, left)</td>	Defines image alignment in relation to the text or the page margin (top, middle, bottom, right, left)
<td>Provides a text description of an image if the browser cannot display the image; always should be used</td>	Provides a text description of an image if the browser cannot display the image; always should be used
<td>Sets the thickness of the border around the image to a value in pixels; default size is 3</td>	Sets the thickness of the border around the image to a value in pixels; default size is 3
<td>Sets the height of the image to a value in pixels; always should be used</td>	Sets the height of the image to a value in pixels; always should be used
<td>Specifies the URL of the image to be displayed; required</td>	Specifies the URL of the image to be displayed; required
<td>Specifies the map of coordinates and links that defines the href within this image</td>	Specifies the map of coordinates and links that defines the href within this image
<td>Sets the width of the image to a value in pixels; always should be used</td>	Sets the width of the image to a value in pixels; always should be used
<input>....</input>	Defines controls used in forms
<td>Provides a short description of the control or image button; for browsers that do not support inline images</td>	Provides a short description of the control or image button; for browsers that do not support inline images

Table A-2 HTML Tags and Attributes (continued)

HTML Tag and Attributes	Description
<input>....</input> (continued)	
checked	Sets radio buttons and check boxes to the checked state
disabled	Disables the control
maxlength= <i>value</i>	Sets a value for the maximum number of characters allowed as input for a text or password control
name= <i>text</i>	Assigns a name to the control
readonly	Prevents changes to the control
size= <i>value</i>	Sets the initial size of the control to a value in characters
src= <i>url</i>	Identifies the location of the image if the control is set to an image
tabindex= <i>value</i>	Specifies the tab order between elements in the form, with 1 as the first element
type= <i>type</i>	Defines the type of control (text, password, check box, radio, submit, reset, file, hidden, image, button)
usemap= <i>url</i>	Associates an image map as defined by the <map> element
value= <i>data</i>	Sets the initial value of the control
<ins>....</ins>	Identifies and displays text as having been inserted in the document in relation to a previous version
cite= <i>url</i>	Specifies the URL of a document that has more information on the inserted text
datetime= <i>datetime</i>	Date and time of a change
<kbd>....</kbd>	Sets enclosed text to display as keyboard-like input
No attributes	
<label>....</label>	Creates a label for a form control
for= <i>data</i>	Indicates the name or ID of the element to which the label is applied
<legend>....</legend>	Assigns a caption to a fieldset element, as defined by the <fieldset> tags
No attributes	
....	Defines the enclosed text as a list item in a list
value= <i>value1</i> *	Inserts or restarts counting with value1
<link />	Establishes a link between the HTML document and another document, such as an external style sheet
charset= <i>character set</i>	Specifies the character encoding of the linked resource
href= <i>url</i>	Defines the URL of the linked document
name= <i>text</i>	Names the current anchor so that it can be the destination for other links
rel= <i>relationship</i>	Indicates the relationship going from the current page to the target
rev= <i>relationship</i>	Indicates the relationship going from the target to the current page
target= <i>name</i>	Defines the name of the frame into which the linked resource will appear
type= <i>mime-type</i>	Indicates the data or media type of the linked document (for example, text/css for linked style sheets)
<map>....</map>	Specifies a client-side image map; must enclose <area> tags
name= <i>text</i>	Assigns a name to the image map
<meta />	Provides additional data (metadata) about an HTML document
content= <i>text</i>	Specifies the value for the <meta> information; required
http-equiv= <i>text</i>	Specifies the HTTP-equivalent name for metadata; tells the server to include that name and content in the HTTP header when the HTML document is sent to the client

Table A-2 HTML Tags and Attributes (continued)

HTML Tag and Attributes	Description
<meta /> (continued)	
name= <i>text</i>	Assigns a name to metadata
scheme= <i>text</i>	Provides additional context for interpreting the information in the content attribute
<noframes>....</noframes>	Defines content to be displayed in browsers that do not support frames; very important to include
No attributes	
<object>....</object>	Includes an external object in the HTML document such as an image, a Java applet, or other external object, not well-supported by most browsers
archive= <i>url</i>	Specifies the URL of the archive containing classes and other resources that will be preloaded for use by the object
classid= <i>url</i>	Specifies the URL of the embedded object
codebase= <i>url</i>	Sets the base URL for the object; helps resolve relative references
codetype= <i>type</i>	Identifies the content type of the data in the object
data= <i>url</i>	Identifies the location of the object's data
Declare	Indicates the object will be declared only, not installed in the page
height= <i>value</i>	Sets the height of the object to a value in pixels
name= <i>text</i>	Assigns a control name to the object for use in forms
standby= <i>text</i>	Defines the message to display while the object loads
tabindex= <i>value</i>	Specifies the tab order between elements, with 1 as the first element
type= <i>type</i>	Specifies the content or media type of the object
usemap= <i>url</i>	Associates an image map as defined by the <map> element
width= <i>value</i>	Sets the width of the object to a value in pixels
....	Defines an ordered list that contains numbered list item elements ()
type= <i>option</i> *	Sets or resets the numbering format for the list; options include: A=capital letters, a=lowercase letters, I=capital Roman numerals, i=lowercase Roman numerals, or 1=Arabic numerals
<option>....</option>	Defines individual options in a selection list, as defined by the <select> element
label= <i>text</i>	Provides a shorter label for the option than that specified in its content
Selected	Sets the option to be the default or the selected option in a list
value= <i>value</i>	Sets a value returned to the server when the user selects the option
Disabled	Disables the option items
<p>....</p>	Delimits a paragraph; automatically inserts a blank line between text
align= <i>position</i> *	Aligns text within the paragraph (left, center, right)
<param>....</param>	Passes a parameter to an object or applet, as defined by the <object> or <applet> element
id= <i>text</i>	Assigns an identifier to the element
name= <i>text</i>	Defines the name of the parameter required by an object
type= <i>type</i>	Specifies the content or media type of the object
value= <i>data</i>	Sets the value of the parameter
valuetype= <i>data</i>	Identifies the type of parameter used in the value attribute (data, ref, object)

Table A–2 HTML Tags and Attributes (continued)

HTML Tag and Attributes	Description
<pre>....</pre>	Preserves the original format of the enclosed text; keeps line breaks and spacing the same as the original
No attributes	
<q>....</q>	Sets enclosed text as a short quotation
lang=option	Defines the language in which the quotation will appear
<samp>....</samp>	Sets enclosed text to appear as sample output from a computer program or script; usually appears in a monospace font
No attributes	
<script>....</script>	Inserts a client-side script into an HTML document
defer	Indicates that the browser should defer executing the script
src=url	Identifies the location of an external script
type=mime-type	Indicates the data or media type of the script language (for example, text/javascript for JavaScript commands)
<select>....</select>	Defines a form control to create a multiple-choice menu or scrolling list; encloses a set of <option> tags to define one or more options
name=text	Assigns a name to the selection list
multiple	Sets the list to allow multiple selections
size=value	Sets the number of visible options in the list
disabled	Disables the selection list
tabindex=value	Specifies the tab order between list items, with 1 as the first element
<small>....</small>	Sets enclosed text to appear in a smaller typeface
No attributes	
....	Creates a user-defined container to add inline structure to the HTML document
No attributes	
<strike>....</strike> *	Sets enclosed text to appear with strong emphasis; usually displayed as bold text
No attributes	
....	Sets enclosed text to appear with strong emphasis; usually displayed as bold text
No attributes	
<style>....</style>	Encloses embedded style sheet rules for use in the HTML document
media=data	Identifies the intended medium of the style (screen, tty, tv, projection, handheld, print, braille, aural, all)
title=data	Indicates the title of the style sheet
type=data	Specifies the content or media type of the style language (for example, text/css for linked style sheets)
<sub>....</sub>	Sets enclosed text to appear in subscript
No attributes	
<sup>....</sup>	Sets enclosed text to appear in superscript
No attributes	
<table>....</table>	Marks the start and end of a table
align=position *	Aligns the table text (left, right, center, justify, char)
border=value	Sets the border around a table to a value in pixels
cellpadding=value	Sets padding around each cell's contents to a value in pixels

Table A-2 HTML Tags and Attributes (continued)

HTML Tag and Attributes	Description
<tfoot>....</tfoot> (continued)	
v-align=position	Sets vertical alignment of cells in a group (top, middle, bottom, baseline)
<th>....</th>	Defines a table header cell; contents are bold and center-aligned by default
bg-color=color *	Defines the background color for the cell
colspan=value	Defines the number of adjacent columns spanned by the cell
rowspan=value	Defines the number of adjacent rows spanned by the cell
width=n or % *	Sets the width of the table in either pixels or a percentage of the whole table width
<thead>....</thead>	Identifies and groups rows into a table header
align=position *	Specifies horizontal alignment (left, center, right, justify, char)
char=character	Specifies a character on which to align column values (for example, a period is used to align monetary values)
charoff=value	Specifies a number of characters to offset data aligned with the character specified in the char property
v-align=position	Sets vertical alignment of cells in a group (top, middle, bottom, baseline)
<title>....</title>	Defines the title for the HTML document; always should be used
No attributes	
<tr>....</tr>	Defines a row of cells within a table
bg-color=color *	Defines the background color for the cell
align=position *	Specifies horizontal alignment (left, center, right, justify, char)
char=character	Specifies a character on which to align column values (for example, a period is used to align monetary values)
charoff=value	Specifies a number of characters to offset data aligned with the character specified in the char property
v-align=position	Sets vertical alignment of cells in a group (top, middle, bottom, baseline)
<tt>....</tt>	Formats the enclosed text in teletype- or computer-style monospace font
No attributes	
<u>....</u> *	Sets enclosed text to appear with an underline
No attributes	
....	Defines an unordered list that contains bulleted list item elements ()
type=option *	Sets or resets the bullet format for the list; options include: circle, disc, square
<var>....</var>	Indicates the enclosed text is a variable's name; used to mark up variables or program arguments
No attributes	

Appendix B

Browser-Safe Color Palette

Browser-Safe Colors

Three hardware components help deliver color to a computer user: the processor, the video card, and the monitor. Because of the wide variety of components that exist, the color quality that users see varies greatly. The software on a user's computer, specifically the Web browser, also affects the way that color is displayed on a monitor. For Web developers, it is the browser that limits color significantly. It is very difficult, if not impossible, to plan for all possible color variations created by a Web browser. Using browser-safe colors allows for the browser variations, but it also limits the number of colors used on the Web page.

A total of 216 browser-safe colors appear well on different monitors, operating systems, and browsers—including both Windows and Macintosh operating systems and Internet Explorer and Mozilla Firefox browsers. When using color on your Web site, keep in mind that using only the 216 browser-safe colors can be very restrictive, especially for the approximately 10% of Web visitors who have 256-color monitors. On those monitors, only the browser-safe colors will be displayed. If you decide to use a non-browser-safe color, the visitor's browser will try to create the color by combining (a process called dithering) any number of the 216 acceptable colors. The resulting color could be slightly different from the color you had intended.

For a complete list of the 216 browser-safe colors, see Table B–1 on the next page or visit the Shelly-Cashman Series HTML Web page (scsite.com/html6e) and click Color Chart. Links to other Web sites with information about browser-safe colors also are available.

Note that you can use the color name as well as the color number when identifying a particular color to use. For instance, you can use the number #000099 (see color sample on the following page) or the word “navy” to specify the same color. Also note that to comply with XHTML standards, color names such as “navy” or “silver” must be all lowercase letters.

Table B-1 Browser-Safe Colors

#ffffff	#ffffcc	#ffff99	#ffff66	#ffff33	#ffff00
#ffccff	#ffcccc	#ffcc99	#ffcc66	#ffcc33	#ffcc00
#ff99ff	#ff99cc	#ff9999	#ff9966	#ff9933	#ff9900
#ff66ff	#ff66cc	#ff6699	#ff6666	#ff6633	#ff6600
#ff33ff	#ff33cc	#ff3399	#ff3366	#ff3333	#ff3300
#ff00ff	#ff00cc	#ff0099	#ff0066	#ff0033	#ff0000
#ccffff	#ccffcc	#ccff99	#ccff66	#ccff33	#ccff00
#ccccff	#cccccc	#cccc99	#cccc66	#cccc33	#cccc00
#cc99ff	#cc99cc	#cc9999	#cc9966	#cc9933	#cc9900
#cc66ff	#cc66cc	#cc6699	#cc6666	#cc6633	#cc6600
#cc33ff	#cc33cc	#cc3399	#cc3366	#cc3333	#cc3300
#cc00ff	#cc00cc	#cc0099	#cc0066	#cc0033	#cc0000
#99ffff	#99ffcc	#99ff99	#99ff66	#99ff33	#99ff00
#99ccff	#99cccc	#99cc99	#99cc66	#99cc33	#99cc00
#9999ff	#9999cc	#999999	#999966	#999933	#999900
#9966ff	#9966cc	#996699	#996666	#996633	#996600
#9933ff	#9933cc	#993399	#993366	#993333	#993300
#9900ff	#9900cc	#990099	#990066	#990033	#990000
#66ffff	#66ffcc	#66ff99	#66ff66	#66ff33	#66ff00
#66ccff	#66cccc	#66cc99	#66cc66	#66cc33	#66cc00
#6699ff	#6699cc	#669999	#669966	#669933	#669900
#6666ff	#6666cc	#666699	#666666	#666633	#666600
#6633ff	#6633cc	#663399	#663366	#663333	#663300
#6600ff	#6600cc	#660099	#660066	#660033	#660000
#33ffff	#33ffcc	#33ff99	#33ff66	#33ff33	#33ff00
#33ccff	#33cccc	#33cc99	#33cc66	#33cc33	#33cc00
#3399ff	#3399cc	#339999	#339966	#339933	#339900
#3366ff	#3366cc	#336699	#336666	#336633	#336600
#3333ff	#3333cc	#333399	#333366	#333333	#333300
#3300ff	#3300cc	#330099	#330066	#330033	#330000
#00ffff	#00ffcc	#00ff99	#00ff66	#00ff33	#00ff00
#00ccff	#00cccc	#00cc99	#00cc66	#00cc33	#00cc00
#0099ff	#0099cc	#009999	#009966	#009933	#009900
#0066ff	#0066cc	#006699	#006666	#006633	#006600
#0033ff	#0033cc	#003399	#003366	#003333	#003300
#0000ff	#0000cc	#000099	#000066	#000033	#000000

Appendix C

Accessibility Standards and the Web

Making the Web Accessible

Nearly 20% of the world population has some sort of disability, a physical condition that limits the individual's ability to perform certain tasks. The U.S. Congress passed the Rehabilitation Act in 1973, which promotes economic independence for those with disabilities. In 1998, Congress amended this act to reflect the latest changes in information technology. Section 508 requires that any electronic information developed, procured, maintained, or used by the federal government be accessible to people with disabilities. Disabilities that inhibit a person's ability to use the Web fall into four main categories: visual, hearing, motor, and cognitive. This amendment has had a profound effect on how Web pages are designed and developed.

Although Section 508 is specific to Web sites created and maintained by the federal government, all competent Web developers adhere to the Section 508 guidelines. It is important to include everyone as a potential user of your Web site, including those with disabilities. To ignore the needs of nearly 20% of our population is just poor practice.

The World Wide Web Consortium (W3C) developed its own set of guidelines, called the Web Accessibility Initiative (WAI), for accessibility standards. These guidelines cover many of the same issues defined in the Section 508 rules and expand on them relative to superior Web site design.

Section 508 Guidelines Examples

The 13 parts of the Section 508 guidelines are as follows:

- Subpart A—General
 - 1194.1 Purpose.
 - 1194.2 Application.
 - 1194.3 General exceptions.
 - 1194.4 Definitions.
 - 1194.5 Equivalent facilitation.

- Subpart B—Technical Standards
 - 1194.21 Software applications and operating systems.
 - 1194.22 Web-based intranet and Internet information and applications. 16 rules.
 - 1194.23 Telecommunications products.
 - 1194.24 Video and multimedia products.
 - 1194.25 Self contained, closed products.
 - 1194.26 Desktop and portable computers.
- Subpart C—Functional Performance Criteria
 - 1194.31 Functional performance criteria.
- Subpart D—Information, Documentation, and Support
 - 1194.41 Information, documentation, and support.

Web developers should review these guidelines thoroughly. We focus on the specific guidelines for intranet and Internet development in the following sections.

Sub-section **§ 1194.22** of Section 508, **Web-based intranet and Internet information and applications**, is the segment of the amendment that impacts Web design. There are 16 paragraphs within § 1194.22, which are lettered (a) through (p). These 16 paragraphs describe how each component of a Web site should be designed to ensure accessibility. The following is a list of the 16 paragraphs:

§ 1194.22 (a) A text equivalent for every non-text element shall be provided (e.g., via “alt”, “longdesc”, or in element content).

Graphical images that contain Web page content should include a text alternative (for example, using the alt or longdesc attributes). For good Web development practice, all images should include the alt attribute to describe that image, as shown in Project 2.

§ 1194.22 (b) Equivalent alternatives for any multimedia presentation shall be synchronized with the presentation.

Audio clips should contain a transcript of the content; video clips need closed captioning.

§ 1194.22 (c) Web pages shall be designed so that all information conveyed with color is also available without color, for example from context or markup.

Although color is an important component of most Web pages, you need to consider those site visitors with forms of color blindness if the color contributes significantly to the Web site content.

§ 1194.22 (d) Documents shall be organized so they are readable without requiring an associated style sheet.

Style sheets have an important role in Web development. Some browsers, however, allow users to create their own customized style sheets, which could alter the style sheets that you have designated. When developing a Web site using style sheets, ensure that the site maintains its functionality, even if your specified style sheets have been turned off.

§ 1194.22 (e) Redundant text links shall be provided for each active region of a server-side image map.

and

§ 1194.22 (f) Client-side image maps shall be provided instead of server-side image maps except where the regions cannot be defined with an available geometric shape.

This means that it is preferable for the Web developer to use client-side image maps unless the map uses a shape that the client-side will not allow. If the Web developer chooses to use server-side image maps, the developer should provide text alternatives for each link on the image map.

**§ 1194.22 (g) Row and column headers shall be identified for data tables.
and**

§ 1194.22 (h) Markup shall be used to associate data cells and header cells for data tables that have two or more logical levels of row or column headers.

You should structure your tables so that they appear in a linear fashion. In other words, the table content should be displayed one cell at a time, working from left to right across each row before moving to the next row.

§ 1194.22 (i) Frames shall be titled with text that facilitates frame identification and navigation.

Nonvisual browsers open frame sites one frame at a time. It is therefore important that the Web developer gives a name to each frame, and that the name reflects the contents of that frame. You can use either the title or the name attribute, but because nonvisual browsers differ in which attribute they use, the Web developer should use both attributes.

§ 1194.22 (j) Pages shall be designed to avoid causing the screen to flicker with a frequency greater than 2 Hz and lower than 55 Hz.

Animations on a Web page can be irritating to many people. However, they also can be quite harmful to people who have certain cognitive or visual disabilities or seizure disorders. You should therefore ensure that animations fall within the ranges stated, and you should limit the use of animations when possible. You also should make certain that necessary page content is available without the animations.

§ 1194.22 (k) A text-only page, with equivalent information or functionality, shall be provided to make a Web site comply with the provisions of this part, when compliance cannot be accomplished in any other way. The content of the text-only pages shall be updated whenever the primary page changes.

If you cannot comply with the other 15 guidelines, you should provide a text-only page to display the content of the page. You should also provide an easily accessible link to that text-only Web page.

§ 1194.22 (l) When pages utilize scripting languages to display content, or to create interface elements, the information provided by the script shall be identified with functional text that can be read by adaptive technology.

Scripts are often used to create a more interesting and dynamic Web page. You should ensure that the functionality of the script is still available for any person using nonvisual browsers.

§ 1194.22 (m) When a Web page requires that an applet, plug-in, or other application be present on the client system to interpret page content, the page must provide a link to a plug-in or applet that complies with 1994.21 (a) through (i).

Any applet or plug-in that is used on your Web pages should also comply with Section 508. The Web developer should provide a link to the applet or plug-in that is compliant with Section 508.

§ 1194.22 (n) When electronic forms are designed to be completed on-line, the form shall allow people using assistive technology to access the information, field elements, and functionality required for completion and submission of the form, including all directions and cues.

Forms need to be accessible to anyone, including those using nonvisual browsers. You should therefore include value attributes or alternative text for buttons, input boxes, and text area boxes on any form included on your Web page.

§ 1194.22 (o) A method shall be provided that permits users to skip repetitive navigation links.

It can be helpful to provide text links at the very top of a Web page so that users of nonvisual browsers can quickly link to the content of the Web site. Some Web developers use a link that allows users to skip to the main content of the Web page immediately by using a transparent image.

§ 1194.22 (p) When a timed response is required, the user shall be alerted and given sufficient time to indicate that more time is required.

Users need to be given sufficient time to react to a time-out from inactivity by notifying users that the process will soon time out. The user should then be given a way to easily request additional time.

WAI Guidelines

The WAI identifies 14 guidelines for Web developers. Within each guideline is a collection of checkpoints that identifies how to apply the guideline to specific Web site features. Each checkpoint is given a priority score that shows how much importance the WAI places on that guideline. All Web developers should review the information at the official Web site at www.w3c.org/WAI for complete information on these guidelines, and should apply the guidelines, together with the following suggestions on the application of the guidelines, to their Web page development.

The three WAI priorities are:

Priority 1: A Web content developer **must** satisfy this checkpoint. Otherwise, one or more groups will find it impossible to access information in the document. Satisfying this checkpoint is a basic requirement for some groups to be able to use Web documents.

Priority 2: A Web content developer **should** satisfy this checkpoint. Otherwise, one or more groups will find it difficult to access information in the document. Satisfying this checkpoint will remove significant barriers to accessing Web documents.

Priority 3: A Web content developer **may** address this checkpoint. Otherwise, one or more groups will find it somewhat difficult to access information in the document. Satisfying this checkpoint will improve access to Web documents.

Table C–1 contains the WAI guidelines together with the checkpoints and corresponding priority value.

Table C–1

WAI Guidelines and Checkpoints	Priority
1. Provide equivalent alternatives to auditory and visual content.	
1.1 Provide a text equivalent for every non-text element (e.g., via “alt”, “longdesc”, or in element content). <i>This includes:</i> images, graphical representations of text (including symbols), image map regions, animations (e.g., animated GIFs), applets and programmatic objects, ASCII art, frames, scripts, images used as list bullets, spacers, graphical buttons, sounds (played with or without user interaction), standalone audio files, audio tracks of video, and video.	1
1.2 Provide redundant text links for each active region of a server-side image map.	1
1.3 Until user agents can automatically read aloud the text equivalent of a visual track, provide an auditory description of the important information of the visual track of a multimedia presentation.	1
1.4 For any time-based multimedia presentation (e.g., a movie or animation), synchronize equivalent alternatives (e.g., captions or auditory descriptions of the visual track) with the presentation.	1
1.5 Until user agents render text equivalents for client-side image map links, provide redundant text links for each active region of a client-side image map.	3
2. Don’t rely on color alone.	
2.1 Ensure that all information conveyed with color is also available without color; for example, from context or markup.	1
2.2 Ensure that foreground and background color combinations provide sufficient contrast when viewed by someone having color deficits or when viewed on a black and white screen.	2
3. Use markup and style sheets and do so properly.	
3.1 When an appropriate markup language exists, use markup rather than images to convey information.	2
3.2 Create documents that validate to published formal grammars.	2
3.3 Use style sheets to control layout and presentation.	2
3.4 Use relative rather than absolute units in markup language attribute values and style sheet property values.	2
3.5 Use header elements to convey document structure and use them according to specification.	2
3.6 Mark up lists and list items properly.	2
3.7 Mark up quotations. Do not use quotation markup for formatting effects such as indentation.	2
4. Clarify natural language usage.	
4.1 Clearly identify changes in the natural language of a document’s text and any text equivalents (e.g., captions).	1
4.2 Specify the expansion of each abbreviation or acronym in a document where it first occurs.	3
4.3 Identify the primary natural language of a document.	3

Table C-1 (continued)

WAI Guidelines and Checkpoints	Priority
5. Create tables that transform gracefully.	
5.1 For data tables, identify row and column headers.	1
5.2 For data tables that have two or more logical levels of row or column headers, use markup to associate data cells and header cells.	1
5.3 Do not use tables for layout unless the table makes sense when linearized. Otherwise, if the table does not make sense, provide an alternative equivalent (which may be a linearized version).	2
5.4 If a table is used for layout, do not use any structural markup for the purpose of visual formatting.	2
5.5 Provide summaries for tables.	3
5.6 Provide abbreviations for header labels.	3
6. Ensure that pages featuring new technologies transform gracefully.	
6.1 Organize documents so they may be read without style sheets. For example, when an HTML document is rendered without associated style sheets, it must still be possible to read the document.	1
6.2 Ensure that equivalents for dynamic content are updated when the dynamic content changes.	1
6.3 Ensure that pages are usable when scripts, applets, or other programmatic objects are turned off or not supported. If this is not possible, provide equivalent information on an alternative accessible page.	1
6.4 For scripts and applets, ensure that event handlers are input-device-independent.	2
6.5 Ensure that dynamic content is accessible or provide an alternative presentation or page.	2
7. Ensure user control of time-sensitive content changes.	
7.1 Until user agents allow users to control flickering, avoid causing the screen to flicker.	1
7.2 Until user agents allow users to control blinking, avoid causing content to blink (i.e., change presentation at a regular rate, such as turning on and off).	2
7.3 Until user agents allow users to freeze moving content, avoid movement in pages.	2
7.4 Until user agents provide the ability to stop the refresh, do not create periodically auto-refreshing pages.	2
7.5 Until user agents provide the ability to stop auto-redirect, do not use markup to redirect pages automatically. Instead, configure the server to perform redirects.	2
8. Ensure direct accessibility of embedded user interfaces.	
8.1 Make programmatic elements such as scripts and applets directly accessible or compatible with assistive technologies (Priority 1 if functionality is important and not presented elsewhere, otherwise Priority 2).	2
9. Design for device-independence.	
9.1 Provide client-side image maps instead of server-side image maps except where the regions cannot be defined with an available geometric shape.	1
9.2 Ensure that any element that has its own interface can be operated in a device-independent manner.	2
9.3 For scripts, specify logical event handlers rather than device-dependent event handlers.	2
9.4 Create a logical tab order through links, form controls, and objects.	3
9.5 Provide keyboard shortcuts to important links (including those in client-side image maps), form controls, and groups of form controls.	3

Table C-1 (continued)

WAI Guidelines and Checkpoints	Priority
10. Use interim solutions.	
10.1 Until user agents allow users to turn off spawned windows, do not cause pop-ups or other windows to appear and do not change the current window without informing the user.	2
10.2 Until user agents support explicit associations between labels and form controls, for all form controls with implicitly associated labels, ensure that the label is properly positioned.	2
10.3 Until user agents (including assistive technologies) render side-by-side text correctly, provide a linear text alternative (on the current page or some other) for <i>all</i> tables that lay out text in parallel, word-wrapped columns.	3
10.4 Until user agents handle empty controls correctly, include default, place-holding characters in edit boxes and text areas.	3
10.5 Until user agents (including assistive technologies) render adjacent links distinctly, include non-link, printable characters (surrounded by spaces) between adjacent links.	3
11. Use W3C technologies and guidelines.	
11.1 Use W3C technologies when they are available and appropriate for a task and use the latest versions when supported.	2
11.2 Avoid deprecated features of W3C technologies.	2
11.3 Provide information so that users may receive documents according to their preferences (e.g., language, content type, etc.).	3
11.4 If, after best efforts, you cannot create an accessible page, provide a link to an alternative page that uses W3C technologies, is accessible, has equivalent information (or functionality), and is updated as often as the inaccessible (original) page.	1
12. Provide context and orientation information.	
12.1 Title each frame to facilitate frame identification and navigation.	1
12.2 Describe the purpose of frames and how frames relate to each other if it is not obvious by frame titles alone.	2
12.3 Divide large blocks of information into more manageable groups where natural and appropriate.	2
12.4 Associate labels explicitly with their controls.	2
13. Provide clear navigation mechanisms.	
13.1 Clearly identify the target of each link.	2
13.2 Provide metadata to add semantic information to pages and sites.	2
13.3 Provide information about the general layout of a site (e.g., a site map or table of contents).	2
13.4 Use navigation mechanisms in a consistent manner.	2
13.5 Provide navigation bars to highlight and give access to the navigation mechanism.	3
13.6 Group related links, identify the group (for user agents), and, until user agents do so, provide a way to bypass the group.	3
13.7 If search functions are provided, enable different types of searches for different skill levels and preferences.	3
13.8 Place distinguishing information at the beginning of headings, paragraphs, lists, etc.	3
13.9 Provide information about document collections (i.e., documents comprising multiple pages).	3
13.10 Provide a means to skip over multi-line ASCII art.	3

Table C-1 (continued)

WAI Guidelines and Checkpoints	Priority
14. Ensure that documents are clear and simple.	
14.1 Use the clearest and simplest language appropriate for a site's content.	1
14.2 Supplement text with graphic or auditory presentations where they will facilitate comprehension of the page.	3
14.3 Create a style of presentation that is consistent across pages.	3

Appendix D

CSS Properties and Values

This appendix provides a brief review of Cascading Style Sheets (CSS) concepts and terminology, and lists CSS level 1 and 2 properties and values supported by most browsers.

CSS Concepts and Terminology

CSS supports three types of style sheets: inline, embedded, and external (or linked). A **style** is a rule that defines the appearance of an element on a Web page. Inline styles are used to change the appearance (or style) for individual elements, such as a heading or a paragraph. A **style sheet** is a series of rules that defines the style for a Web page or an entire Web site. The **style statement** changes that specific element, but does not affect other elements in the document. With an embedded style sheet, you add the style sheet within the `<style></style>` container between the `<head></head>` tags of the HTML document. An embedded style sheet allows you to define the style for an entire Web page. With a linked, or external, style sheet, you create a text file that contains all of the styles that you want to apply, and save the text file with the file extension .css. You then add a link to this external style sheet on any Web page in the Web site in which you want to use those styles. External style sheets give you the most flexibility and are ideal to apply the same formats to all of the Web pages in a Web site. External style sheets also make it easy to change formats quickly across Web pages.

The part of the style statement that identifies the page element that you want to change is called the selector. In the example below, the selector is the h1 (heading size 1) element. The part of the style statement that identifies how the element(s) should appear is called the declaration. In this example, the declaration is everything between the curly brackets `{color: red}`. This includes the property named color and the value named red. You could use the statement below in both an embedded and an external style sheet. With an external (linked) style sheet, you save the file with the selectors and declarations as a .css file. You then link that file into any Web page into which you want those styles to apply.

```
h1 {color: red}
```

To add the same style statement into an embedded style sheet, you have to put the selector and declaration within the `<style></style>` container as shown below. This code would be inserted within the `<head></head>` container.

```
<style type="text/css">
h1 {color: red}
</style>
```

These two style sheets (embedded and external) give you the most flexibility. For instance, if you want all h1 and h2 headings to be the color red, you would simply add the code below:

```
h1, h2 {color: red}
```

You also could make the style change to the h1 heading as an inline style. You insert that code within the body of the Web page in the following format:

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

Although this can be very useful, understand that you would have to insert the same declaration for every h1 (or h2 or h3) heading within the Web page. That makes the inline style less flexible than the other style sheets.

As shown in Table D–1, the three style sheets supported by CSS control the appearance of a Web page at different levels. Each style sheet type also has a different level of precedence or priority in relationship to the others. An external style sheet, for example, is used to define styles for multiple pages in a Web site. An embedded style sheet is used to change the style of one Web page, but overrides or takes precedence over any styles defined in an external style sheet. An inline style sheet is used to control the style within an individual HTML tag and takes precedence over the styles defined in both embedded and external style sheets.

Table D–1 CSS Precedence

Type	Level and Precedence
Inline	<ul style="list-style-type: none"> • To change the style within an individual HTML tag • Overrides embedded and external style sheets
Embedded	<ul style="list-style-type: none"> • To change the style of one Web page • Overrides external style sheets
External	<ul style="list-style-type: none"> • To change the style of multiple pages in a Web site

Because style sheets have different levels of precedence, all three types of style sheets can be used on a single Web page. For example, you may want some elements of a Web page to match the other Web pages in the Web site, but you also may want to vary the look of certain sections of that Web page. You can do this by using the three types of style sheets.

A newer version of Cascading Style Sheets, CSS3, is currently being defined, but is not covered in this appendix. CSS3 utilizes a modularized approach to style sheets, which allows CSS to be updated in a more timely and flexible manner.

For a more comprehensive list of CSS properties and values, see the www.w3.org Web site. In addition to an abundance of information about CSS levels 1 and 2, the w3 site also has extensive information about CSS3, from its history to its use with browsers today. The Web site also includes many online tutorials for learning CSS levels 1 and 2 as well as CSS3.

CSS Properties

Tables D–2 through D–10 show the property names, descriptions, and valid values for various categories of CSS properties. Values listed in bold are the default.

Acceptable Units of Measure

Many of the properties below use units of measure for their attribute values. Table D–2 lists the acceptable units of measure that can be used.

Table D–2 Units of Measure

Property Name	Description	Values
color	A color is either a keyword or a numerical RGB specification	[keyword – aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, and yellow] [#rrggbb]
length	Indicates both relative (em, ex, px) and absolute (in, cm, mm, pt, pc) lengths	em – relative to size of capital M of browser default font ex – relative to small x of browser default font px – represents one pixel, smallest unit of measure in – one inch cm – one centimeter mm – one millimeter pt – 1/72 of an inch pc – 1/12 of an inch
percentage	Values are always relative to another value; default if not defined	percentage of width or height of parent element

Background and Color Styles

Colors and subtle backgrounds can enhance the style of a Web page significantly. You can set the background or color of an element using these style sheet properties. Not all browser versions support these style attributes, however, so be aware that not all users will be able to see the background and color styles set by these properties. Table D–3 provides a list of background and color properties.

Table D–3 Background and Color Properties

Property Name	Description	Values
background	The background property is a shorthand property for setting the individual background properties (i.e., background-color, background-image, 'background-repeat', 'background-attachment, and background-position) at the same place in the style sheet.	
background-attachment	Sets the background image to fixed, or scrolls with the page	scroll fixed
background-color	Sets the background color of an element	transparent [color]
background-image	Sets an image as the background	none [url]
background-position	Sets the starting position of a background image	[length] [percentage] bottom center left right top
background-repeat	Sets if/how a background image will be repeated	repeat repeat-x repeat-y no-repeat
color	Sets the foreground color of an element	[color] transparent

Border Styles

Many changes can be made to the style, color, and width of any or all sides of a border using the border properties listed in Table D–4. Using the border-color, border-width, or border-style border properties allows you to set the style for all sides of a border. Using style properties such as border-top-width, border-right-color, or border-bottom-style gives you the option to set the width, color, or style for only the top, right, bottom, or left border of a table cell. If you do not make changes to the border style using style sheet properties, the default border will be displayed.

Table D–4 Border Properties

Property Name	Description	Values
border-color	Sets the color of the four borders; can have from one to four colors	[color] transparent
border-top-color border-right-color border-bottom-color border-left-color	Sets the respective color of the top, right, bottom, and left borders individually	[color]
border-style	Sets the style of the four borders; can have from one to four styles	none dashed dotted double groove inset outset ridge solid
border-top-style border-right-style border-bottom-style border-left-style	Sets the respective style of the top, right, bottom, and left borders individually	none dashed dotted double groove inset outset ridge solid
border-width	Shorthand property for setting the width of the four borders in one declaration; can have from one to four values	medium [length] thick thin
border-top-width border-right-width border-bottom-width border-left-width	Sets the respective width of the top, right, bottom, and left borders individually	medium [length] thick thin

Classification Styles

These properties classify elements into categories more than they set specific visual parameters. Table D–5 lists common classification properties that can be used.

Table D–5 Classification Properties

Property Name	Description	Values
display	Describes how/if an element is displayed on the canvas, which may be on a printed page, a computer monitor, etc.	block inline list-item none
white-space	Declares how whitespace inside the element is handled: the ‘normal’ way (where whitespace is collapsed), as <i>pre</i> (which behaves like the <code><pre></code> element in HTML) or as <i>nowrap</i> (where wrapping is done only through <code>
</code> elements)	normal <i>pre</i> <i>nowrap</i>

Font Styles

An element’s font can be changed using the `font` attribute and various font properties. When you set the font family for an element, you can set one or more fonts or font families by using a comma-delimited list. Each font family generally includes several font definitions. For example, the Arial font family includes Arial Black and Arial Narrow. If you specify more than one font, the browser assesses the user’s system and uses the first font family installed on the system. If the system has none of the font families specified in the style sheet, the browser uses the default system font. Table D–6 lists common font properties.

Table D–6 Font Properties

Property Name	Description	Values
font	Shorthand property for setting font-style, font-variant, font-weight, font-size, line-height, and font-family at the same place in the style sheet	
font-family	A prioritized list of font-family names and/or generic family names for an element	[family-name] cursive fantasy monospace sans-serif serif
font-size	Sets the size of a font	[length] [percentage] large medium small x-large x-small xx-large xx-small
font-style	Sets the style of a font	normal italic oblique
font-variant	Displays text in a small-caps font or a normal font	normal small-caps
font-weight	Sets the weight of a font	normal bold bolder lighter

List Styles

Using the properties associated with list styles allows you to set the kind of marker that identifies a list item. An unnumbered list marker, for example, can be a filled disc, an empty circle, or a square. A numbered list marker can be a decimal, lower-alpha, lower-Roman numeral, upper-alpha, or upper-Roman numeral. Table D–7 provides compatible browser list properties.

Table D–7 List Properties

Property Name	Description	Values
list-style-image	Sets an image as the list-item marker	none url
list-style-position	Indents or extends a list-item marker with respect to the item's content	outside inside
list-style-type	Sets the type of list-item marker	disc circle square decimal lower-alpha lower-Roman upper-alpha upper-Roman

Margin and Padding Styles

Many changes can be made to the width and spacing around an element using the margin and padding properties listed in Table D–8. Padding is the space that occurs between the edge of an element and the beginning of its border. If you increase padding around an element, you add space inside its border. The border, therefore, has a larger area to cover.

You can use the margin or padding property to set the widths of margins and padding amounts along all four sides of an element. Using margin and padding properties such as margin-top, margin-right, padding-left, or padding-bottom gives you the option to set the margin or padding for only the top, right, bottom, or left side of an element.

Table D–8 Margin and Padding Properties

Property Name	Description	Values
margin	Shorthand property for setting margin-top, margin-right, margin-bottom, and margin-left at the same place in the style sheet.	
margin-top margin-right margin-bottom margin-left	Sets the top, right, bottom, and left margin of an element individually	[length] [percentage] auto
padding	Shorthand property for setting padding properties in one declaration	
padding-top padding-right padding-bottom padding-left	Sets the top, right, bottom, and left padding of an element individually	[length] [percentage]

Miscellaneous Formatting Styles

Other changes can be made to the width and spacing around elements using the properties listed in Table D–9. These properties can be applied to text elements. They are most useful, however, with elements such as images.

You can use the width or height property to set the widths or heights of images or text, or the size of the box. The float and clear properties alter the position of the element on a Web page. You can float images on a Web page and clear the sides where floating elements are not accepted.

Table D–9 Miscellaneous Formatting Properties

Property Name	Description	Values
width	Can be applied to text elements, but it is most useful with elements such as images	[length] [percentage] auto
height	Can be applied to text elements, but it is most useful with elements such as images	[length] auto
float	With the value none, the element will be displayed where it appears in the text; with a value of left (right) the element will be moved to the left (right) and the text will wrap on the right (left) side of the element	left right none
clear	Specifies if an element allows floating elements on its sides	none left right both

Text Styles

Text styles can be used to change the letter-spacing, alignment, line-height (not recommended), and text decoration, along with other text properties. The text-transform property can change text into all uppercase, all lowercase, or be used to change the first letter of each word to uppercase. With text-align, you can align text left, right, center, or justify the text. The text style properties are listed in Table D–10.

Table D–10 Text Properties

Property Name	Description	Values
letter-spacing	Increases or decreases the space between characters	normal [length]
line-height	Sets the spacing between text baselines	normal [length] [number] [percentage]
text-align	Aligns the text in an element	left right center justify
text-decoration	Adds decoration to text	none blink line-through overline underline

Table D–10 Text Properties (Continued)

Property Name	Description	Values
text-indent	Indents the first line of text in an element	[length] [percentage]
text-transform	Controls text capitalization	none capitalize lowercase uppercase
vertical-align	Sets the vertical positioning of text	baseline [length] [percentage] bottom middle sub super text-bottom text-top top
word-spacing	Increases or decreases the space between words	normal [length]

This page intentionally left blank

Appendix E

Publishing Web Pages to a Web Server

Publishing your Web site means transferring your files to a Web server (Web host) that will make your pages available 24/7 on the Web. Publishing involves two basic steps: choosing a Web host and uploading your Web site files (usually via FTP) to that host.

Choosing a Web Host

There are many options available for Web hosting, as detailed in the “Finding a Web Hosting Site” section of the Special Feature on “Attracting Visitors to Your Web Site.” Common options are to use the ISP that you use to connect to the Internet or to use a Web hosting service.

Your Internet service provider (ISP) may provide space for its clients to host a Web site. If it does, you should contact your network system administrator or technical support staff at your ISP to determine if their Web server supports FTP, and to obtain necessary permissions to access the Web server. There are other options for hosting Web sites as well. You can search for free Web hosts using any browser. Whatever Web host you choose, you must secure a username and password in order to gain access to the host.

Uploading Files to the Host

Once you have chosen a Web host, you’ll need a program to transfer your files to the Web server. The most common file transfer program is called FTP. **File Transfer Protocol (FTP)** is an Internet standard that allows computers to exchange files with other computers on the Internet. FTP was developed to promote sharing files across a variety of computers reliably and efficiently. FTP programs that run on PCs are sometimes called FTP clients.

There are many FTP programs available for free on the Internet. Search for FTP in any browser and you will find a variety of programs. Read the documentation to assure that the program works with your computer and operating system. Then download and install the FTP program of your choice.

Now you can use your FTP program to upload your Web pages to the server. Be sure to include all HTML files, CSS files, and any graphic files that make up your Web site. Table E-1 shows the steps to use FTP to upload your files. All of the necessary information (e.g., username) should be provided to you by the network administrator for the Web host. Specific keystrokes or mouse clicks to accomplish each step may vary among FTP clients.

Table E-1 Using FTP to Upload Your Web Files

1. Start your FTP program.
2. Type in the host name/address as provided by the network administrator.
3. Select the host type.
4. Enter the user ID and password that you were given and then click the OK button.
5. You should see both a local system (your computer) as well as the remote system (the Web host) in the FTP dialog box.
6. Use the up arrow to find the folders in which you stored your files; use the up arrow to locate the folders on the remote system also.
7. Highlight the files that you want to upload and click the right arrow to move the files to the remote system.

Appendix F

Symbols and Characters

Quick Reference

Using Symbols and Special Characters

There is a way to insert special characters into your HTML and XHTML code by using the entity character reference. Tables F-1 through F-3 contain the most commonly used entity characters, mathematical and technical characters, and arrow characters. Several projects in the book use the predefined entity characters to enhance content on the Web pages that you create.

You can find a complete list of characters at www.unicode.org.

Table F-1 Most Commonly Used Predefined Entity Characters

Symbol	Entity	Description
&	&	Ampersand
	¦	Broken vertical bar
¢	¢	Cent sign
©	©	Copyright sign
¤	¤	Currency sign
†	†	Dagger
‡	‡	Double dagger
€	€	Euro
>	>	Greater-than sign
«	«	Left-pointing double angle quotation mark
<	<	Less-than sign
—	—	Em dash
[Tab]	 	Non-breaking space (can be used for tabbing on a Web page)
—	–	En dash
¬	¬	Not sign
¶	¶	Paragraph sign
£	£	Pound
“	"	Quotation mark = APL quote
®	®	Registered mark sign
»	»	Right-pointing double angle quotation mark
§	§	Section sign
™	™	Trademark sign
¥	¥	Yen

Table F-2 Mathematical and Technical Characters

Symbol	Entity	Description
\wedge	∧	Logical and
\angle	∠	Angle
\approx	≈	Almost equal to
\cap	∩	Intersection
\cup	∪	Union
\circ	°	Degree sign
\div	÷	Division sign
\equiv	≡	Identical to
\exists	∃	There exists
f	ƒ	Function
\forall	∀	For all
$\frac{1}{2}$	&fract12;	Fraction one half
$\frac{1}{4}$	&fract14;	Fraction one quarter
$\frac{3}{4}$	&fract34;	Fraction three quarters
\geq	≥	Greater-than or equal to
∞	∞	Infinity
\int	∫	Integral
\in	∈	Element of
\leq	≤	Less-than or equal to
μ	µ	Micro sign
∇	∇	Backward difference
\neq	≠	Not equal to
\ni	∋	Contains as a member
∂	∂	Partial differential
\perp	⊥	Perpendicular
\pm	±	Plus-minus sign
\prod	∏	n-ary product
\propto	∝	Proportional to
$\sqrt{ }$	√	Square root
\sim	∼	Tilde
\sum	∑	n-ary summation
\therefore	∴	Therefore

Table F-3 Arrow Characters

Symbol	Entity	Description
\downarrow	↓	Downward arrow
\leftrightarrow	↔	Left right arrow
\leftarrow	←	Leftward arrow
\rightarrow	→	Rightward arrow
\uparrow	↑	Upward arrow

Index

Note: Page numbers in boldface indicate key terms.

Symbols

<!----> tags: comment codes, HTML 119, APP 2
<!DOCTYPE> tag (DOCTYPE statement), HTML 39, HTML 40, HTML 43, HTML 255, HTML 271, APP 2
. (period): class name symbol, HTML 131, HTML 178, HTML 180
. (periods): move up folder symbol, HTML 113
| (pipe symbol): text link separator, HTML 175, HTML 180
(pound sign): link address symbol, HTML 98, HTML 139

A

absolute path, **HTML 112–113**
Access data: Web pages from, HTML 14
accessibility standards, HTML 21
researching, HTML 31
Section 508 guidelines, HTML 16, APP 13–16
WAI guidelines, APP 13, APP 16, APP 17–20
Acrobat files: converting into HTML documents/files, HTML 14
action attribute of <form> tag, **HTML 296**
ActiveX controls, HTML 370

alternate text for images, HTML 64
 See also alt attribute
Amazon.com, HTML 5
American Disabilities Act (ADA) standards
 for Web site frames, HTML 394
ampersand (&), HTML 307
analyzing Web sites, HTML 16
 questions regarding, HTML 15
anchor element. *See* a element
anchor tag. *See* <a> tag
Apple:
 audio-file formats used by, 368
 iMovie software for, HTML 366
 plug-ins of, 367
 QuickTime (.mov) video file format, HTML 369, HTML 370; commonly used parameters, HTML 372–373
 See also Mac computers
application documents: converting into HTML documents/files, HTML 14
<area> tag, APP 2
 in image maps, HTML 248, HTML 252, HTML 265
arrow characters, APP 34
<a> tag (anchor tag), HTML 95–96, HTML 97, HTML 98, HTML 99, HTML 127, APP 2
attributes, HTML 107–108, APP 2
 (See also href attribute); name attribute, HTML 107, HTML 139, APP 2;
 subject and body attributes, HTML 110
attributes (of HTML tags), **HTML 8–9**, APP 2–10
 deprecated attributes, **HTML 11**, APP 2
 image attributes, HTML 66
 obsolete attributes, APP 2
 Web site resources, HTML 10, HTML 11, HTML 24
XHTML coding standards, HTML 12, APP 1
attribute values, APP 2–10
 XHTML coding standard, HTML 12, APP 1
AU (.au) audio file format, HTML 368
audio file formats, HTML 368
adding to Web pages, HTML 369–378;
 Apple QuickTime parameters, HTML 372–373; background sounds, HTML 385–386; common formats, HTML 368; creating music, HTML 388; object statement HTML code, HTML 375; object tag, **HTML 369–370**; object tag parameters, HTML 372–373; planning phases, HTML 373; playing simultaneously with video files, HTML 389–390; validating/viewing using ActiveX controls, HTML 376–378; Windows Media Player parameters, HTML 372
common formats, HTML 368
autoplay parameter (for Apple QuickTime), HTML 372

autoplay parameter (for Windows Media Player), HTML 372
AVI (.avi) audio/video file format, HTML 369

B

background properties:
 options, HTML 118
 and values, APP 24
backgrounds, **HTML 37**
balance parameter (for Windows Media Player), HTML 372
banner images:
 adding, HTML 102–103
 logo banner, HTML 102, HTML 171
<base /> tag, APP 2
baseUrl parameter (for Windows Media Player), HTML 372
bgcolor parameter (for Apple QuickTime), HTML 372
<big> tag, HTML 115, APP 3
blank lines, HTML 10
 inserting, HTML 45, HTML 47–48, HTML 134
blink property value, APP 28
<blockquote> tag, HTML 115, APP 3
body (of an HTML document), HTML 10, **HTML 37**
 styles, HTML 183
body attribute (<a> tag), HTML 110
<body> tag, HTML 39, HTML 40, HTML 44, HTML 256, APP 3
body text, HTML 45
bolding text:
 with an inline style, HTML 52
 with bold tag, HTML 115–116
bold tag. *See* tag
bookmarks (favorites), HTML 37
border colors for image links, HTML 94
 changing, HTML 95
borderless tables, HTML 162, HTML 171
border properties:
 defined, **HTML 343**
 options, HTML 118
 and values, APP 25
borders:
 image link border colors, HTML 94;
 changing, HTML 95
table borders, **HTML 165**, HTML 170
tables with, HTML 192; creating, HTML 193–194
box model, **HTML 343**
Bright Idea Web site, HTML 234–235, HTML 243, HTML 245, HTML 248, HTML 255, HTML 257, HTML 261, HTML 266–267, HTML 283
broad Web sites, HTML 19–20
browsers. *See* Web browsers
browser-safe colors, HTML 68, APP 11–12

 tag, HTML 40, APP 3

 tag, HTML 115, APP 2
 bulleted lists. *See* unordered lists
 bulleting paragraphs, HTML 45
bullets, HTML 45
 changing, HTML 51
 images as, HTML 105–106
 options, HTML 49, APP 27
 businesses (companies): Web site uses, HTML 5, HTML 15
See also users
 button bar, on image maps, **HTML 238**

C

capitalization: property and values, APP 28, APP 29
 captions. *See* table captions
<caption> tag, APP 3
 Cascading Style Sheets (CSS), HTML 2, **HTML 11**, HTML 21
 basic concepts/terminology, HTML 62–64, APP 21–22
 code for external style sheet, HTML 271
 creating a pop-up using, HTML 344–345; applying text, HTML 358
 determining Web site use, HTML 359
 finding information online about, HTML 358
 measurement values for, HTML 339, HTML 340
 newer version (CSS3), APP 23
 planning phase, HTML 328, HTML 332
 position property values, HTML 340
 precedence review, HTML 332
 properties and values, HTML 271
 pseudoclasses, HTML 339
 researching, HTML 31
 Sapperzein Galleries Web site project.
See Sapperzein Galleries Web site project
 saving and printing, 272
 style experimentation, HTML 358
 types of, HTML 329–332
 uses for, HTML 329
See also embedded style sheets; external style sheets; inline style sheets; properties and values (CSS); styles; style sheets; style statements
 case for tags, HTML 39, HTML 44
cellpadding, HTML 197, HTML 198
 adding, HTML 199
cells (in tables), HTML 163–164
cellspacing, HTML 197, HTML 198
 adding, HTML 199
 centering images, HTML 173
<center> tag, HTML 173, APP 3
CGI scripts, HTML 260
 defined, **HTML 296**
 in form creation, HTML 294, HTML 296, HTML 307
 character encoding: UTF-8, HTML 39
 check box (checkbox), HTML 289, HTML 290
 adding to forms on Web pages, HTML 300–301
check box control, HTML 290–291
circle value, HTML 265
<cite> tag, APP 3
 classes (in style statements) (CSS), **HTML 130–131**
 in embedded style sheets, HTML 130–132
 in external style sheets, HTML 178–179
 IDs for ActiveX controls, **HTML 370**
 menu bar classes, HTML 174, HTML 175, HTML 180, HTML 185
 naming, HTML 131, HTML 178
 pseudo-classes, **HTML 119**, HTML 131
 researching, HTML 218
 wrapping text around images with, HTML 133–134, HTML 136–137; stopping/clearing wrapping, HTML 138
classid attribute
 choosing, HTML 375
 code for adding audio clip, HTML 370, HTML 373, HTML 375
 code for adding video clip, HTML 370, HTML 379, HTML 382, HTML 383
 defined, **HTML 370**
 class IDs for ActiveX controls:
 in Apple Quicktime, HTML 373
 choice determination, HTML 375
 HTML code: for adding to audio clip, HTML 375; for adding to video clip, HTML 382
 in Windows Media Player, **HTML 370**, HTML 379
 classification properties: and values, APP 26
 class name symbol (.), HTML 131, HTML 178, HTML 180
 clear property and values, APP 28
 clickable areas, in hotspots and image maps, HTML 234, HTML 237, HTML 243, HTML 244, HTML 247–248, HTML 251–252, HTML 254, HTML 265–266, HTML 268
See also hotspots (in images)
client-side image map, HTML 242–243, HTML 248, **HTML 260–261**
 closing Notepad++, HTML 80, HTML 146, HTML 206
 closing tags, HTML 12, HTML 40, APP 1
<code> tag, APP 3
 coding standards/practices:
 HTML, HTML 10, HTML 11; organizational standards, HTML 16–17, HTML 27, HTML 28
 modification in Notepad++, HTML 273
 UTF-8 character encoding, HTML 39
 XHTML, HTML 12, HTML 39, APP 1
 Coffee Cup Image Mapper, HTML 254
<colgroup> tag, APP 3
 color codes, common, HTML 68
 coloring headings, HTML 67, HTML 68, HTML 114
 with embedded style sheets, HTML 120
 with inline styles, HTML 70, HTML 121
color property and values, APP 23
 background color, APP 24
 font color, HTML 114
 colors, for links. *See* link colors, property.
See also color property and values
 browser-safe colors, HTML 68, APP 11–12
 common color codes, HTML 68
 finding, HTML 183
 predefined colors, HTML 68
cols: as textarea control attribute, **HTML 291**
colspan attribute (<th>/<td> tag), HTML 200
<col> tag, APP 3
columns (in tables), HTML 163–164
 rules between, HTML 166
 spanning, HTML 168, HTML 200, HTML 202
 column spanning, HTML 168, HTML 200, HTML 202
 comment codes (<!...>), HTML 119, APP 2
Common Gateway Interface (CGI)
 scripting. *See* CGI scripts
 companies. *See* businesses
 compatibility testing (of Web sites on browsers), HTML 22–23, HTML 58
 containers (tag sets):
 <div>/<div> tags (external style sheets), HTML 343
<head>/<head> tags (HTML documents), HTML 39, HTML 40, HTML 42, HTML 179
<object>/<object> tags: for audio clips, HTML 373; for video clips, HTML 379, HTML 381, HTML 382
/ tags (inline styles), HTML 121
<style>/<style> tags (embedded style sheets), HTML 117, HTML 161, HTML 310, HTML 366
content (of Web sites/pages):
 defining, HTML 16
 entering, HTML 45–52
 organizational standard, HTML 17, HTML 28
 testing, HTML 22–23, HTML 125, HTML 145, HTML 189, HTML 313
 testing links, HTML 125–126, HTML 178, HTML 196
 content owners and authors: identifying, HTML 16
 controller parameter (for Apple QuickTime), HTML 372

converting application documents into HTML documents/files, HTML 14
coordinate pair (x- and y- coordinates), **HTML 246**
coordinating headings with graphics, HTML 102, HTML 105
coords attribute, **HTML 248**, **HTML 265**
copying and pasting HTML code, HTML 143, HTML 190
CSS. *See* Cascading Style Sheets
.css extension, HTML 63, HTML 329, APP 21

D

data cells (in tables), **HTML 163**
data input control, **HTML 289**
dates of modification: including, HTML 17
<dd> tag, HTML 52, APP 4
declaration (in style statements), **HTML 63**, **HTML 117**, HTML 118, HTML 130, APP 21
deep Web sites, HTML 20–21
definition lists, **HTML 51–52**
creating, HTML 52, HTML 82–83
deprecated tags/attributes, **HTML 11**, **HTML 39**, HTML 115, APP 2
designing and developing Web sites, HTML 16–21, HTML 29–30, HTML 32, HTML 161
questions regarding, HTML 15
using logs, HTML 24
digital images. *See* images
disability law. *See* Section 508 (Rehabilitation Act)
disabled attribute, HTML 294
dividing Web pages into sections, HTML 65, HTML 102–103
<div> tag, HTML 65, HTML 102–103, HTML 343, APP 4
<dl> tag, HTML 52, APP 4
DOCTYPE statement. *See* <!DOCTYPE> tag
document header, HTML 39
Document Object Model (DOM), **HTML 11–12**
Document Type Definition (DTD), **HTML 39**
DOM (Document Object Model), **HTML 11–12**
double period (..): move up folder symbol, HTML 113
drop-down menu, HTML 327
adding to embedded style sheets, HTML 329, HTML 335–340
home page example, HTML 330
moving code, HTML 333
DTD (Document Type Definition), **HTML 39**
<dt> tag, HTML 52, APP 4
DVD Studio Pro software, HTML 366

E

e-commerce, **HTML 5**
editors for creating HTML documents/files, HTML 13–14, HTML 24
e-mail addresses:
including, HTML 17
linking to, HTML 98, HTML 99, HTML 109
See also e-mail links
e-mail links, **HTML 98–99**
adding, HTML 98, HTML 99, HTML 109
adding subjects and information to, HTML 109–110
from image maps, HTML 237, HTML 245
embedded style sheets, HTML 62, **HTML 63**, HTML 116, HTML 117–120, APP 21, APP 22
adding, HTML 119–120, HTML 332, HTML 340–341
adding a drop-down menu, HTML 335–340
adding a navigation menu, HTML 334–342
classes in, HTML 130–132
code, HTML 335, HTML 336–337, HTML 339, HTML 340–341
containers for, HTML 117, HTML 161
creating, HTML 333
creating Web pages with, HTML 117–120, HTML 130–132
external style sheets vs., HTML 178
in form creation for Web pages, HTML 312–313
precedence, HTML 63, HTML 116, HTML 119, APP 22
setting body and link style, HTML 335–340
shorthand property, **HTML 337**
style statements in, HTML 117, HTML 118–119
uses for, HTML 329–330; single Web page styles, HTML 332
vs. inline styles, HTML 64, HTML 117, HTML 180
wrapping text around images with, HTML 133–134, HTML 136–137; stopping/clearing wrapping, HTML 138
<embed> tag, HTML 367
emphasizing important information, HTML 17, HTML 28
 tag, HTML 115, APP 4
enabled parameter (for Windows Media Player), HTML 372
endtime parameter (for Apple QuickTime), HTML 372
entity characters, APP 33
errors in HTML code, HTML 73–74
common validation errors, HTML 122
evaluating:
external style sheets, HTML 218

user surveys, HTML 27
Web sites, HTML 28–29, HTML 218, HTML 219
Excel documents: converting into HTML documents/files, HTML 14
Extensible Hypertext Markup Language. *See* XHTML
Extensible Markup Language (XML), **HTML 12**, APP 1
external style sheets, HTML 62, **HTML 63**, HTML 116, HTML 178–189, APP 21, APP 22
adding, HTML 178, HTML 180, HTML 356–357
analyzing/correcting errors, HTML 354
audio file linked to, HTML 369
classes in, HTML 178–179
code for, HTML 271, HTML 344–345; adding comments to, HTML 342
creating, HTML 179, HTML 180, HTML 181–182, HTML 270–272, HTML 342, HTML 348, HTML 353–354; for defining Web page style, HTML 345; for printing, HTML 348–351, HTML 355; using CSS, HTML 344–345
evaluating, HTML 218
linking from, HTML 342
linking to, HTML 179, HTML 180, HTML 186–187, HTML 199, HTML 203, HTML 342, HTML 346–348, HTML 349
page box model, **HTML 348**
@page rule, **HTML 348**
planning, HTML 270
pop-up function, HTML 331, HTML 358; adding, HTML 329, HTML 332–333, HTML 343–347, HTML 357–358; creating using CSS, HTML 344–345; developing, HTML 357–358
precedence, HTML 63, HTML 116, HTML 119, APP 22
style8.css, HTML 371, HTML 374, HTML 379
style statements in, HTML 183–185
testing, HTML 350
for use across a Web site, HTML 332
uses for, HTML 329, HTML 332
vs. embedded style sheets, HTML 178
vs. inline styles, HTML 64, HTML 178
Web page structure, HTML 343; box model, **HTML 343**; padding, **HTML 343**
extranets, **HTML 5**
best practice resources, HTML 5

F

favorites (bookmarks), HTML 37
fieldset control, **HTML 291–292**
for form grouping creation, **HTML 308–310**

- <fieldset> tag, HTML 291, APP 4
function of, HTML 293
- Fight Against World Hunger Web page project, HTML 34–35
general guidelines, HTML 36
See also throughout the rest of the chapter
- File Transfer Protocol. *See* FTP
- finding colors, HTML 183
finding tables on the Web, HTML 218
Flash video (.flv) video file format, HTML 369
floating images, HTML 132–134
float property, HTML 132, HTML 133
in form creation for Web pages, HTML 311
and values, APP 28
- folder structure, HTML 21, HTML 53, HTML 111
access paths, HTML 111–114
- fonts:
families, HTML 118; property and values, HTML 114, APP 26
properties: options, HTML 118; specifying, HTML 119–120, HTML 180, HTML 183; values, HTML 114, APP 26; and values, HTML 114
specifying, HTML 119–120, HTML 180, HTML 183; alternatives, HTML 118
font-size property and values, HTML 114, APP 26
font-style property and values, HTML 71, APP 26
- tag, HTML 114, APP 4
formatting images, HTML 64, HTML 130
formatting text, HTML 38, HTML 104, HTML 155
bolding text, HTML 52
with embedded style sheets, HTML 130–131
with external style sheets, HTML 178–189, HTML 183
headings, HTML 114–116. *See also* coloring headings
with inline styles, HTML 121, HTML 147–148
planning, HTML 45, HTML 102, HTML 104
tags for, HTML 115–116
form creation, for Web pages, 285–324
applications: course evaluation
form, HTML 317; fieldset controls/form organization, HTML 322–323; recording studio survey, HTML 321–322; restaurant questionnaire, HTML 318–319; school bookstore survey, HTML 320–321
- CGI scripts, HTML 294, HTML 296
check boxes, HTML 300–301
embedded style sheets, HTML 312–313
form groupings organization, HTML 308–310
get and post methods, **HTML 96**
- HTML tags used in, HTML 293;
attributes of, HTML 293–294
identification of form process, HTML 296
input controls, **HTML 289–292**, HTML 297–298
project planning guidelines, HTML 288, HTML 294, HTML 297
radio buttons, HTML 297, HTML 299–300, HTML 307
selection menu, HTML 301–302; advanced menus, HTML 302–304
submit/reset buttons, HTML 306–307
textarea box, HTML 305–306
text boxes, HTML 298–299; additional boxes, HTML 304–305
text messages, changing, HTML 297–298
- <form> tag, HTML 289, HTML 293
action attribute, **HTML 296**
method attribute, **HTML 296**
- frames, **HTML 394**
See also Web sites, converting to frames
frameset document type, **HTML 39**, HTML 394
- <frameset> tag, HTML 396, HTML 400–401, APP 4
- <frame> tag, HTML 396, HTML 400–401, APP 4
framing images, HTML 163
- FTP (File Transfer Protocol), APP 31
FTP programs (software), HTML 24, APP 31
uploading files with, APP 32
- fullscreen parameter (for Windows Media Player), HTML 372
functionality (of Web sites/pages): defining, HTML 16
testing, HTML 22–23, HTML 58
- G**
- get method (in form creation), **HTML 296**
- Getting Greener Web site project, HTML 90–91
planning guidelines, HTML 92
See also throughout the rest of the chapter
- GIF (Graphics Interchange Format) files, HTML 65, HTML 66
goals (for Web sites): identifying, HTML 15
graphical images. *See* images
graphics-editing programs, HTML 66
Graphics Interchange Format (GIF) files, HTML 65, HTML 66
- H**
- <h1> through <h6> tags, HTML 40, HTML 45, HTML 47, HTML 115, HTML 257, HTML 259, APP 5
<head>/<head> tags (HTML documents), HTML 39, HTML 40, HTML 42, HTML 43, HTML 117, HTML 179, HTML 256, APP 5
- header rows, HTML 165
headers, table. *See* table headers
heading cells (in tables), **HTML 163**, HTML 165
- headings, **HTML 38**
coloring. *See* coloring headings
consistent use of, HTML 45, HTML 46, HTML 105
coordinating with graphics, HTML 102, HTML 105
entering, HTML 47
formatting, HTML 114–116
formatting. *See also* coloring headings
main heading, HTML 45, HTML 46, HTML 47
organizational standard, HTML 16, HTML 28
sizing, HTML 45, HTML 46
in tables. *See* table headers
tags. *See* <h1> through <h6> tags (HTML documents)
height attribute (tag), HTML 64, HTML 66, APP 5
height property and values, APP 28
hierarchical Web site structure, HTML 18
deep Web sites, HTML 20–21
home page (start page), **HTML 4**, **HTML 57**, HTML 99
creating (steps), HTML 99–101, HTML 255–270; defining web page structure, HTML 255–256; inserting special characters, HTML 258–259, HTML 260, APP F; saving HTML file, HTML 256; starting notepad++, HTML 255; table creation, HTML 257–258
- Fight Against World Hunger Web page, HTML 34–35, HTML 36
- formatting text on, HTML 104
- Getting Greener Web site home page, HTML 90, HTML 91
- linking to, HTML 17–18, HTML 18–19, HTML 19–20, HTML 95–96, HTML 128–129
- name options, HTML 53
Sabatina's Pizza Web site home page, HTML 158, HTML 159
- horizontal menu bar. *See* menu bar, horizontal
- horizontal rules, **HTML 38**, HTML 67, HTML 69
adding, HTML 69, HTML 72
organizational standard, HTML 17, HTML 28
in tables, HTML 166
- hosts. *See* Web servers
- hotspots (in images), **HTML 38**, HTML 234, HTML 242, HTML 243
in image maps, HTML 234, HTML 237, HTML 251, HTML 254, HTML 265–266, HTML 268; circular, HTML 252–253, HTML 260;

- determining, HTML 283; dividing, HTML 239, HTML 246; linking to Web pages, HTML 284; mapping x- and y- coordinates, HTML 243, HTML 246–248; sketching the borders of, HTML 243–245, HTML 265; spillage prevention/testing, HTML 236
- hover styles of links, HTML 119, HTML 120
- href attribute (`<a>` tag), HTML 96, HTML 97, HTML 98, HTML 99, HTML 107, HTML 112, HTML 113, HTML 265, APP 2
- `
` tag, HTML 38, HTML 40, APP 5
- hspace attribute (`` tag), HTML 66
- .htm extension, HTML 53
- HTML (Hypertext Markup Language), HTML 2, **HTML 8–11**
- code. *See* HTML code
- elements. *See* HTML elements
- platform independence, HTML 9
- quick reference, APP 1–10
- styling limitation, HTML 11
- tags. *See* tags (HTML)
- versions, HTML 11
- HTML code (source code)
- adding: audio files, HTML 370, HTML 373, HTML 375; embedded style sheets, HTML 312–313; radio buttons, HTML 300; selection menu, HTML 302, HTML 304; submit/reset buttons, HTML 305–307; textarea boxes, HTML 305–306; text boxes, HTML 298–299, HTML 305; video clip, HTML 370, HTML 379, HTML 382, HTML 383
 - changing text messages, HTML 297
 - copying and pasting, HTML 143, HTML 190
 - defining Web page structure, HTML 256
 - embedded style sheets, HTML 335, HTML 336–337, HTML 339, HTML 340–341
 - errors in, HTML 73–74; common validation errors, HTML 122
 - external style sheets, HTML 271, HTML 342, HTML 344–345
 - files. *See* HTML documents/files
 - form groupings creation using fieldset controls, HTML 309–310
 - image maps, HTML 243, HTML 248–249, HTML 266
 - for image maps, HTML 243, HTML 248–249, HTML 266
 - indenting, HTML 175
 - puffed up code, HTML 14
 - starting the next line, HTML 43
 - for table creation, HTML 258
 - text link separator (!), HTML 175, HTML 180
 - validating, HTML 22, HTML 73–74, HTML 75–76, HTML 122, HTML 144, HTML 187, HTML 194, HTML 200, HTML 205, HTML 267,
- HTML 313, HTML 342, HTML 376–378, HTML 384
- viewing, HTML 75, HTML 77
- HTML documents/files
- coding standards/practices, HTML 10, HTML 11; organizational, HTML 16–17, HTML 27, HTML 28
 - converting application documents into, HTML 14
 - defining, HTML 39–40, HTML 42–44, HTML 100–101, HTML 171–172
 - elements. *See* HTML elements
 - modification dates, HTML 17
 - naming, HTML 53
 - opening, HTML 129, HTML 198, HTML 202
 - printing, HTML 79, HTML 123, HTML 144, HTML 188, HTML 200, HTML 205
 - saving, HTML 53–56, HTML 101–102, HTML 121, HTML 144, HTML 172–173, HTML 176–177, HTML 191, HTML 194, HTML 200, HTML 205, HTML 299, HTML 342
 - storing, HTML 53, HTML 55
 - tools for creating, HTML 13–14, HTML 24
- types supported by W3C, HTML 39
- uploading to Web servers, APP 31–32
- validating. *See* validating HTML code
- HTML elements, HTML 9, HTML 10
- nesting, HTML 12, APP 1
 - See also* attributes (of HTML tags); tags (HTML); Web page elements
- .HTML extension, HTML 53
- HTML Goodies Web site, HTML 38
- HTML object editors, **HTML 14**
- `<html>` tags, HTML 39, HTML 40, HTML 43, APP 5
- See also* tags (HTML)
- HTML text editors, **HTML 14**, HTML 24
- HTTP (Hypertext Transfer Protocol), **HTML 3**
- hyperlinks. *See* links
- hypertext links. *See* text links
- Hypertext Markup Language. *See* HTML
- Hypertext Transfer Protocol (HTTP), **HTML 3**
- I**
- identifying:
- content owners and authors, HTML 16
 - goals, HTML 15
 - users and user tasks, HTML 15–16
- `<iframe>` tag, APP 5
- image attributes, HTML 66
- image link border colors, HTML 94
- changing, HTML 95
- image links, HTML 38
- adding, HTML 143–144, HTML 155
- border colors, HTML 94; changing, HTML 95
- linking to the home page, HTML 95–96, HTML 128–129
- thumbnails, HTML 135
- image map button bar, **HTML 238**
- image maps, **HTML 38**
- application usage examples, HTML 239–242
 - `<area>` tag in, HTML 248, HTML 252, HTML 265
- client-side image map, **HTML 242–243**
- components (two) of, HTML 243
- creation of: coding the map, HTML 248–249, HTML 265–267; header/paragraphs of text, HTML 262; horizontal menu bar with text links, HTML 264; image insertion, HTML 261; image selection, HTML 243–244; mapping image coordinates, HTML 246–248; sketching hotspot borders, HTML 244–245; tags/tag attributes, HTML 265–266; testing links, HTML 273; title/unordered list, HTML 263
- e-mail links from, HTML 237, HTML 245
- external style sheet creation, HTML 270–272
- general project guidelines, HTML 236
- home page creation. *See under* home page
- hotspots in, HTML 234, HTML 237, HTML 251, HTML 254, HTML 265–266, HTML 268; circular, HTML 252–253, HTML 260; determining, HTML 283; dividing, HTML 239, HTML 246; linking to Web pages, HTML 284; mapping x- and y- coordinates, HTML 243, HTML 246–248; sketching the borders of, HTML 243–245, HTML 265; spillage prevention/testing, HTML 236
- HTML code creation, HTML 243, HTML 248–249, HTML 266
- image selection for, HTML 243
- `` tags in, HTML 249, HTML 260
- inserting an image in, HTML 260–265
- `<map>` tag in, HTML 248, HTML 261, HTML 265
- planning phase, HTML 243
- server-side image map, **HTML 242–243**
- software tools, HTML 253
- tags used for creation, HTML 261
- text links to URLs, HTML 237
- usemap attribute in, HTML 260, HTML 261
- use of, HTML 237–238
- See also* Bright Idea Web site; Paint program
- images (graphics), banners. *See* banner images, HTML 38, HTML 65, HTML 102
- adding (inserting), HTML 66–67, HTML 132–134, HTML 173

alternate text for, HTML 64. *See also* alt attribute
 attributes, HTML 66
 as bullets, HTML 105–106
 centering, HTML 173
 coordinating headings with, HTML 102, HTML 105
 floating, HTML 132–134
 formatting, HTML 64, HTML 130
 framing, HTML 163
 functions, HTML 66
 getting (sources), HTML 64, HTML 132
 hotspots in, HTML 38
 inline images, HTML 38
 linking to, HTML 143–144, HTML 155
 locating x- and y- coordinates, HTML 251–253
 optimizing, HTML 65
 overusing, HTML 66
 placing with tables, HTML 162, HTML 213–214
 resizing, HTML 135
 selection for image maps, HTML 243–244
 storing, HTML 53
 thumbnails as links to larger images, HTML 135
 turned on/off, on Web pages, HTML 238
 types, HTML 65–66
 use of, in image maps, HTML 236
 utilizing, HTML 102, HTML 170
 wrapping text around, HTML 132–134, HTML 136–137; stopping/clearing, HTML 138

Imaptool (image map software tool), HTML 254

 tag, HTML 64, HTML 132, APP 5
 attributes, HTML 66, HTML 261, APP 5
 in image maps, HTML 249, HTML 260

iMovie software, HTML 366

implementing and maintaining Web sites, HTML 24
 questions regarding, HTML 15
 important information: emphasizing, HTML 17, HTML 28

indenting (indentation), HTML 10
 code, HTML 175
 lists, HTML 51
 property and values, APP 29

individuals (persons): Web site uses, HTML 6–7, HTML 15
See also users

information. *See* content

informational Web pages: creating, HTML 85–86

initial tags: defining Web page structures with, HTML 39–40, HTML 42–44, HTML 100–101, HTML 171–172

inline images, HTML 38

inline style sheets, HTML 62–63, HTML 64, HTML 116, HTML 117,

HTML 326, HTML 332, HTML 333, APP 21, APP 22
 adding, HTML 356–357
 adding banner images with, HTML 102–103
 bolding text with, HTML 52
 coloring headings with, HTML 70, HTML 121
 containers for, HTML 121
 embedded style sheets vs., HTML 64, HTML 117, HTML 180, HTML 332, HTML 333
 external style sheets vs., HTML 64, HTML 178, HTML 328, HTML 332
 formatting text with, HTML 121, HTML 147–148
 for individual styles, HTML 332
 precedence, HTML 63, HTML 116–117, HTML 119, APP 22
 restyling bulleted/unordered lists with, HTML 68, HTML 71, HTML 105
 style statements in, HTML 63–64
 uses for, HTML 332
 Web site resource, HTML 63
 wrapping text around images with, HTML 133

input controls, HTML 289–292

<input> tag, HTML 290, HTML 293, HTML 301, HTML 307, APP 5–6

insertion point: moving, HTML 43

<ins> tag, APP 6

interactive Web pages, HTML 11

Internet, HTML 2–3
 accessibility standards, APP 14–20
 services available on, HTML 3; validation services, HTML 73
 Web site resource, HTML 3
See also World Wide Web (the Web)

Internet backbone, HTML 3

Internet Explorer:
 changing security level in, HTML 378
 plug-in location, HTML 367
 testing external style sheets with, HTML 350
 testing Web pages with, HTML 381, HTML 388, HTML 389
 turning on graphics in, HTML 283
 use of ActiveX controls, HTML 368, HTML 376
 validating Web pages with, HTML 267, HTML 313
 viewing Web pages with, HTML 267, HTML 273

Internet Service Providers (ISPs), HTML 3

Internet sites. *See* Web sites

intranets, HTML 5
 accessibility standards, APP 14–20
 best practice resources, HTML 5

ismap attribute, HTML 261

ISPs (Internet Service Providers), HTML 3

<i> tag, HTML 115, APP 5
 italic tag. *See* <i> tag

J

java applet, HTML 365, HTML 370
 Joint Photographic Experts Group (JPEG) files, HTML 65
 JPEG (Joint Photographic Experts Group) files, HTML 65

K

<kbd> tag, APP 6
 knowing: users, HTML 15–16
See also identifying

L

<label> tag, APP 6
 Learn It Online exercises, 274, HTML 25, HTML 80, HTML 146, HTML 207
<legend> tag, HTML 291, HTML 293, HTML 308, APP 6
 length property and values, APP 23
 letter-spacing property and values, APP 28
 linear Web site structure, HTML 17
 broad Web sites, HTML 19–20
 line breaks, HTML 10
 line-height property and values, APP 28
 line-through property value, APP 28
 link address symbol (#), HTML 98, HTML 139
 link colors:
 changing, HTML 95
 image link border colors, HTML 94; changing, HTML 95
 text link colors, HTML 93, HTML 94; changing, HTML 95
 linked style sheets. *See* external style sheets
 link element. *See* a element
 linking to e-mail addresses, HTML 98, HTML 99, HTML 109
 linking to external style sheets, HTML 179, HTML 180, HTML 186–187, HTML 199, HTML 203, HTML 346–347, HTML 349
 linking to images, HTML 143–144, HTML 155
 linking to link targets, HTML 141–143
 linking to text:
 on another Web site, HTML 96–97, HTML 111
 on the same site, HTML 95–96, HTML 107–108
 linking to Web pages:
 on another site, HTML 96–97, HTML 111
 on the same site, HTML 95–96, HTML 107–108, HTML 113, HTML 154–155; the home page, HTML 17–18, HTML 18–19, HTML 19–20, HTML 95–96, HTML 128–129; the same page, HTML 95, HTML 98, HTML 127–128, HTML 139–143

links (hyperlinks/hyperlink elements), **HTML 7–8, HTML 38, HTML 90, HTML 93–99, HTML 107, HTML 130**
 adding. *See* linking entries
 colors. *See* link colors
 creating Web pages with, HTML 93–99, HTML 150–151, HTML 152–153, HTML 156, HTML 214–215
 element types used as, HTML 93
 e-mail. *See* e-mail links
 image. *See* image links
 from image maps to URLs, HTML 237
 mouse pointer in movement over, HTML 93
 researching, HTML 93
 states, HTML 93, HTML 119
 styles, HTML 119, HTML 120, HTML 179, HTML 183
 targets. *See* link targets
 testing, HTML 125–126, HTML 178, HTML 196, HTML 273
 text. *See* text links
 types, HTML 90
See also hotspots (in images)
<link /> tag, HTML 179, HTML 186, APP 6
 link targets:
 creating/setting, HTML 139, HTML 140, HTML 148–149
 linking to, HTML 141–143
 list items: defining, HTML 50
 list properties:
 options, HTML 118
 and values, APP 27
 lists, **HTML 38**
 definition lists, **HTML 51–52, HTML 82–83**
 indenting, HTML 51
 ordered lists (numbered lists), HTML 46, **HTML 49–50**
 organizational standard, HTML 17, HTML 28
 properties and values, APP 27
 styles, HTML 48
 types, HTML 46, HTML 51
 unordered. *See* unordered lists (bulleted lists)
 list-style-image property and values, HTML 105, HTML 106, APP 27
 list styles, HTML 48
 list-style-type property and values, HTML 51, APP 27
** tag, HTML 40, HTML 50, APP 6**
 logical style tags, **HTML 115**
 logo banner, HTML 102, HTML 171
 logos as bullets, HTML 105–106
 logs for Web sites, HTML 24
 long file names: operating systems supporting, HTML 53
 loop parameter (for Apple QuickTime), **HTML 373**

M

Mac computers:
 multimedia software on, HTML 366–367
 user identification (vs. PCs), HTML 368
 MAGIX Movie Edit Pro software, HTML 366
<mailto> action, HTML 296
main heading, HTML 45, HTML 46, HTML 47
 logo banner as, HTML 102, HTML 171
 maintaining Web sites, HTML 24
 questions regarding, HTML 15
 Mapedit (image map software tool), HTML 254
<map> tag, APP 6
 in image maps, HTML 248, HTML 261, HTML 265
 margin properties, HTML 132–133
 defined, **HTML 343**
 options, HTML 118
 and values, APP 27
 mathematical and technical characters, APP 34
maxlength (maximum length), **HTML 289–290**
 measurement units for property values, APP 23
 media players and plug-ins, HTML 367–368
See also Apple; RealAudio (.ra, .ram) audio file format; Windows Media Player; individual players and plug-ins
 menu bar, horizontal: creating, HTML 174–175, HTML 180, HTML 185
 menu class, HTML 175, HTML 180
 menupipe class, HTML 174, HTML 175, HTML 180, HTML 185
 menuword class, HTML 174, HTML 175, HTML 180, HTML 185
 menuwordselect class, HTML 174, HTML 175, HTML 180, HTML 185
<meta /> tag, HTML 39, HTML 40, HTML 43, HTML 255, APP 6–7
 method attribute: of <form> tag, **HTML 296**
 Microsoft Silverlight plug-in, HTML 367
 MIDI (.mid, .rmi) audio file format, HTML 368
 modification dates (HTML files): including, HTML 17
 monitoring Web sites, HTML 24
 mouse pointer in movement over links, HTML 93
 move up folder symbol (..), HTML 113
 MP3 (.mp3) audio file format, HTML 368
 MP4 (.mp4) audio file format, HTML 368
 MPEG (.mpg, .mpeg, .mp3) video file format, HTML 369, HTML 373
 MP4 (.mp4) video file format, HTML 369
 multimedia technologies on Web sites. *See* Web pages, adding multimedia content
 multiple attribute, HTML 294

mute parameter:
 for Apple QuickTime, HTML 373
 for Windows Media Player, HTML 372

N

name attribute, **HTML 292**
<a> tag, HTML 107, HTML 139, HTML 265, HTML 294, APP 2
<input /> tag, HTML 293
<select> tag, HTML 294
<textarea> tag, HTML 294
 naming:
 classes, HTML 131, HTML 178
 HTML documents/files, HTML 53
 text links, HTML 93
 Web pages, HTML 21
 navigation bars: placing, HTML 174
 nesting HTML elements, HTML 12, APP 1
 networks, **HTML 3**
<noframes> tag, APP 7
 normal text, **HTML 38**
 Notepad++, HTML 13, HTML 34
 activating, HTML 61
 adding to the Start menu/taskbar, HTML 41
 code modification in, HTML 273
 enabling word wrap in, HTML 42
 external style sheet creation, HTML 272
 for form creation, HTML 295
 help information on, HTML 41
 for home page creation, HTML 255
 printing an HTML file, HTML 268
 quitting (closing), HTML 80, HTML 146, HTML 206, HTML 273, HTML 316, HTML 351
 saving file in, HTML 257
 starting, HTML 40–41, HTML 100, HTML 171, HTML 295, HTML 335, HTML 374, HTML 399
 numbered lists. *See* ordered lists
 numbers for ordered lists:
 changing, HTML 51
 options, APP 27

O

object editors: HTML editors, **HTML 14**
<object>/</object> tags:
 for audio clips, HTML 373
 for video clips, HTML 379, HTML 381, HTML 382
 objects, **HTML 369**
 object statement:
 code: for adding audio clips to Web pages, HTML 375; for adding video clips to Web pages, HTML 379
 height/width attributes, **HTML 370**
<object> tag, **HTML 369–371, APP 7**
 class IDs for ActiveX controls, HTML 370
 parameters, **HTML 372–374**

- observing and learning from Web sites, HTML 24
- obsolete elements (tags/attributes), HTML 115, APP 2
- <col> tag, HTML 49, APP 7
- opening HTML files, HTML 129, HTML 198, HTML 202
- operating systems: long file names support, HTML 53
- <optgroup> tag, HTML 304
- optimizing images, HTML 65
- option button. *See* radio button (radio)
- <option> tag, HTML 293, HTML 304, APP 7
- ordered lists (numbered lists), HTML 46, **HTML 49–50**
- See also* numbers for ordered lists
- organizational standards for HTML elements, HTML 16–17, HTML 27, HTML 28
- organizations:
- Web site uses, HTML 15
 - See also* users
- overlining property value, APP 28
- overusing images, HTML 66
-
- ## P
- padding properties:
- defined, **HTML 343**
 - and values, APP 27
 - See also* cellpadding
- page box, **HTML 348**
- page length: organizational standard, HTML 17, HTML 28
- @page rule, **HTML 348**
- Paint program (for finding x- and y- coordinates in image map), HTML 246–253
- Help feature, HTML 250
- instructions for starting, HTML 249–250
- locating x- and y- coordinates, HTML 251–253, HTML 274
- opening an image file, HTML 251
- starting, HTML 249
- window elements, HTML 250
- paragraphs, HTML 10
- adding/entering, HTML 47–48, HTML 104–105, HTML 176
 - bulletting, HTML 45
 - formatting beginning/middle/end paragraphs, HTML 131–132
 - organizational standard, HTML 17, HTML 28
 - separating, HTML 45
 - styles, HTML 179, HTML 183
- paragraph tag. *See* <p> tag
- <param> tag, APP 7
- password control, **HTML 290**
- creation of text box by, HTML 290
 - password text box (password), **HTML 289**
- paths: absolute vs. relative, HTML 111–114
- p element (paragraphs), HTML 131, HTML 183
- percentage property and values, APP 23
- period (): class name symbol, HTML 131, HTML 178, HTML 180
- periods (..): move up folder symbol, HTML 113
- personal Web pages: creating, HTML 86–87, HTML 87–88
- persons. *See* individuals
- physical style tags, **HTML 115**
- pipe symbol (!): text link separator, HTML 175, HTML 180
- placing text and images with tables, HTML 162, HTML 213–214
- planning tables, HTML 167–168, HTML 170, HTML 202
- planning Web pages:
- choosing a style level, HTML 117, HTML 180
 - external style sheets, HTML 180
 - formatting images, HTML 64, HTML 130
 - formatting text, HTML 45, HTML 102, HTML 104
 - getting images, HTML 64
 - links, HTML 107, HTML 130
 - project guidelines, HTML 160–161
 - storing HTML files, HTML 53, HTML 55
 - styles, HTML 191
 - tables, HTML 167–168, HTML 170, HTML 202
 - table spacing/cellspacing/cellpadding, HTML 198
 - testing, HTML 125
 - utilizing images (graphics), HTML 102, HTML 170
 - visual enhancements, HTML 67
- planning Web sites, HTML 15–16, HTML 161, HTML 236
- project guidelines, HTML 92
 - questions regarding, HTML 15, HTML 30–31
- platform independence of HTML, HTML 9
- playcount parameter (for Windows Media Player), HTML 372
- plug-ins (add-in/add-on), HTML 367
- See also* individual plug-ins
- PNG (Portable Network Graphics) files, HTML 65
- podcasts, HTML 364
- pointing hand mouse pointer, HTML 93
- pop-up function in external style sheets, HTML 331, HTML 358
- creating using CSS, HTML 344–345; applying text, HTML 358
 - developing for, HTML 332–333, HTML 357–358
- Portable Network Graphics (PNG) files, HTML 65
- post method (in form creation), **HTML 296**
- pound sign (#): link address symbol, HTML 98, HTML 139
- PowerPoint documents: converting into HTML documents/files, HTML 14
- precedence of styles/style sheets, HTML 63, HTML 116–117, HTML 119, APP 22
- predefined colors, HTML 68
- predefined entity characters, APP 33
- <pre> tag, HTML 115, APP 8
- printing HTML files, HTML 79, HTML 123, HTML 144, HTML 188, HTML 200, HTML 205, HTML 268–269, HTML 314
- printing Web pages, HTML 78, HTML 127, HTML 145, HTML 189, HTML 194, HTML 200, HTML 206, HTML 273, HTML 314, HTML 348, HTML 378, HTML 384
- properties and values (CSS), **HTML 63–64**, APP 23–29
- for backgrounds, APP 24; options, HTML 118
 - for borders, APP 25; options, HTML 118
 - for Cascading Style Sheets, HTML 271
 - for colors, APP 23; background color, APP 24; font color, HTML 114
 - for fonts, HTML 114, APP 26; options, HTML 118
 - for lists, APP 27; options, HTML 118
 - main property options, HTML 118
 - for margins, APP 27; options, HTML 118
 - for padding, APP 27
 - researching, HTML 87
 - for spacing, APP 27, APP 28–29
 - for text, APP 28–29; options, HTML 118
 - units of measure, APP 23
 - using, HTML 88
 - for visual parameters, APP 26
 - Web site resource, APP 23
- pseudo-classes in style statements, **HTML 119**, HTML 131
- <p> tag, HTML 40, APP 7
- inserting blank lines, HTML 45, HTML 47–48, HTML 134
 - in style sheets, HTML 329
- publishing (Web pages), **HTML 4**, HTML 24, APP 31–32
- puffed up HTML code, HTML 14
-
- ## Q
- <qp> tag, APP 8
- questionnaires (in usability testing), HTML 22, HTML 23
- questions regarding Web development phases, HTML 15
- planning phase questions, HTML 15, HTML 30–31
- QuickTime (.mov) video file format, HTML 367, HTML 369, HTML 370
- commonly used parameters, HTML 372–373

quitting Notepad++, HTML 80,
HTML 146, HTML 206, HTML 273
quitting Web browser, HTML 273
quotation mark (") symbol,
HTML 259

R

radio button (radio), HTML 289,
HTML 290
adding to forms on Web pages,
HTML 297, HTML 299–300,
HTML 307
radio control, **HTML 290**
rate parameter (for Windows Media Player), HTML 372
RealAudio (.ra, .ram) audio file format,
HTML 368, HTML 370
class IDs for ActiveX controls,
HTML 370
vs. Windows Media Player, HTML 391
RealVideo (.rm, .rv) file format,
HTML 369
refreshing the view in a browser,
HTML 73
registered trademark (®) symbol,
HTML 259
Rehabilitation Act. *See* Section 508
relative path, **HTML 113–114**
rel attribute (<a> tag), HTML 107, APP 2
reset button (reset), HTML 289,
HTML 292
adding to form on Web page,
HTML 306–307
reset control, **HTML 292**
resizing images, HTML 135
restyling unordered lists (bulleted lists),
HTML 67, HTML 68, HTML 71,
HTML 105
rev attribute (<a> tag), HTML 108, APP 2
rows (in tables), **HTML 163–164**
header rows, HTML 165
rules between, HTML 166
spanning, HTML 168, HTML 200–202,
HTML 204–205
as textarea control attribute, **HTML 291**
rowspan attribute (<th>/<td> tag),
HTML 200
row spanning, HTML 168,
HTML 200–202, HTML 204–205
Roxio Creator software, HTML 366
rules: in tables, HTML 166,
HTML 210–211
See also horizontal rules

S

Sabatina's Pizza Web site project,
HTML 158–160
general guidelines, HTML 160–161
See also throughout the rest of the chapter

<samp> tag, APP 8
Sapperzein Galleries Web site project,
HTML 326–327, HTML 329
adding a navigation menu with an embedded style sheet, HTML 334–342
adding pop-ups with an external style sheet, HTML 332–333,
HTML 343–347
adding styles sheets, HTML 332–332
creating an external style sheet for printing, HTML 347–351
saving HTML files, HTML 53–56,
HTML 101–102, HTML 121,
HTML 144, HTML 172–173,
HTML 176–177, HTML 191,
HTML 194, HTML 200, HTML 205
saving Web page structures, HTML 100
schedules in tables, HTML 216–217,
HTML 218
schools: Web site uses, HTML 6
See also users
<script> tag, APP 8
Section 508 (Rehabilitation Act), APP 13
guidelines on accessibility, HTML 16,
APP 13–16
select control, **HTML 291**
selected attribute, HTML 294
selection menu, HTML 291
adding to forms on Web page,
HTML 301–304
selector (in style statements), **HTML 63**,
HTML 117, HTML 118, HTML 130,
APP 21
pseudo-class applied to, HTML 119,
HTML 131
<select> tag, HTML 291, HTML 293,
HTML 303–304, APP 8
servers. *See* Web servers (hosts)
server-side image map, **HTML 242–243**,
HTML 260
services available on the Internet,
HTML 3
validation services, HTML 73
shape attribute, **HTML 265**
Shockwave Flash (.swf) video file format,
HTML 369, HTML 370
shorthand property, **HTML 337**
showaudiocontrols parameter (for Windows Media Player), HTML 372
showcontrols parameter (for Windows Media Player), HTML 372
showdisplay parameter (for Windows Media Player), HTML 372
showstatusbar parameter (for Windows Media Player), HTML 372
size, **HTML 289**
size attribute
for <input /> tag, HTML 293
for <select> tag, HTML 294
slideshows (multimedia presentation),
HTML 365
<small> tag, HTML 115, APP 8
software tools for image maps, HTML 253
source code, **HTML 75**
See also HTML code

spacing, HTML 10
properties and values, APP 27,
APP 28–29
in tables, HTML 198 (*See also* cellspacing)
See also blank lines
spacing properties: and values, APP 27,
APP 28–29
spanning rows (columns) in tables,
HTML 168, HTML 200–202,
HTML 204–205
/ tags (inline styles),
HTML 121
 tag, HTML 121, APP 8
special characters:
arrow characters, APP 34
entity characters, APP 33
inserting in home page, HTML 259,
HTML 260
mathematical and technical characters,
APP 34
src attribute (tag), HTML 66,
HTML 113, APP 5
src parameter (for Apple QuickTime),
HTML 373
standards. *See* accessibility standards;
coding standards/practices
starting:
browsers, HTML 57–58
the next line of code, HTML 43
Notepad++, HTML 40–41, HTML 100,
HTML 171
Start menu: adding Notepad++ to,
HTML 41
start page. *See* home page
starttime parameter (for Apple QuickTime), HTML 373
storing:
HTML files, HTML 53, HTML 55
images, HTML 53
stress testing Web sites, HTML 23
stretchtofit parameter (for Windows Media Player), HTML 372
strict document type, **HTML 39**
<strike> tag, APP 8
strike-through: property value, APP 28
 tag, HTML 115, APP 8
structure(s). *See* folder structure; Web page structures; Web site structures
style8.css external style sheet, HTML 371,
HTML 374, HTML 379
styles, **HTML 11**, **HTML 62**,
HTML 116, APP 21
for the body of an HTML document,
HTML 183
choosing a level, HTML 117,
HTML 180
defined, HTML 329
for links, HTML 119, HTML 120,
HTML 179, HTML 183
for paragraphs, HTML 179, HTML 183
planning, HTML 191

precedence, HTML 63,
 HTML 116–117, HTML 119, APP 22
 for tables, HTML 164, HTML 178–179,
 HTML 184–185
See also style sheets; style statements
 style sheets, **HTML 11**, **HTML 62–64**,
 HTML 116, APP 21–23
 choosing a level, HTML 117,
 HTML 180
 embedding into HTML file,
 HTML 352–353
 precedence, HTML 63, HTML 116–
 117, HTML 119, HTML 332, APP 22
 using, HTML 329
See also Cascading Style Sheets (CSS);
 embedded style sheets; external style
 sheets; inline style sheets
 style statements, **HTML 63–64**,
APP 21–22
 classes in. *See* classes (in style statements)
 in embedded style sheets, HTML 117,
 HTML 118–119
 in external style sheets, HTML 183–185
 in inline styles, HTML 63–64
 pseudo-classes in, HTML 119,
 HTML 131
<style></style> tags, APP 8
 embedded style sheets, HTML 117,
 HTML 161, HTML 310, HTML 366
 logical/physical, **HTML 115**
 style tags. *See* <style></style> tags
 subject attribute (<a> tag), HTML 110
 submit button (submit), HTML 289,
 HTML 292
 adding to form on Web page,
 HTML 306–307
 submit control, **HTML 292**
<sub> tag, HTML 115, APP 8
<sup> tag, HTML 115, APP 8
 surveys. *See* user surveys
 Systems Development Life Cycle for Web
 Development. *See* Web development
 life cycle

T

tabbing, HTML 10
 tabindex attribute, HTML 294
 table borders, **HTML 165**, HTML 170
 table captions, **HTML 165**, HTML 200,
 HTML 202
 adding, HTML 203
 styles, HTML 179, HTML 185
 table headers, **HTML 165**
 header rows, HTML 165
 heading cells, **HTML 163**, HTML 165
 styles, HTML 185
 tables, **HTML 161–178**, HTML 163
 borderless tables, HTML 162,
 HTML 171
 borders, **HTML 165**, HTML 170
 with borders, HTML 192; creating,
 HTML 193–194
 captions. *See* table captions
 cellpadding, **HTML 197**, HTML 198,
 HTML 199
 cellspacing, **HTML 197**, HTML 198,
 HTML 199
 creating, HTML 166–170; with borders,
 HTML 193–194; with multiple
 images, HTML 213–214; with rules,
 HTML 210–211; tutorials online,
 HTML 168
 creating a menu bar with,
 HTML 174–175
 determining if needed, HTML 166–167
 editing, HTML 208–209
 elements, HTML 163–164
 features, HTML 165–166
 finding on the Web, HTML 218
 framing images with, HTML 163
 headers. *See* table headers
 placing text and images with,
 HTML 162, HTML 213–214
 planning, HTML 167–168, HTML 170,
 HTML 202
 rules in, HTML 166, HTML 210–211
 schedules in, HTML 216–217,
 HTML 218
 spacing in, HTML 198. *See also*
 cellspacing
 spanning rows (columns), HTML 168,
 HTML 200–202, HTML 204–205
 styles, HTML 164, HTML 178–179,
 HTML 184–185
 tags, HTML 168–169; attributes,
 HTML 170
<table> tag, HTML 169, APP 8–9
 attributes, HTML 170
 in inserting special character, HTML 260
tags (HTML), **HTML 8–9**, HTML 118,
 APP 2–10
 case, HTML 39, HTML 44
 closing, HTML 12, HTML 40, APP 1
 deprecated tags, **HTML 11**, **HTML 39**,
 HTML 115, APP 2
 for formatting text, HTML 115–116
 for form creation, HTML 293; attributes,
 HTML 293–294
 for image maps, HTML 261
 initial. *See* initial tags
 logical style tags, **HTML 115**
 nesting, HTML 12, APP 1
 obsolete tags, HTML 115, APP 2
 physical style tags, **HTML 115**
 quick reference, APP 2–10
 researching, HTML 72
 sets. *See* containers
 for tables, HTML 168–169; attributes,
 HTML 170
 Web site resources, HTML 10,
 HTML 11, HTML 24, APP 2
 XHTML coding standards, HTML 12,
 APP 1
See also specific tags
 taskbar:
 adding Notepad++ to, HTML 41
 starting Paint from, HTML 249
<tbody> tag, APP 9
<td> tag, HTML 169, APP 9
 attributes, HTML 170, HTML 200
 in creating table data cells,
 HTML 257–258
 in inserting special character, HTML 260
 testing external style sheets, HTML 350
 testing links, HTML 125–126,
 HTML 178, HTML 196
 testing Web pages, HTML 125,
 HTML 145, HTML 189, HTML 313
 testing Web sites, HTML 22–23,
 HTML 58
 questions regarding, HTML 15
 text, HTML 38
 aligning. *See* aligning text
 alternate text for images, HTML 64. *See*
also alt attribute
 bolding, HTML 52
 formatting. *See* formatting text
 linking to: on another Web site,
 HTML 96–97, HTML 111; on
 the same site, HTML 95–96,
 HTML 107–108
 placing with tables, HTML 162,
 HTML 213–214
 properties and values, APP 28–29
 wrapping around images,
 HTML 132–134, HTML 136–137;
 stopping/clearing, HTML 138
See also headings; paragraphs
text-align property and values, APP 28
 textarea box (textarea), **HTML 289**
 adding to form on Web page,
 HTML 305–306
 textarea control: attributes, **HTML 291**
 textarea fields, HTML 291, HTML 292
<textarea> tag, HTML 293, APP 9
 text-based Web browsers, HTML 237
 text boxes (text), **HTML 289**
 adding to forms (on Web pages),
 HTML 298–299; additional text boxes,
 HTML 304–305
 password control creation of, HTML 290
 text control, **HTML 289**
 text-decoration property and values,
 HTML 95, APP 28
 text editors, **HTML 13**
 HTML editors, **HTML 14**, HTML 24
 WordPad, HTML 40
See also Notepad++
 text-indent property and values, APP 29
 text input control, **HTML 289**
 text link colors, HTML 93, HTML 94
 changing, HTML 95
 text links, HTML 38
 code separator (!), HTML 175,
 HTML 180
 colors for, HTML 93, HTML 94;
 changing, HTML 95
 creating a menu bar with, HTML 175
 image maps vs., HTML 234

linking to a Web page on another Web site, HTML 96–97, HTML 111
 linking to a Web page on the same site, HTML 95–96, HTML 107–108
 naming, HTML 93
 underlined text for, HTML 38, HTML 93
 text link separator (!), HTML 175, HTML 180
 text messages (on forms), HTML 297
 text properties:
 options, HTML 118
 and values, APP 28–29
 text-transform property and values, APP 28, APP 29
 text wrapping. *See* wrapping text around images
<tfoot> tag, APP 9–10
<thead> tag, APP 10
<th> tag, HTML 169, APP 10
 attributes, HTML 170, HTML 200
 thumbnail images as links to larger images, HTML 135
 titles, HTML 10, **HTML 37**
 organizational standard, HTML 16, HTML 28
<title> tag, HTML 39, HTML 40, HTML 44, APP 10
 tools for creating HTML documents/files, HTML 13–14, HTML 24
 trademark (™) symbol, HTML 259
 transitional document type, **HTML 39**, HTML 44, HTML 115
<tr> tag, HTML 169, APP 10
 attribute, HTML 170
 in creating table data cells, HTML 257–258
 in inserting special character, HTML 260
<tt> tag, HTML 115, APP 10
 tutorials online: for creating tables, HTML 168
 type attribute (<a> tag), HTML 108

U

uimode parameter (for Windows Media Player), HTML 372
 tag, HTML 40, HTML 49, APP 10
 underlined text (for text links), HTML 38, HTML 93
 disabling, HTML 95
 underlining property value, APP 28
 Unicode Transformation Format (UTF), HTML 39
 Uniform Resource Locators (URLs), **HTML 7**
 text links to, from image maps, HTML 237
 units of measure for property values, APP 23
 unordered lists (bulleted lists), HTML 45, HTML 46, **HTML 48–49**
 creating, HTML 50–51; with images as bullets, HTML 105–106

restyling, HTML 67, HTML 68, HTML 71, HTML 105
 updating Web sites, HTML 24
 uploading HTML files to Web servers, APP 31–32
 URL parameter (for Windows Media Player), HTML 372
 URLs (Uniform Resource Locators), **HTML 7**
 usability, **HTML 22**
 usability testing, **HTML 22–23**
 USB flash drives:
 printing CSS file from, HTML 272
 saving files to: external style sheet, HTML 345; frame files, HTML 398, HTML 402, HTML 403; HTML files, HTML 54–56, HTML 102, HTML 257, HTML 299, HTML 342; image map files, HTML 251; print style sheet, HTML 349
 testing links on Web pages from, HTML 268
 usemap attribute, HTML 260, HTML 261
 user interface design, HTML 18, HTML 60
 users:
 identifying (knowing), HTML 15–16
 observing, HTML 22
 surveying. *See* user surveys
 Web site uses, HTML 5–7, HTML 15
 user surveys, HTML 22
 creating, HTML 32
 evaluating, HTML 27
 user tasks: identifying, HTML 16
<u> tag, APP 10
 UTF (Unicode Transformation Format), HTML 39
 UTF-8 character encoding, HTML 39

V

validating HTML code, HTML 22, HTML 73–74, HTML 75–76, HTML 122, HTML 144, HTML 187, HTML 194, HTML 200, HTML 205, HTML 267, HTML 313, HTML 342, HTML 376–378, HTML 384
 validation errors, common, HTML 122
 validation services, HTML 73
 valign="top" attribute, HTML 258
 value attribute, **HTML 292**, HTML 294
 values. *See also* attribute values; properties and values
<var> tag, APP 10
 versions (HTML), HTML 11
 vertical-align property and values, APP 29
 vertical rules: in tables, HTML 166
 video file formats, HTML 369
 viewing HTML code, HTML 75, HTML 77, HTML 342
 viewing Web pages, HTML 59–60, HTML 73, HTML 124, HTML 145, HTML 177, HTML 189, HTML 194, HTML 200, HTML 206

HTML 313, HTML 342, HTML 376, HTML 384
 visual enhancements, HTML 68–73
 planning, HTML 67
 visual parameters: classification properties and values, APP 26
 volume parameter:
 for Apple QuickTime, HTML 373
 for Windows Media Player, HTML 372
 vspace attribute (tag), HTML 66

W

WAI (Web Accessibility Initiative):
 guidelines on accessibility, APP 13, APP 16, APP 17–20
 guidelines on server-side image maps, HTML 260
 priority levels (priorities), APP 16
 WAV (.wav) audio file format, HTML 368
 W3C (World Wide Web Consortium):
 HTML document types supported, HTML 39
 HTML standards, HTML 10, HTML 11
 HTML updates, APP 2
 image map information, HTML 240
 Internet and Web history, HTML 3
 WAI guidelines, APP 13, APP 16, APP 17–20
 Web page validation testing, HTML 236, HTML 267
 XHTML standards, HTML 236
 W3C markup validation service, HTML 73
 the Web. *See* World Wide Web (the Web)
 Web Accessibility Initiative. *See* WAI
 webbed Web site structure, HTML 18–19
 Web browsers (browsers), **HTML 7**
 newer types, HTML 12
 quitting, HTML 273, HTML 316
 refreshing the view, HTML 73
 starting, HTML 57–58
 text-based, HTML 237
 viewing frames in, HTML 398–399, HTML 405
 viewing HTML code in, HTML 75, HTML 77
 viewing Web pages in, HTML 59–60, HTML 73, HTML 124, HTML 145, HTML 177, HTML 189, HTML 194, HTML 200, HTML 206
 Web site compatibility testing, HTML 22–23, HTML 58
 Web site resource, HTML 58
 See also Internet Explorer
 Web development life cycle, **HTML 14**
 in form creation for Web pages, HTML 288
 phases and questions re, HTML 14, HTML 15; analysis phase, HTML 236; planning phase questions, HTML 15, HTML 30–31

- See also* analyzing Web sites; designing and developing Web sites; implementing and maintaining Web sites; planning Web sites; testing Web sites
- Web development projects:
 Fight Against World Hunger Web page project, HTML 34–35, HTML 36
 Getting Greener Web site project, HTML 90–91, HTML 92
 guidelines: general, HTML 36, HTML 160–161; planning, HTML 92
 History class Web site project, HTML 362–365, HTML 367, HTML 371, HTML 376–378, HTML 380–384
 Sabatina's Pizza Web site project, HTML 158–160, HTML 160–161, HTML 286–287, HTML 289, HTML 294–296, HTML 311, HTML 313
 Sapperzein Galleries Web site project, HTML 326, HTML 329, HTML 332–337, HTML 340–342, HTML 346, HTML 349, HTML 358
- Web page elements, HTML 10, HTML 37–38
 organizational standards, HTML 16–17, HTML 27, HTML 28
See also backgrounds; body; images; links; text; titles; *and also* HTML elements
- Web pages:
 from Access data, HTML 14
 adding image links to, HTML 143–144, HTML 155
 box model, **HTML 343**
 creating, HTML 84–87, HTML 87–88, HTML 155; a second Web page, HTML 189–194; guidelines. *See under* Web development projects; linked pages, HTML 214–215; with links, HTML 93–99, HTML 150–151, HTML 152–153, HTML 156, HTML 214–215; Web development projects. *See* Web development projects; Web site resource, HTML 38; with embedded style sheets. *See* embedded style sheets; with external style sheets. *See* external style sheets; with inline style sheets. *See* inline style sheets
 dividing into sections, HTML 65, HTML 102–103
 editing/improving, HTML 61–73, HTML 81–82, HTML 83, HTML 87, HTML 149–150, HTML 212, HTML 218, HTML 219 (*See also* images; links; styles; tables); the second Web page, HTML 127–129
 elements. *See* Web page elements
 form creation. *See* form creation, for Web pages
 home. *See* home page
 image map enhancement of, HTML 237–238
 images. *See* images
 with images turned off, HTML 238
 with images turned on, HTML 238
 interactive pages, HTML 11
 linked pages, HTML 214–215
 linking: to e-mail addresses, HTML 98, HTML 99, HTML 109; to external style sheets, HTML 179, HTML 180, HTML 186–187, HTML 199, HTML 203; external style sheets into, HTML 342; hotspots to, HTML 284; to link targets, HTML 141–143; to a page on another site, HTML 96–97, HTML 111
 linking to a page on the same site, HTML 95–96, HTML 107–108, HTML 113, HTML 154–155; the home page, HTML 17–18, HTML 18–19, HTML 19–20, HTML 95–96, HTML 128–129; the same page, HTML 95, HTML 98, HTML 127–128, HTML 139–143
 links. *See* links
 naming, HTML 21
 organizational standards, HTML 16–17, HTML 27, HTML 28
 padding, **HTML 343**
 planning. *See* planning Web pages
 printing, HTML 78, HTML 127, HTML 145, HTML 189, HTML 194, HTML 200, HTML 206, HTML 273, HTML 314
 publishing, **HTML 4**, HTML 24, APP 31–32
 saving. *See* saving HTML files
 saving and reusing structures, HTML 100
 storing, HTML 53, HTML 55
 structures. *See* Web page structures
 styles. *See* styles
 testing, HTML 125, HTML 145, HTML 189, HTML 268, HTML 273, HTML 329
 validating, HTML 22, HTML 144, HTML 267 (*See also* validating HTML code); ActiveX controls for, HTML 376–378
 viewing, HTML 59–60, HTML 73, HTML 124, HTML 145, HTML 177, HTML 189, HTML 194, HTML 200, HTML 206, HTML 267, HTML 273; ActiveX controls for, HTML 376–378
 Web pages, adding multimedia content, HTML 21, HTML 362–392
 audio file formats, HTML 368, HTML 369–378; Apple QuickTime parameters, HTML 372–373; background sounds, HTML 385–386; common formats, HTML 368; creating music, HTML 388; object statement HTML code, HTML 375; object tag, **HTML 369–371**; object tag parameters, HTML 372–373; planning phases, HTML 373; playing simultaneously with video files, HTML 389–390; validating/viewing using ActiveX controls, HTML 376–378; Windows Media Player parameters, HTML 372
 classid attribute: choosing, HTML 375; code for adding audio clip, HTML 370, HTML 373, HTML 375; code for adding video clip, HTML 370, HTML 379, HTML 382, HTML 383; defined, **HTML 370**
 combining audio and video files, HTML 386–387
 content choice, HTML 363–364; embedded vs. external media, HTML 366–367
 creating or finding multimedia files, HTML 366
 determining factors, HTML 364, HTML 368
 inserting multimedia, HTML 364
 media players and plug-ins, **HTML 367–368**; audio file formats, HTML 368; video file formats, HTML 369
 PC vs. Mac, HTML 368
 planning phase, HTML 363
 sources of information online, HTML 391
 testing pages, HTML 364
 using multimedia, **HTML 364–365**
 video file formats, HTML 369, HTML 379–384; on a full screen, HTML 389; HTML code, HTML 382–383; playing simultaneously with audio files, HTML 389–390; validating/viewing Web pages, HTML 384
See also class IDs for ActiveX controls
 Web page structures:
 with an external style sheet, HTML 343
 defining, HTML 39–40, HTML 42–44, HTML 100–101, HTML 171–172
 saving and reusing, HTML 100
 Web servers (hosts), **HTML 4**
 choosing, APP 31
 publishing Web pages to, **HTML 4**, HTML 24, APP 31–32
 uploading HTML files to, APP 31–32
 Web sites, **HTML 4**, **HTML 5**
 accessibility. *See* accessibility standards
 analyzing, HTML 16; questions regarding, HTML 15
 broad Web sites, HTML 19–20
 content. *See* content
 creating. *See* Web development projects
 deep Web sites, HTML 20–21
 designing and developing, HTML 16–21, HTML 29–30, HTML 32, HTML 161; questions regarding, HTML 15; using logs, HTML 24
 determining use for CSS, HTML 359
 evaluating, HTML 28–29, HTML 218, HTML 219
 folder structure. *See* folder structure
 functionality, HTML 16; testing, HTML 22–23, HTML 58

- implementing and maintaining, HTML 24; questions regarding, HTML 15
- logs, HTML 24
- monitoring, HTML 24
- multimedia technologies, HTML 21
- observing and learning from, HTML 24
- planning. *See* planning Web sites
- publishing, APP 31–32
- structures. *See* Web site structures
- testing, HTML 22–23, HTML 58; questions regarding, HTML 15
- types, HTML 4–5, HTML 30
- user interface design, HTML 18, HTML 60
- uses, HTML 5–7, HTML 15
- See also* Web pages and specific Web sites
- Web sites, converting to frames, HTML 393–407
- code: for copying for menu bar, HTML 402; for copy/pasting into other Web pages, HTML 404; for creating structure, HTML 397; for frame layouts, HTML 399–401; for pasting into home page/editing, HTML 403
- conversion strategy determination, HTML 401–407
- frame definition file, HTML 396–399
- frame tag attributes, HTML 396
- layout types, HTML 396, HTML 400
- planning phase, HTML 395–396
- project description, HTML 394–396
- validating/printing files, HTML 404
- viewing in Web browser, HTML 405
- viewing site with frames, HTML 398–399
- Web site structures, HTML 17–21, HTML 26
- improving, HTML 31
- Web site resources, HTML 21
- See also* Web page structures
- width attribute (tag), HTML 64, HTML 66, APP 5
- width property and values, APP 28
- window elements, HTML 37
- See also* backgrounds; body; titles
- windowlessvideo parameter (for Windows Media Player), HTML 372
- Windows Media Player, HTML 367, HTML 370
- ActiveX controls for, HTML 370, HTML 379–380; class IDs, HTML 370
- audio clip insertion code, HTML 370
- classid code for, HTML 370, HTML 373, HTML 375, HTML 379, HTML 382, HTML 383
- control panel views for, HTML 370
- Flash vs., HTML 391
- parameters used for, HTML 372
- video clip insertion code, HTML 381, HTML 382, HTML 390
- video file formats for, HTML 369
- vs, RealAudio, HTML 391
- Windows Media (.wmv) video file format, HTML 369
- controlling panel views for, HTML 370
- Windows Movie Maker (software), HTML 366
- Word documents: converting into HTML documents/files, HTML 14
- WordPad, HTML 40
- word-spacing property and values, APP 29
- word wrap, HTML 42
- enabling in Notepad++, HTML 42
- World Wide Web (the Web), HTML 3–4
- See also* Web browsers; Web development life cycle; Web development projects; Web pages; Web pages, adding multimedia content; Web servers; Web sites
- World Wide Web Consortium. *See* W3C
- wrapping text around images, HTML 132–134, HTML 136–137
- stopping/clearing, HTML 138
- WYSIWYG editors, HTML 14, HTML 24
- ## X
- x- and y- coordinates (for linkable areas), HTML 243
- mapping, in image maps, HTML 246–248
- Paint program for finding, HTML 246, HTML 249–253
- XHTML (Extensible Hypertext Markup Language), HTML 2, HTML 12, APP 1
- in code to define Web page structures, HTML 256
- coding standards/practices, HTML 12, HTML 39, HTML 236, HTML 288, APP 1
- external media file noncompliance, HTML 366–367
- making pages compliant, HTML 255
- testing for standards, HTML 236, HTML 288, HTML 329, HTML 364
- validating HTML tags, HTML 255
- W3C standards, HTML 236
- XML (Extensible Markup Language), HTML 12, APP 1
- in code to define Web page structure, HTML 256
- ## Y
- YouTube Web site, HTML 366, HTML 367, HTML 369, HTML 379