

Microsoft®

Start Here!™



Learn
HTML5

Faithe Wempen

Ready to learn HTML?

Start Here!™

Learn the fundamentals of HTML5—and begin building your first standards-based web pages from the ground up. If you have absolutely no previous experience, no problem—simply start here! This book introduces must-know concepts and getting-started techniques through easy-to-follow explanations, examples, and exercises. Then you'll put it all together by creating your first web pages.

Here's where you start learning HTML5

- Create a website using HTML5 in a simple text editor
- Use semantic tags to build intuitive and accessible page layouts
- Format content using tags and Cascading Style Sheets (CSS)
- Add graphics to your site to increase visual appeal
- Build navigation bars to help users browse your site
- Develop forms to collect user information
- Embed audio and video in your web pages

GET CODE SAMPLES ONLINE

Ready to download at
<http://go.microsoft.com/fwlink/?Linkid=263540>

For system requirements, see the *Introduction*.

Start Here! Learn HTML5

Skill Level: Beginner

Prerequisites: None

DEVELOPER ROADMAP

Start Here!

- Beginner-level instruction
- Easy-to-follow explanations and examples
- Code samples to use in your first projects



Step by Step

- For experienced developers learning a new topic
- Focus on fundamental techniques and tools
- Hands-on tutorial with practice files plus eBook



Developer Reference

- Professional developers; intermediate to advanced
- Expertly covers essential topics and techniques
- Features extensive, adaptable code examples



Focused Topics

- For programmers who develop complex or advanced solutions
- Specialized topics; narrow focus; deep coverage
- Features extensive, adaptable code examples



About the Author

Faithe Wempen, M.A., is a Microsoft Office Master Instructor. She has written more than 120 books on PC hardware and software, including *HTML5 Step by Step*, which is designed for experienced programmers new to the language.

microsoft.com/mspress

Microsoft®

ISBN: 978-0-7356-6982-6



9 780735 669826

U.S.A. \$34.99

Canada \$36.99

[Recommended]

Programming/HTML

**Start
Here!**[™]

Learn
HTML5

Faithe Wempen

Copyright © 2013 by Faithe Wempen

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

ISBN: 978-0-7356-6982-6

1 2 3 4 5 6 7 8 9 LSI 7 6 5 4 3 2

Printed and bound in the United States of America.

Microsoft Press books are available through booksellers and distributors worldwide. If you need support related to this book, email Microsoft Press Book Support at mspinput@microsoft.com. Please tell us what you think of this book at <http://www.microsoft.com/learning/booksurvey>.

Microsoft and the trademarks listed at <http://www.microsoft.com/about/legal/en/us/IntellectualProperty/Trademarks/EN-US.aspx> are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

The example companies, organizations, products, domain names, email addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

This book expresses the author's views and opinions. The information contained in this book is provided without any express, statutory, or implied warranties. Neither the authors, Microsoft Corporation, nor its resellers, or distributors will be held liable for any damages caused or alleged to be caused either directly or indirectly by this book.

Acquisitions and Developmental Editor: Russell Jones

Production Editor: Melanie Yarbrough

Editorial Production: Box Twelve Communications

Technical Reviewer: John Mueller

Indexer: WordCo Indexing Services

Cover Design: Jake Rae

Cover Composition: Zyg Group

Illustrator: Rebecca Demarest

To Margaret

Contents at a Glance

Introduction xv

PART I **GETTING STARTED WITH HTML**

CHAPTER 1	HTML Basics: The Least You Need to Know	3
CHAPTER 2	Setting Up the Document Structure	19
CHAPTER 3	Formatting Text with Tags	29
CHAPTER 4	Using Lists and Backgrounds	49
CHAPTER 5	Creating Hyperlinks and Anchors	71

PART II **STYLE SHEETS AND GRAPHICS**

CHAPTER 6	Introduction to Style Sheets	87
CHAPTER 7	Formatting Text with CSS	105
CHAPTER 8	Formatting Paragraphs with CSS	125
CHAPTER 9	Inserting Graphics	141

PART III **PAGE LAYOUT AND NAVIGATION**

CHAPTER 10	Creating Navigational Aids	163
CHAPTER 11	Creating Division-Based Layouts	179
CHAPTER 12	Creating Tables	199
CHAPTER 13	Formatting Tables	223
CHAPTER 14	Creating User Forms	241
CHAPTER 15	Incorporating Sound and Video	261
CHAPTER 16	HTML and Microsoft Expression Web	275

PART IV **APPENDICES**

APPENDIX A	Designing for Usability	307
APPENDIX B	Designing for Accessibility	313
APPENDIX C	Quick Reference	321

Glossary 323

Index 327

Contents

Introductionxv

PART I GETTING STARTED WITH HTML

Chapter 1	HTML Basics: The Least You Need to Know	3
	Understanding HTML Tags	3
	Opening a Webpage in Notepad	6
	Other Ways of Opening Webpages in Notepad	9
	Adding a Location to the Favorites List	11
	Previewing HTML Files in a Web Browser	13
	Making, Saving, and Viewing Changes	16
	Key Points	17
Chapter 2	Setting Up the Document Structure	19
	Specifying the Document Type	20
	Creating the HTML, Head, and Body Sections	21
	Creating Paragraphs and Line Breaks	23
	Specifying a Page Title and Metatags	26
	Key Points	28
Chapter 3	Formatting Text with Tags	29
	Creating Headings	30
	Applying Bold and Italic Formatting	33
	Applying Superscript and Subscript Formatting	35

What do you think of this book? We want to hear from you!

Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

microsoft.com/learning/booksurvey

Using Monospace and Preformatted Text	36
Formatting a Block Quotation	42
Configuring View Settings in Internet Explorer	44
Key Points.	48
Chapter 4 Using Lists and Backgrounds	49
Creating Bulleted and Numbered Lists.	50
Nesting Lists	51
Changing the Bullet or Number Character	52
Specifying the Start of a Numbered List	53
Creating Definition Lists.	56
Inserting Special Characters	58
Inserting Horizontal Lines	60
Choosing Background Colors and Foreground Colors.	64
Understanding How Colors are Specified on Webpages	64
Applying a Background Color	65
Applying a Foreground Color	66
Specifying a Background Image File.	67
Key Points.	70
Chapter 5 Creating Hyperlinks and Anchors	71
Hyperlinking to a Webpage	72
Using Partial Paths and Filenames	72
Using Relative and Absolute Paths	73
Setting a Target Window	74
Hyperlinking to an Email Address	76
Creating and Hyperlinking to Anchors	79
Hyperlinking to Other Content	82
Key Points.	84

Chapter 6	Introduction to Style Sheets	87
	Understanding Styles	88
	Constructing Style Rules	89
	Creating Styles for Nested Tags	93
	Creating Classes and IDs for Applying Styles	95
	Applying Styles to Hyperlinks with Pseudo-Classes	98
	Creating and Linking to External Style Sheets	100
	Key Points	102
Chapter 7	Formatting Text with CSS	105
	Specifying a Font Family	106
	Specifying a Font Size and Color	109
	Applying Bold and Italics	114
	Applying Strikethrough and Underlining	116
	Creating Inline Spans	118
	Adjusting Spacing between Letters	120
	Key Points	124
Chapter 8	Formatting Paragraphs with CSS	125
	Indenting Paragraphs	126
	Applying a Border to a Paragraph	129
	Specifying a Border Style	129
	Setting Border Padding	130
	Specifying Border Width and Color	131
	Formatting Border Sides Individually	131
	Setting All Border Attributes at Once	132
	Specifying the Horizontal Alignment of a Paragraph	134
	Specifying Vertical Space within a Paragraph	137
	Key Points	139

Chapter 9	Inserting Graphics	141
	Understanding Graphic Size and Resolution	142
	Inserting a Graphic	143
	Clearing a Graphic	146
	Controlling Image Size and Padding	148
	Using a Graphic as a Hyperlink	151
	Creating Thumbnails	153
	Including Alternate Text for Graphics	155
	Adding Figure Captions	156
	Key Points	159

PART III **PAGE LAYOUT AND NAVIGATION**

Chapter 10	Creating Navigational Aids	163
	Planning Your Site's Organization	164
	Creating a Text-Based Navigation Bar	166
	Creating a Graphical Navigation Bar	169
	Redirecting to Another URL	173
	Creating Custom Error Pages	175
	Key Points	178
Chapter 11	Creating Division-Based Layouts	179
	Understanding HTML5 Semantic Tags	180
	Beginning to Think in Divisions	181
	Creating Divisions	182
	Creating an HTML5 Semantic Layout	184
	Positioning Divisions	187
	Floating a Division to the Right or Left	187
	Positioning a Division on the Page	188
	Formatting Divisions	195
	Key Points	198

Chapter 12 Creating Tables	199
Creating a Simple Table	200
Specifying the Size of a Table	204
Specifying the Width of a Column	209
Spanning (Merging) Table Cells	213
Using Tables for Page Layout	216
Key Points	221
Chapter 13 Formatting Tables	223
Applying Table Borders	224
Applying Borders by Using Attributes	224
Applying Borders by Using Styles	226
Applying Background and Foreground Fills	232
Changing Cell Padding, Spacing, and Alignment	235
Setting Cell Padding	237
Setting Cell Spacing	237
Setting Horizontal and Vertical Alignment	238
Key Points	240
Chapter 14 Creating User Forms	241
Creating a Basic Form	242
Creating a Text Box	243
Special Field Types for Email and Web Addresses	244
Creating a Text Area	244
Creating a Submit or Clear Button	244
Adding Default or Placeholder Text	245
Creating Check Boxes and Option Buttons	249
Creating Menu Lists	252
Additional Input Types in HTML5	256
Understanding CGI and Other Advanced Tools	257
Key Points	258

Chapter 15 Incorporating Sound and Video **261**

Understanding Audio and Video in HTML5 262

HTML Multimedia Basics 262

 Multimedia Formats and Containers 263

 Codecs: Decoding the Video and Audio 263

 Which Format to Choose? 264

 Encoding Video 264

Embedding Video Clips 266

 Introducing the `<video>` Tag 266

 The `<embed>` Tag: Your Fallback Plan 268

 Place a Video Clip on a Webpage 269

Incorporating Audio on a Webpage 271

 Playing Audio with the `<audio>` Tag 271

 Playing Audio in Older Browsers 272

Key Points 274

Chapter 16 HTML and Microsoft Expression Web **275**

Exploring the Expression Web Interface 276

Creating Websites and Webpages 281

 Create a Page by Using a CSS Template 285

Insert Graphics 288

 Import an Images Folder 288

 Place Images on a Page 289

 Add a Background Image to a Division 290

Formatting Text 292

Formatting a Division 297

Inserting Hyperlinks 299

Key Points 302

PART IV APPENDICES

Appendix A Designing for Usability 307

- Understanding Usability 307
- Planning for Usability 308
- Sketching the Site Organization 309
- Designing a Consistent Page Template 309
- Designing the Content of Individual Pages 310
- Performing Usability Testing 311

Appendix B Designing for Accessibility 313

- Guideline 1: Provide Equivalent Alternatives to Auditory and Visual Content 314
- Guideline 2: Don't Rely on Color Alone 314
- Guideline 3: Use Markup and Style Sheets, and Do So Properly 315
- Guideline 4: Clarify Natural Language Usage 316
- Guideline 5: Create Tables that Transform Gracefully 316
- Guideline 6: Ensure Pages that Feature New Technologies Transform Gracefully 317
- Guideline 7: Ensure User Control of Time-Sensitive Content Changes .. 317
- Guideline 8: Ensure Direct Accessibility of Embedded User Interfaces .. 318
- Guideline 9: Design for Device Independence 318
- Guideline 10: Use Interim Solutions 318
- Guideline 11: Use W3C Technologies and Guidelines 319
- Guideline 12: Provide Context and Orientation Information 319
- Guideline 13: Provide Clear Navigation Mechanisms 320
- Guideline 14: Ensure that Documents are Clear and Simple 320

What do you think of this book? We want to hear from you!
Microsoft is interested in hearing your feedback so we can continually improve our books and learning resources for you. To participate in a brief online survey, please visit:

microsoft.com/learning/booksurvey

Appendix C Quick Reference	321
Tags Added in HTML 5.....	321
Tags Removed in HTML 5.....	322
<i>Glossary</i>	323
<i>Index</i>	327

Introduction

Hypertext Markup Language (*HTML*) is the basic programming language of the World Wide Web. It's the common thread that ties together virtually every website, from large-scale corporate sites like Microsoft's to single-page classroom projects at the local grade school.

In simple terms, a **webpage (or HTML document)** is a plain text file that has been encoded using HTML so that it appears nicely formatted in a web browser. Here's what HTML means, word-by-word:

- **Hypertext** Text that you click to jump from document to document. This is a reference to the ability of webpages to link to one another.
- **Markup** Tags that apply layout and formatting conventions to plain text. Literally, the plain text is "marked up" with the tags.
- **Language** A reference to the fact that HTML is considered a programming language.

Don't let the phrase "programming language" intimidate you. Creating a webpage with HTML is much simpler than writing a computer program in a language like Microsoft Visual Basic or C++. You can create a simple webpage in just a few minutes.



Tip When people think of computer programming, they usually think of writing a *compiled* program. A compiled programming language runs the human-readable programming code through a utility that converts it to an executable file (usually with an *.exe* or *.com* extension), which is then distributed to users. In contrast, HTML is an *interpreted* programming language. That means the program is distributed in human-readable format to users, and the program in which it is opened takes care of running it. The HTML code for webpages resides in files. Each time your web browser opens a webpage, it processes the HTML code within the file.

This book teaches beginner-level HTML in a rather fundamentalist way: by creating plain text files in Notepad. There are so many good website creation programs on the market nowadays that you may be wondering why this book takes this approach.

Simply put, it's because doing your own coding is the best way to learn HTML. In this book you'll build a website from the ground up, writing every line of code yourself. It's slower and not as much fun as a fancy graphical program, but it's great training.

The last chapter of this book shows how to use Microsoft Expression Web to create web content, and you may eventually choose to move to a program like that. However, you will be a much better web designer—and understand what is going on in design programs much better—if you tough it out with Notepad in the beginning.

Who Should Read This Book

This book is designed for non-programmers who are brand-new to HTML. It doesn't assume any previous web design or programming knowledge of any kind, so it's perfect for home hobbyists and self-educators, as well as high school and college classrooms.

Assumptions

This book expects that you have basic computer literacy skills and you can use the operating system on your PC. The detailed procedures for this book were written with Windows developers in mind, but the general information applies to building webpages on any operating system. You should know how to manage files, run programs, and access the Internet.

I'm also assuming that if you're reading this book, you are actually interested in learning HTML and not just knocking out a quick webpage only to never think about HTML again. This book provides you with a solid foundation in HTML that you'll be able to draw from for years to come.

Who Should Not Read This Book

This book is not for experienced programmers who already have significant HTML knowledge and are just interested in updating their skills for HTML5. This book doesn't cover many of HTML5's advanced features that experienced web developers are likely to be interested in learning about; instead, it provides a solid foundation for the beginner.

Organization of This Book

This book is divided into three sections, each of which focuses on a different aspect of webpage development. Part I, "Getting Started with HTML," explains how HTML works and helps you set up the structure of a page and place basic text elements on it such as

headings, paragraphs, and lists. Part II, “Style Sheets and Graphics,” explains how to use cascading style sheets (CSS) to apply formatting to a page or a group of pages and how to include graphic elements on a page. Part III, “Page Layout and Navigation,” explains how to create intuitive page layouts and navigational aids as well as how to add special elements such as forms and multimedia.

Conventions and Features in This Book

This book presents information using conventions designed to make the information readable and easy to follow.

- Each exercise consists of a series of tasks, presented as numbered steps (1, 2, and so on) listing each action you must take to complete the exercise.
- Boxed elements with labels such as “Note” provide additional information or alternative methods for completing a step successfully.
- Text that you type displays in **boldface**. For example, if you are instructed to type some text to an existing block of code, the text you are to type is displayed in bold.
- Glossary terms marked in the text are explained in the glossary at the end of the book.

System Requirements

You will need the following hardware and software to complete the practice exercises in this book:

- A computer running the operating system of your choice. The examples in this book use Windows 8, but you can use any Windows, Mac, Linux, or UNIX operating system.
- A text-editing program. The examples in this book use Notepad, which comes with all Windows versions.
- One or more web browsers. The examples in this book use Internet Explorer 10. You will probably want at least one other browser in which to check your work, because different browsers might display content in different ways.
- An Internet connection to download the practice files.

Code Samples

Each chapter includes exercises that let you interactively try out new material learned in the main text. All files you will need to complete these exercises can be downloaded from the following page:

<http://www.microsoftpressstore.com/title/9780735669826>

Follow the instructions to download the *9780735669826_files.zip* file.

Installing the Code Samples

Follow these steps to install the code samples on your computer so that you can use them with the exercises in this book.

1. Unzip the *9780735669826_files.zip* file that you downloaded from the book's website (name a specific directory along with directions to create it, if necessary).
2. If prompted, review the displayed end user license agreement. If you accept the terms, select the accept option and then click Next.



Note If the license agreement doesn't display, you can access it from the same webpage from which you downloaded the *9780735669826_files.zip* file.

Using the Code Samples

The folder created by the Setup.exe program contains subfolders for each chapter. Within a chapter's folder are additional subfolders that hold the files for each exercise. Each time you begin an exercise, make sure that you are using the files provided specifically for that exercise.

Each chapter folder contains a *_Solutions* subfolder, which contains samples of the finished files for that chapter's exercises. You can use these to check your work.

Acknowledgments

Thank you to my editorial team at O'Reilly for a job well done, including Russell Jones (Acquisitions/Development), John Mueller (Technical Editor), Jeff Riley (Copy Editor), and Melanie Yarbrough (Production Editor). It is a pleasure to work with a professional team of editors who make the complex process of producing a book go as smoothly as possible.

Errata & Book Support

We've made every effort to ensure the accuracy of this book and its companion content. Any errors that have been reported since this book was published are listed on our Microsoft Press site:

<http://www.microsoftpressstore.com/title/9780735669826>

If you find an error that is not already listed, you can report it to us through the same page.

If you need additional support, email Microsoft Press Book Support at mspinput@microsoft.com.

Please note that product support for Microsoft software is not offered through the addresses above.

We Want to Hear from You

At Microsoft Press, your satisfaction is our top priority, and your feedback our most valuable asset. Please tell us what you think of this book at:

<http://www.microsoft.com/learning/booksurvey>

The survey is short, and we read every one of your comments and ideas. Thanks in advance for your input!

Stay in Touch

Let's keep the conversation going! We're on Twitter: <http://twitter.com/MicrosoftPress>

PART I

Getting Started with HTML

CHAPTER 1	HTML Basics: The Least You Need to Know	3
CHAPTER 2	Setting Up the Document Structure.	19
CHAPTER 3	Formatting Text with Tags	29
CHAPTER 4	Using Lists and Backgrounds.	49
CHAPTER 5	Creating Hyperlinks and Anchors.	71



HTML Basics: The Least You Need to Know

In this chapter, you will:

- Learn how HTML tags work
- Open a webpage in Notepad
- Preview a webpage in Internet Explorer
- Make, save, and view changes
- Learn how to publish your pages to a server

As you work through this book's exercises, you'll learn HTML by creating and editing text files in Notepad (or any plain text editor), and then viewing them in a web browser to check your work. This chapter teaches the important basic skills you need in order to work in these programs.



Important Before you can use the practice files provided for this chapter, you need to download and install them from the book's companion content location. For more information, see "Code Samples" at the beginning of this book.

Understanding HTML Tags

An HTML file (also known as a webpage file or document) is a plain-text file. That's why you can use a plain-text editor like Notepad to edit an HTML file. Within that file is literal text (that is, text that will literally appear on the screen when you display the page in a browser) plus tags that tell the browser how to format and arrange the text.

HTML tags are codes inside angle brackets, like this: `<p>`. There are dozens of tags defined in the HTML5 standard. You can use these tags to specify where formatting should be applied, how the layout should appear, what pictures should be placed in certain locations, and more.

For example, suppose you wanted a certain word to appear in bold, like this:

Save up to **50%** on clearance items.

In HTML, there's no Bold button to click (like there is in a word-processing program). Therefore, you have to "tag" the word or characters that you want to be bold. The tag to turn on bold is ``, and the code to turn bold off is ``. Your HTML code would look something like this:

Save up to `50%` on clearance items.

That's an example of a *two-sided tag*, which encloses text between an opening tag and a closing tag, in this case `` and ``. Note the forward slash (/) in the closing tag (``). That slash differentiates an opening tag from a closing tag. With a two-sided tag, there is always a corresponding closing tag for every opening tag.

To understand how this system of tagging came about, you need to know that back in the early days of the Internet, nearly everyone connected to it by using dial-up modems at speeds ranging from 2400 bps to 28.8 Kbps. That's really slow. Text files transfer much faster than binary files, so for any type of information-sharing system to be popular, it had to be text-based. Otherwise, people would doze off while waiting for a page to load.

People designing webpages also wanted their pages to be attractive. They couldn't just format pages in a word processor, though, because every word processor handled formatting differently and it was impossible to know which word processor a visitor might be using. Word-processing files are also much larger than plain-text files.

The web's creators developed an elegant solution. Instead of sending the formatted pages over the Internet, they created an application—a web browser—that could interpret plain-text code (HTML tags) as formatting instructions. The text could be sent quickly and efficiently in plain-text format, and then be processed and displayed attractively and graphically on the local PC.

HTML worked great all by itself for all kinds of text formatting, but some web designers wanted to include graphics on their pages. To accommodate this, the `` tag was created, which designers use to refer to a graphic stored on a server. When the web browser gets to that tag, it requests that the image file be downloaded from the server and displayed on the page. (You'll learn how to insert images in Chapter 9, "Inserting Graphics.")

The `` tag is different in several ways from the `` tag. It is one-sided, meaning it does not have a closing tag, and it takes attributes. An *attribute* is text within the tag that contains information about how the tag should behave. For example, for an `` tag, you have to specify a source (abbreviated *src*). Here's an example:

```

```

This `` tag uses the `src=` attribute and specifies that the file *tree.gif* be displayed.

Many tags accept attributes, either optional or required. You'll see many examples throughout the exercises in this book. Attributes are usually written in pairs in the form *name=value*, where the name of the attribute is on the left side of the equals sign, and the value on the right.

With HTML, you can also create hyperlinks from one page to another. When a visitor clicks a hyperlink, the web browser loads the referenced page or jumps to a marked section (a "bookmark") within the same page. You will learn to create hyperlinks in Chapter 5, "Creating Hyperlinks and Anchors."

The tag for a hyperlink is `<a>`, a two-sided tag, but most people wouldn't recognize it without the attribute that specifies the file or location to which to jump. For example, to create a hyperlink with the words Click Here that jumps to the *index.htm* file when clicked, the coding would look like this:

```
<a href="index.htm">Click Here</a>
```

There's a lot more to HTML, but that's basically how it works. You mark up plain text with tags that indicate where you want to apply elements such as formatting, hyperlinks, and graphics. Then a web browser interprets those tags and displays the page in its formatted state, a process called *rendering*. The trick, of course, is to know which tags to use, where they're appropriate, and what attributes they need. And that's the subject of this book. Starting in Chapter 2, you'll begin building a website from scratch, learning about each individual tag as you go. By the time you finish this book, you will have created a multi-page website using nothing but Notepad and the tags that you have personally learned about and typed. Talk about boot camp! There are professional web designers out there today who have never done this and who don't have the fundamental HTML coding skills that you'll have when you finish this book.

What is XHTML?

There is a language related to HTML called *Extensible Markup Language (XML)* that programmers use to create their own tags. It's widely used for web databases, for example, because it can define tags for each data field. Because XML can be so completely customized, programmers can create almost any other markup language within it just by re-creating all the officially accepted tags of that language.

XHTML, then, is HTML written within the larger language of XML. Because it is virtually identical to HTML in its functionality, the basic set of tags is the same and you can learn both HTML and XHTML at the same time. You can also use XHTML to create new tags and extensions, which is a valuable feature for advanced web developers.

There's just one thing about XHTML to watch out for: it's not tolerant of mistakes. For example, in HTML, technically you are supposed to begin each paragraph with `<p>` and end each paragraph with `</p>`. But in HTML you can leave out the closing `</p>` tag if you want (or if you forget it). That won't work in XHTML. There are lots of little ways that XHTML is picky like that. I'll point out some of them along the way in this book.

At one point, it was thought that XHTML would eventually replace HTML 4, but due to interoperability problems, that has not happened; instead HTML5 is poised to succeed HTML 4. This book doesn't explicitly cover XHTML, but most of what you will learn can be applied to XHTML coding.

Opening a Webpage in Notepad

The rest of this book assumes that you will be working in Notepad or some other text editor, so you'll need a basic understanding of whatever text editor you select. Notepad is included with all versions of Windows. It's a simple text editor that saves only in plain-text format. That's ideal for HTML editing because you don't need to worry about any extra word-processing formatting being included in the file.



Note You are welcome to use a different text editor application to complete the exercises in this book. Notepad is just a suggestion.

When saving or opening files in Notepad, the default file extension is `.txt`. The Save dialog boxes and Open dialog boxes are set by default to filter file listings so only those files with `.txt` extensions appear. That means each time you browse for a file, you need to change the file type to All Files so you can browse for webpages (which have `.htm` or `.html` extensions).

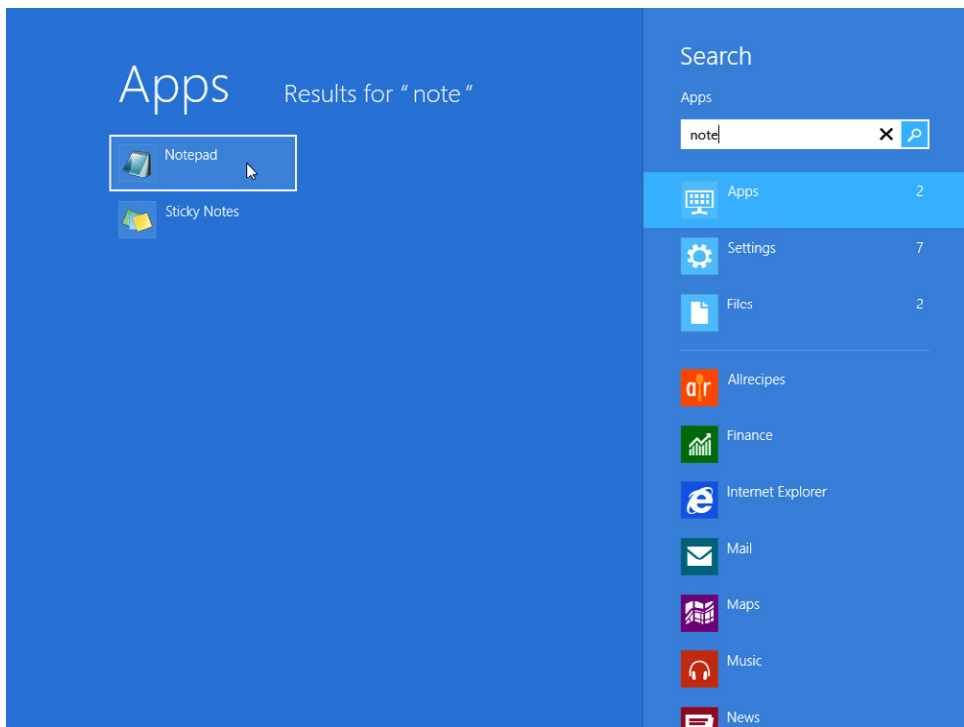


Note You may run into various extensions on webpage files on the Internet, such as *.php*, *.asp*, and *.jsp*. Those are all special formats designed for use with specific server technologies. This book covers developing only the basic type of webpage: the type with an *.htm* extension.

In the following exercise, you will open a webpage in Notepad and examine its text and tags.

Open a Webpage in Notepad

1. In Windows 8, press the Windows key to display the Start screen. Begin typing Notepad, and when you see Notepad at the left, click it.



Or, if you're working in an earlier Windows version, from the Start menu, click All Programs | Accessories | Notepad.



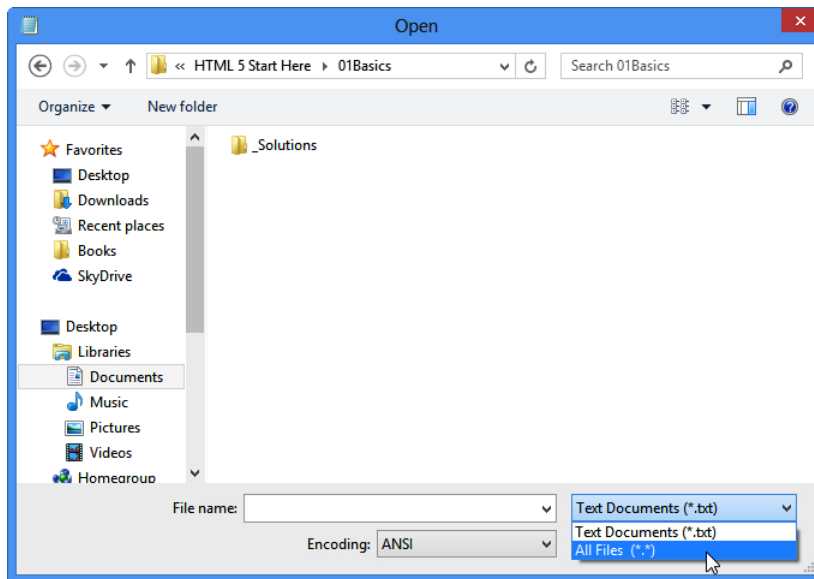
Note Because you will be using Notepad extensively in this book, you might want to create a shortcut for Notepad on the taskbar. To do so in Windows 8, after locating the Notepad app as shown in the previous figure, right-click its tile, and then click Pin to Taskbar. Or, in an earlier Windows version, after locating Notepad on the Accessories submenu, right-click it there and then click Pin to Taskbar.

2. In the untitled Notepad window, click File | Open.
3. Navigate to the folder containing the practice files for this chapter. To do so, on the Places bar, click Documents (or My Documents if you are using Windows XP). In the Open dialog box, double-click Microsoft Press, double-click HTML 5 Start Here, and then double-click 01Basics.



Note You won't see any files in the list at this point. The only thing that you should see is a `_Solutions` folder. (That folder contains the solution files for the lesson, but you don't need those now.)

4. Click the Files Of Type drop-down arrow and then click All Files.



5. In the Open dialog box, click `welcome.htm` and then click Open.

The `welcome.htm` file opens in Notepad.



Note The *.htm* extension might not appear on the welcome file in the Open dialog box. By default, file extensions for known file types are turned off in Windows. To turn them on, open File Explorer (Windows 8), Computer (Windows Vista or 7), or My Computer (Windows XP). In Windows 8, on the View tab, select the File name extensions check box. In earlier Windows versions, click the Tools menu (press Alt for the menu bar if you don't see it) and then click Folder Options. On the View tab of the Folder Options dialog box, clear the Hide Extensions For Known File Types check box and then click OK.

```
welcome.htm - Notepad
File Edit Format View Help
<!DOCTYPE HTML>

<html>
<head>
<title>Margie's Travel</title>
</head>
<body>
<h1>Margie's Travel</h1>
<p>Let us show you the world!</p>
</body>
</html>
```

6. Locate the `<html>` and `</html>` tags.

These tags signify the beginning and end of the HTML. They are typically the first and last tags in document, with the exception of the DOCTYPE, which you'll learn about later.

7. Locate the `<body>` and `</body>` tags.

These tags signify the beginning and end of the visible portion of the webpage when viewed in a browser.

8. Locate the `<p>` and `</p>` tags.

These tags signify the beginning and end of a paragraph.

Leave Notepad open for later use. You don't have to save your work because you didn't make any changes.

Other Ways of Opening Webpages in Notepad

A quick way to open most file types in their default applications is to double-click them from any File Explorer (or Windows Explorer) window. However, the problem with doing that for HTML files is that the default application is your web browser, not Notepad. So when you double-click an HTML file, the file doesn't open in Notepad; the file opens in your web browser. One way to get around this is to right-click a file in File Explorer, point to Open With, and then click Notepad (see Figure 1-1). This opens Notepad and loads the file.

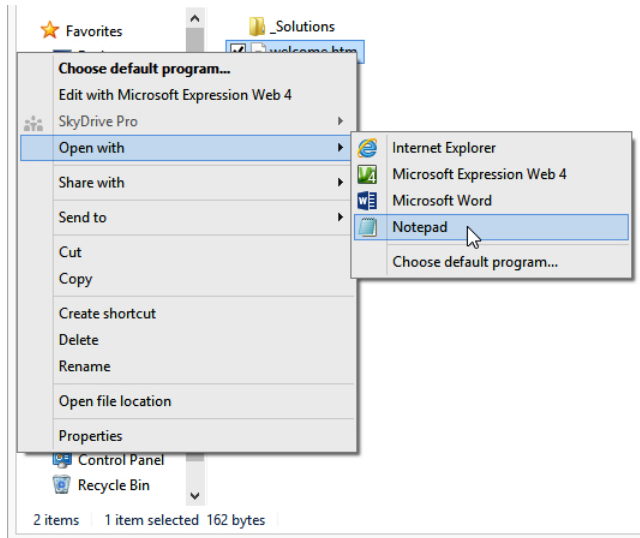


FIGURE 1-1 You can open a file with Notepad by right-clicking the file, pointing to Open With, and then clicking Notepad.



Note File Explorer is the file management interface in Windows 8. In earlier versions of Windows, that same interface is called Windows Explorer.

Here's another method: You can create a shortcut to Notepad on your desktop and then drag-and-drop individual HTML files onto that shortcut whenever you want to open an HTML file in Notepad. To create a desktop shortcut for Notepad in Windows 8, locate Notepad from the Start screen, right-click its tile, and then click Open File Location. In the File Explorer window that displays, right-click Notepad, point to Send To, and then click Desktop (Create Shortcut). See Figure 1-2. In earlier Windows versions, locate Notepad on the Start menu (Start | All Programs | Accessories), right-click it, point to Send To, and then click Desktop (Create Shortcut).

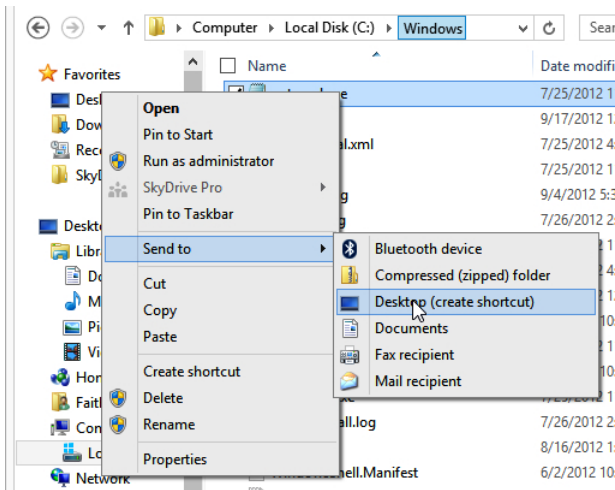


FIGURE 1-2 To copy the Notepad shortcut to the desktop, right-click it, point to Send to, and then click Desktop (create shortcut).



Caution Theoretically, you could set Notepad as the default application for opening files that have an `.htm` or `.html` extension. But that would cause more problems than it's worth, because then *all* your HTML content would open in Notepad not just the pages you are using for these lessons.

Adding a Location to the Favorites List

While working through this book, you will open many files in Notepad. To save yourself the trouble of navigating to the data file folder each time, you might want to add that folder to your Favorites bar, which displays in the navigation pane in the Open dialog box.

In the following exercise, you learn how to add a shortcut that brings you directly to the HTML 5 Start Here folder from the Favorites list.

Add a Location to Your Favorites List

1. In Notepad, click File | Open.



Note You can actually do this in almost any application (especially the Microsoft ones), but Notepad is handy because you worked with it in the preceding exercise.

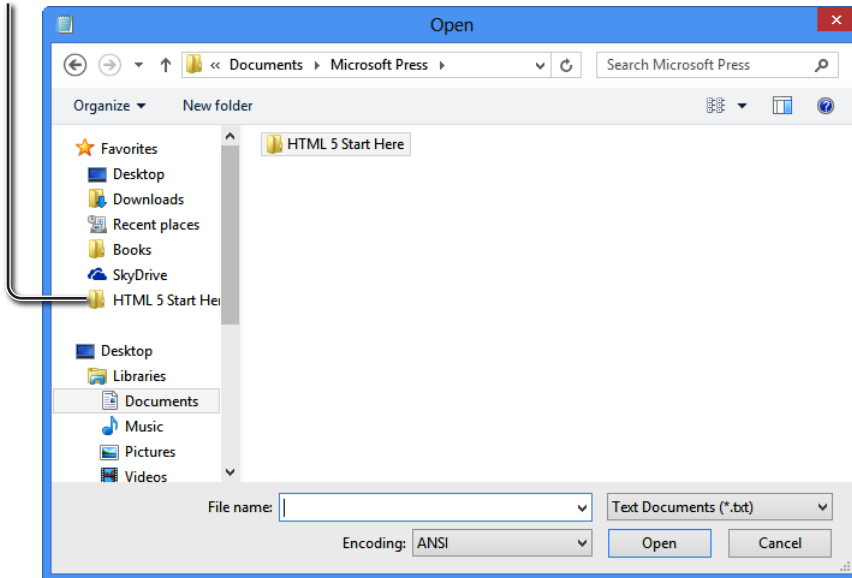
2. Navigate to the folder containing the practice files for this book.

On the Places bar, click Documents (or click My Documents if you are using Windows XP). In the Open dialog box, double-click Microsoft Press. The HTML 5 Start Here folder displays as an icon. Depending on your View setting, the icon might be a different size than shown here. The icon size is not important for the task at hand.

3. Drag the HTML 5 Start Here folder icon to the Favorites list on the left side of the window.

A shortcut for that folder displays on the Favorites list.

Drag the
folder here



Now, the next time you want to open a file in the Open dialog box, you can click that shortcut, and then double-click the folder for the chapter you are working on, which is much more convenient!

Previewing HTML Files in a Web Browser

Because Notepad is not a WYSIWYG (“What You See Is What You Get”) program, you won’t be able to immediately see how the tags you type will look in the finished product. To work around this, most people like to keep a browser window open next to Notepad so they can see their work by looking at their browser.

You can preview your work in any browser; you do not need to use Internet Explorer 10 (although that’s what I use in this book’s examples). In fact, as you progress with your web development skills, you will probably want to acquire several different browsers in which to test your pages, because each browser might display page elements a little differently. For beginners, though, Internet Explorer is a good choice because it’s one of the most popular browsers—one that a large percentage of your target audience is likely to be using. Other popular browsers include Google Chrome, Firefox, Safari, and Opera.

If you are using Windows 8, you have two different versions of Internet Explorer: the regular desktop version and the customized Windows 8 app version. You can start the desktop version of Internet Explorer from the Internet Explorer icon that’s pinned to the left of the taskbar on the desktop. You can start the custom version from the Internet Explorer tile on the Start screen. This book uses the desktop version of Internet Explorer, but you might want to check your work in the custom version when you start developing your own sites.



Caution Versions of Internet Explorer prior to version 10 do not support some of the HTML5 features. (Internet Explorer 9 supports most of the features, but not all.) You will probably want to test your webpages in an earlier version of Internet Explorer to make sure that people who use those versions will be able to view your pages. But don’t use an early version of Internet Explorer as you work through this book’s examples; you won’t get the full effect of the new HTML5 features.



Tip If the video card in your computer has two monitor connectors on it, or if you have an additional video card that you could install in your system, you might want to set up two monitors side-by-side. That way you can work on your HTML code in Notepad on one monitor and display the page full-screen in Internet Explorer in the other. All recent versions of Windows support at least two monitors, and some versions support many more.

In the following exercise, you will display an HTML file in Internet Explorer. I use the desktop version of Internet Explorer 10 for the examples in this book. To see the displayed file and the underlying code at the same time, open the file in Notepad and then arrange the windows so that both are visible at once.

Open a Webpage in Internet Explorer

1. Using File Explorer (or Windows Explorer), open *Documents\Microsoft Press\HTML 5 Start Here\01Basics* and then double-click *welcome.htm*. The file opens in Internet Explorer (or your default browser, if you're not using Internet Explorer).



Note If Internet Explorer is not your default browser but you want it to be, from within Internet Explorer, click Tools | Internet Options, click the Programs tab, and then click Make Default.

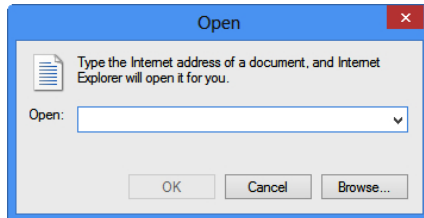
That's by far the easiest way if File Explorer is already open and displaying the file's location. Follow the next steps to learn another way that might be preferable when Internet Explorer is already open but File Explorer (or Windows Explorer) isn't.

2. In Internet Explorer, click File | Open.

The Open dialog box displays.



Note If the menu bar does not display in Internet Explorer, press the Alt key to display it.

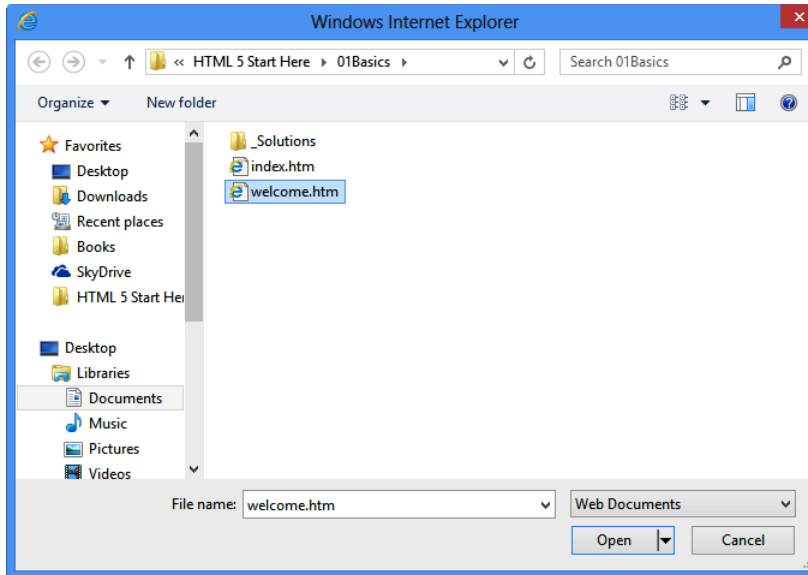


3. Click the Browse button and then browse to *Documents\Microsoft Press\HTML 5 Start Here\01Basics*.

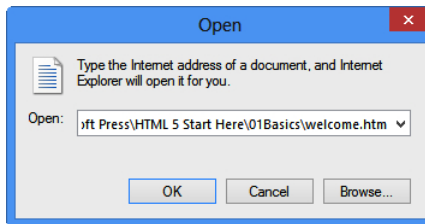


Tip If you created the shortcut in the Favorites bar earlier in the chapter, you can use that to save a few clicks when browsing for the location.

4. Click *welcome.htm* and then click Open.

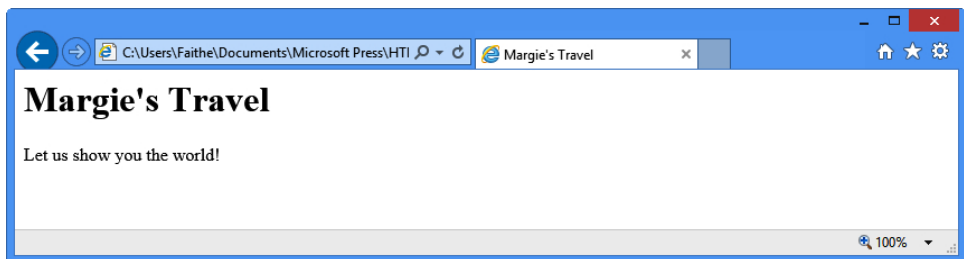


The path to the file displays in the Open dialog box.



5. Click OK.

The file opens in Internet Explorer. Leave Internet Explorer open for the next exercise.





Tip Remember, not all web browsers display pages the same way. For example, one browser's idea of what text should look like might be different from another. It's a good idea to check your pages in multiple web browsers, such as Firefox, Netscape, and Opera. These are available as free downloads from www.firefox.com, www.netscape.com, and www.opera.com, respectively.

Making, Saving, and Viewing Changes

After you've made a change to a webpage, you will probably want to see how that change looks in a browser. If you set up your Internet Explorer and Notepad windows side by side in the preceding two exercises, it's easy to view those changes. Simply save your work in Notepad and then refresh the display in Internet Explorer by clicking the Refresh button or pressing F5.

In the following exercise, you will change *us* to *me* in the *welcome.htm* file and then preview that change in Internet Explorer. This exercise builds on the previous two, so make sure you have completed them before starting this exercise. You can use this procedure throughout the rest of the book to preview your work from each exercise.

Make and View Changes to an HTML File

1. In both Notepad and Internet Explorer, reopen the *welcome.htm* file (if it is not still open) located in the *Documents\Microsoft Press\HTML 5 Start Here\01Basics* folder.

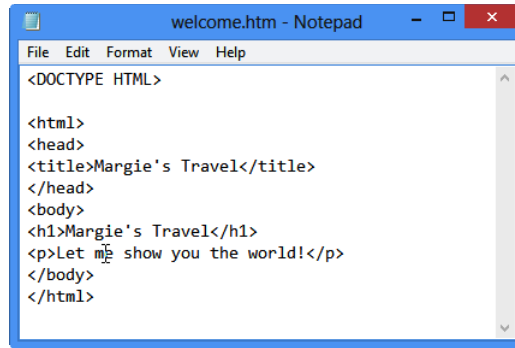
Remember, if you're using Windows 8, we're using the desktop version of Internet Explorer. In earlier versions of Windows that's not an issue because there is no custom version.

2. In Notepad, locate the word *us* and change it to *me*, as shown in the bold text in the following code:

```
<p>Let me show you the world!</p>
```

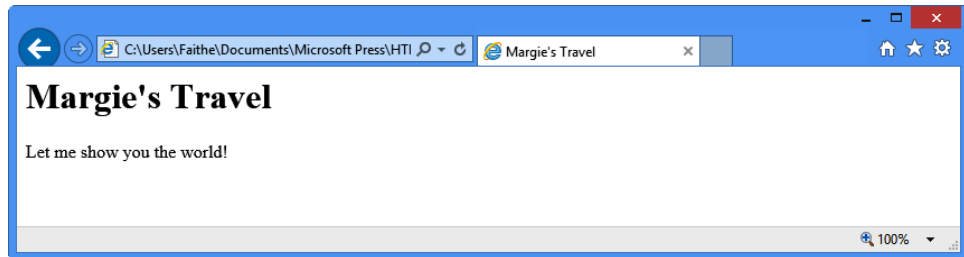
3. Save your work (File | Save).
4. Switch to Internet Explorer and click the Refresh button (or press F5).

The webpage now displays *me* rather than *us*.



```
welcome.htm - Notepad
File Edit Format View Help
<DOCTYPE HTML>

<html>
<head>
<title>Margie's Travel</title>
</head>
<body>
<h1>Margie's Travel</h1>
<p>Let me show you the world!</p>
</body>
</html>
```



5. Close Internet Explorer and close Notepad.

Key Points

In this chapter, you explored HTML and you were introduced to some basic skills that will be essential in following along with the exercises in upcoming chapters. Here are the key points to remember from this chapter:

- Any plain text editor, including Notepad, can be an HTML editor.
- Most webpages have an *.htm* or *.html* extension. You can open these pages in Notepad, but first you need to change the Files Of Type setting in the Open dialog box to All Files. You must change this setting each time you use the Open dialog box.
- An alternative way to open a webpage in Notepad is to right-click it in File Explorer, point to Open With, and then click Notepad.
- To preview a page in a web browser, click File | Open from the browser's menu.
- You can double-click an *.htm* or *.html* file in File Explorer to open it automatically in your default web browser.
- To see the changes you make in Notepad reflected in your web browser, save your work in Notepad and then, with the file already displayed in your browser, click Refresh in the browser window.

Setting Up the Document Structure

In this chapter, you will:

- Specify the document type
- Create the HTML, head, and body sections
- Create paragraphs and line breaks
- Specify a page title and keywords

Every society needs an infrastructure with certain rules that everyone agrees to for the general public good. For example, we have all agreed that a red light means “stop” and a green light means “go.” Everyone who wants to participate in the transportation system must play by those rules. If not, chaos ensues.

HTML is the same way. You can get creative with your web content, but there must be an underlying structure in place for web browsers to read and render your webpages properly. That means the document must contain certain tags that identify its major sections and that indicate to the browser what type of coding the document uses.

In this chapter, you’ll learn how to structure a document with the correct underlying tags. You’ll learn how to specify the type of HTML you are writing and how to create Head sections and Body sections. You’ll also learn how to create paragraph breaks and line breaks, how to specify a page title, and how to enter hidden keywords by which your page can be found in search engines.



Important Before you can use the practice files provided for this chapter, you need to download and install them from the book’s companion content location. See “Code Samples” at the beginning of this book for more information.

Specifying the Document Type

The *document type* tells the browser how to interpret the tags. Including a document type tag is not strictly necessary, but if you're trying to get into good HTML habits from the get-go, you'll want to make sure it's included.

When creating an HTML5 document, the first line of the document should be this tag:

```
<!DOCTYPE html>
```

The *DOCTYPE* tag always begins with an exclamation point and is always placed at the beginning of the document, before any other tag. Most HTML tags are not case-sensitive, but the word *DOCTYPE* should always be uppercase.

Using the *DOCTYPE* tag is like signing a contract. It is an optional tag, but when you use it, you are promising that your coding will conform to certain standards. When a web browser encounters a *DOCTYPE* tag, it processes the page in standards mode. When it doesn't encounter the *DOCTYPE* tag, it can't assume that the page conforms to a standard, so it processes the page in quirks mode. When the browser sees the tag `<!DOCTYPE html>`, it assumes you are using HTML5.

The distinction between *standards mode* and *quirks mode* came about in earlier days, when there were problems with standardization between web browsers. With some browsers, to display pages properly, you needed to get a little creative with the HTML code. Modern HTML coding does not allow that, but some older pages still include these obsolete workarounds. By using the *DOCTYPE* tag, you are making a promise to the web browser that there is nothing but pure HTML code in the page. (And, since I'm teaching you to do everything the *right* way, that's a reasonable promise you can make.)

Earlier versions of HTML used more complex *DOCTYPE* tags. If you were using HTML 4.01, for example, the syntax for the tag would be:

```
<!DOCTYPE HTML PUBLIC "-//W3C/DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

And, if you were using XHTML, the syntax for the tag would be:

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



Note If you are writing XHTML code, the *DOCTYPE* tag is required.

Creating the HTML, Head, and Body Sections

All your HTML coding—except the *DOCTYPE* tag—should be placed in an `<html>` section. Recall from Chapter 1 that when a tag is two-sided, as `<html>` is, it requires a corresponding closing tag that is identical to the opening tag but contains a slash immediately after the opening angle bracket (for example, `</html>`). The tags `<html>` and `</html>` serve as a “wrapper” around all the other tags in the document except the *DOCTYPE* tag.

In addition, your HTML file should have two sections: a Head section and a Body section. The *Head section* is defined by the two-sided tag `<head>`. The Head section contains the page title, which is the text that will display in the title bar or page tab of the web browser. It also includes information about the document that is not displayed, such as its metatags (which you’ll learn about later in this chapter). You can also include lines of code that run scripts, like Javascript.

The *Body section* is defined by the two-sided `<body>` tag and it contains all the information that displays in the web browser when you view the page.



Note The `<html>`, `<head>`, and `<body>` tags are all optional in HTML, but you should still use them because it’s a good design practice. They are required in XHTML. In addition, in XHTML you must add an argument to the `<html>` tag that declares its XML namespace, a reference to the fact that XHTML is created within XML (as you learned in Chapter 1). Here’s how the opening `<html>` tag should look in an XHTML document:
`<html xmlns="http://www.w3.org/1999/xhtml">`.

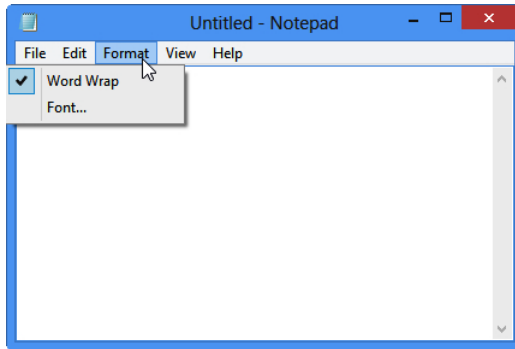
In the following exercise, you will create an HTML5 template file that you can reuse later for your own work. That way you don’t have to retype those same *DOCTYPE*, head, and body tags every time you want to create a new webpage.

Create an HTML5 Template

1. Open Notepad and then click the Format menu. Word Wrap should have a check mark next to it. If it does not, click it to enable the Word Wrap feature.



Tip Using Word Wrap makes it easier to see long lines of HTML coding without scrolling.



2. In the Notepad window, type the following:

```
<!DOCTYPE html>
```

3. Press Enter and then type:

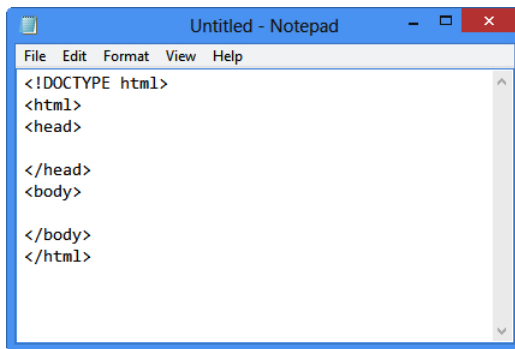
```
<html>  
<head>
```

4. Press Enter two or three times to insert some blank lines and then type:

```
</head>  
<body>
```

5. Press Enter two or three times to insert some blank lines and then type:

```
</body>  
</html>
```



6. Save the file as *HTML5.htm* on your Windows desktop (or to any other location that is convenient for you).



Note Most of the files you work with in this book will be stored elsewhere, but you might find it helpful to keep the templates created in this exercise handy for reuse. The desktop is a convenient place to store them. Of course, you can store them anywhere you like.

7. Close Notepad.

You now have a template for creating any HTML documents you like. You can reopen this template file and save it under different names.



Tip If you want to avoid accidentally editing the template in the future, make it read-only. To do so, in Windows Explorer, right-click the file and then click Properties from the contextual menu. In the Properties dialog box, select the Read-Only check box. When you try to save changes to a read-only file, an error message displays and a Save As dialog box prompts you to save a copy of it with a new name.

Creating Paragraphs and Line Breaks

Within the `<body>` section of the document, you type the text that will display on the webpage. Each paragraph of text should be enclosed in a two-sided tag that indicates its type.

The most basic paragraph type is the body paragraph, indicated by the `<p>` tag. It is a two-sided tag, so the paragraph text is placed between a `<p>` and a `</p>`.



Note In HTML, the code will still work even if the `</p>` is omitted; in XHTML, however, it won't work without a closing tag. Even if you never plan on coding in XHTML, it is a good practice to include the `</p>` tag. This way, you won't fall into any sloppy habits.

When a browser displays a webpage, it inserts vertical white space between paragraphs (see Figure 2-1).

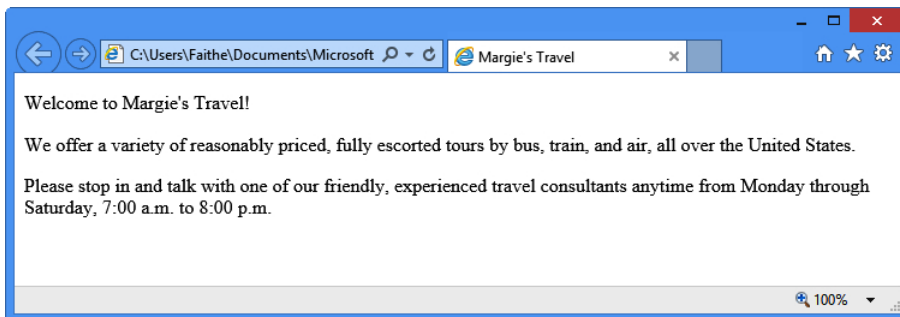


FIGURE 2-1 The browser inserts a blank vertical space between each paragraph.

That spacing is usually convenient, but it can be a problem when the extra space between lines is unwanted, such as with an address (see Figure 2-2).

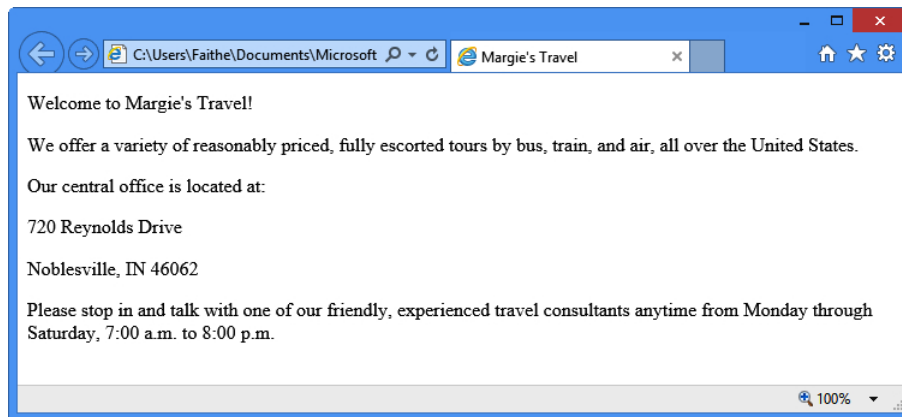


FIGURE 2-2 Sometimes you do not want extra space between each line, as shown in this example.

To create a line break without officially starting a new paragraph (and thereby adding that extra white space), use the line break tag: `
`. This is a one-sided tag placed within a paragraph, at the end of each line, like this:

```
<p>720 Reynolds Drive<br>Noblesville, IN 46062</p>
```



Note In XHTML, the line break tag is `
`. The end slash (and the space preceding it) is necessary to indicate that it's a self-closing tag. Notice that the slash is placed after the letters, not before, as with the closing end of a two-sided tag. In XHTML, one-sided tags must end with a slash to indicate that they are self-closing. The space between the text and the final slash is also required, so the tag will be recognized in HTML.

In the following exercise, you will add text to an HTML file template and then preview it in Internet Explorer.

Add Text to an HTML File

1. Open the *HTML5.htm* file located in the *Documents\Microsoft Press\HTML 5 Start Here\02Structure\CreatingParagraphs* folder.
2. Save the file as *index.htm* in the *Documents\Microsoft Press\HTML 5 Start Here\02Structure* folder.



Note It is customary to name the opening page of a website *index.htm*, *index.html*, *default.htm*, or *default.html*. When users type a URL in their web browsers but omit the file name (like typing *www.microsoft.com* rather than *www.microsoft.com/filename.htm*), most servers will automatically respond with the index or default page (if one exists).

3. In Internet Explorer, open the *index.htm* file and arrange the Notepad window and the Internet Explorer window so that both are visible.

The index file displayed in Internet Explorer is currently blank.

4. In the Notepad window, type the following text between the `<body>` and `</body>` tags:

```
<p>Welcome to Margie's Travel!</p>
```

```
<p>We offer a variety of reasonably priced, fully escorted tours by bus,  
train, and air, all over the United States.</p>
```

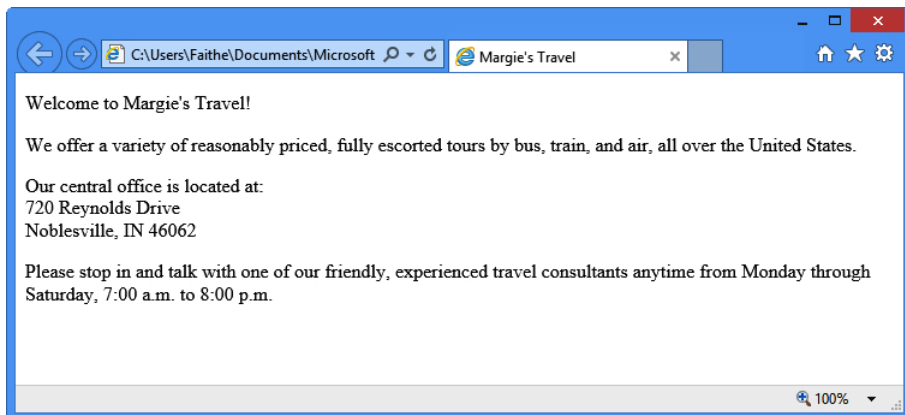
```
<p>Our central office is located at:<br>
```

```
720 Reynolds Drive West<br>
```

```
Noblesville, IN 46062</p>
```

```
<p>Please stop in and talk with one of our friendly, experienced travel  
consultants anytime from Monday through Saturday, 7:00 a.m. to 8:00 p.m.</p>
```

5. Save your work and then switch to Internet Explorer and press F5 or click the Refresh button to see the result of the changes. Leave both windows open for the next exercise.



6. Close Notepad and Internet Explorer.



Note Your screen might look slightly different, depending on the settings you have configured in your browser.

Specifying a Page Title and Metatags

Perhaps you noticed in the preceding exercise that the complete path to the file displayed in the page tab or in the browser title bar (depending on the browser you are using). Usually when you view a webpage, a friendly, descriptive title displays in that spot instead. That text is specified by a `<title>` tag placed in the `<head>` section (also called the header). Here's an example:

```
<head>  
<title>Margie's Travel</title>  
</head>
```



Troubleshooting Make sure you place the `<title>` tag in the `<head>` section; don't place it in the `<body>` section.

Another element you can place in the header is the `<meta>` tag. The `<meta>` tag has several purposes. One of these is to identify keywords related to your page. Placing appropriate keywords on your page can make it easier for people to find your page when they are searching the web using a search engine. When some search engines index your page, they rely not only on the full text of the page, but also on any keywords they find in the `<meta>` tag area.



Note Not all search engines refer to `<meta>` tags. Google does not, for example; it indexes only the text contained in the `<body>` area. Because of the potential for abuse of the system, such as web developers packing their pages with unrelated keywords, fewer and fewer search engines these days are using them.

For example, suppose the Margie's Travel site would be useful to people who are searching for guided travel tours and vacations that are customized for senior citizens. Perhaps the phrase "senior citizen" is not mentioned on the main page, but you want people who search for that phrase to be directed to the main page anyway. You could place the following in the `<head>` section:

```
<meta name="keywords" content="senior, senior citizen, travel, tours">
```

Notice that the `<meta>` tag in this code is a single-sided tag that contains two attributes: name and content. The values for each of those arguments follow the equals sign and are contained in double quotation marks.



Note If you are coding in XHTML, you would add a space and a / at the end of `<meta>` tag because it is a one-sided (self-closing) tag. This is not necessary in HTML.

The `<meta>` tag can also be used to redirect visitors to another page. For example, suppose you told everyone the address of your website and then you needed to move it to another URL. You could place a “We’ve Moved” page at the original address and use the `<meta>` tag to redirect users to the new address after five seconds, like this:

```
<meta http-equiv="refresh" content="5; url=http://www.contoso.com/newpage.htm">
```

Here’s yet another common use: the `<meta>` tag can specify a character encoding scheme. This is not a big issue if you are coding only in English (or in a language like English that uses a Roman character set), but it is considered a tidy coding practice to include anyway. If you want, you can add `<meta charset="utf-8">` to the `<head>` section of your document to explicitly spell out that your page is in English.

In the following exercise, you will add a page title and some keywords to the `index.htm` page you created in the preceding exercise.

Specify a Page Title and Metatags

1. In Notepad, open the `index.htm` file from the previous exercise. If you have not completed the previous exercise, open the `index.htm` file located in the `Documents\Microsoft Press\HTML 5 Start Here\02Structure\SpecifyingTitle` folder.
2. Between the `<head>` and `</head>` tags, type the following text to create the page title:

```
<title>Margie’s Travel</title>
```

3. After the title, type the following `<meta>` tag:

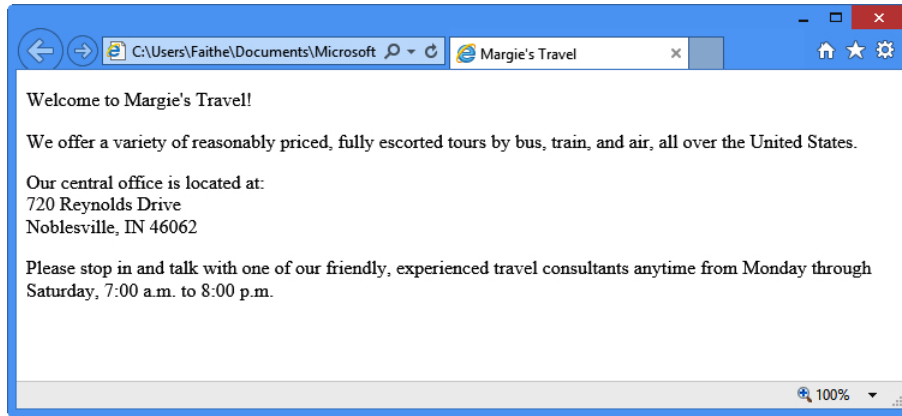
```
<meta name="keywords" content="senior, senior citizen, travel, tours">
```

Press Enter to start a new line and then type the following `<meta>` tag:

```
<meta encoding="utf-8">
```

4. Save your work and then view the file in Internet Explorer.

The tab displays the site name, but notice that the inclusion of the `<meta>` tags caused no apparent difference in the displayed text of the page. This is because the keywords and encoding specification do not display on the webpage itself.



5. Close Notepad and Internet Explorer.

Key Points

In this chapter, you learned how to create the overall structure of a document and how to divide it into head and body sections. You learned how to create paragraphs and how to add a page title. Here are the key points to remember from this chapter:

- To specify HTML5 as the document type, type `<!DOCTYPE html>` at the beginning of the file.
- All the HTML coding in a document (except the `DOCTYPE`) is enclosed within a two-sided `<html>` tag.
- The `<html>` and `</html>` tags enclose the `<head>` and `<body>` sections.
- The `<head>` area contains the page title (`<title>`) and any `<meta>` tags. The `<body>` area contains all the displayable text for the page.
- Enclose each paragraph in a two-sided `<p>` tag. Most browsers add space between paragraphs when displaying the page.
- To create a line break without starting a new paragraph, use the one-sided `
` tag.
- When coding for XHTML, end one-sided tags with a space and a slash (`/`). The space is required for recognition in HTML, and the slash is necessary for recognition in XHTML.
- Use the `<title>` and `</title>` tags to enclose the text that should display in the browser's title bar. Place these in the `<head>` section of the file.
- Use `<meta>` tags in the `<head>` section to indicate keywords and the document encoding language.

Index

Symbols

- ¢ (cent) entity, 59
- £ (pound) entity, 59
- ¥ (yen) entity, 59
- 404 error message, 176
- & (ampersand) entity, 59
- .asp (file extension), 7
- .css file extension, 100
- { } curly braces, 88
- † (dagger) entity, 59
- ° (degree) entity, 59
- > (greater than) entity, 59
- .htm (file extension), 6
- .html (file extension), 6
- .jsp (file extension), 7
- < (less than) entity, 59
- @nbsp; code, 168
- (nonbreaking space) entity, 59
- .php (file extension), 7
- ± (plus or minus) entity, 59
- # (pound sign), 80

A

- AAA buttons, 169
- AAC (audio encoding format), 264
- <abbr> tag, 316
- absolute paths, 73–74
- accessibility, 313–320
 - absolute division positioning and, 188
 - active content and, 317
 - alt attribute (tags), 155
 - autoplay attribute (<video> tag) and, 267
 - color choices and, 314
 - common issues with, 313–314

- complex page layouts and, 319
- device independence and, 318
- divisions and, 179
- embedded objects and, 318
- flashing text and, 116
- graphical navigation bars and, 171
- graphics and, 155–156
- <h> tags and, 31
- interim solutions for older browsers, 318–319
- language usage, specifying, 316
- navigational aids and, 320
- navigation bars and, 166
- new technologies, providing alternatives to, 317
- page layout choices and, 315
- ScreenTips, 77
- simplicity, 320
- tables and, 199, 316–317
- title option (hyperlinks), 77
- W3C technologies/guidelines, 319
- Accessibility Properties dialog box (Expression Web), 289
- active content, accessibility and, 317
- active (hyperlink pseudo-class), 99
- Adobe Reader, 82
- align attribute (<td> tags), 238
- alignment (table cells), 236
- alt attribute (tag), 155
- & (ampersand) entity, 59
- anchors, 79–81
 - linking to, 80
 - setting, 79
- Android, <audio>/<video> tag support in, 262
- animations in GIF files, 142
- Apple iPhone browser
 - <audio>/<video> tag support in, 262
 - email/URL text boxes and, 244
- <article> tags, 179, 180

<aside> tags

- <aside> tags, 179, 180
- ASP (Active Server Pages), 7
 - default.aspx and, 73
- .asp (file extension), 7
- <a> tags, 5, 71–84
 - anchors, 79
 - graphical hyperlinks and, 151
 - href attribute, 72
 - name attribute, 79
 - subject attribute, 76
 - target attribute, 74
 - title attribute, 77
- audio
 - embedding, 271–273
 - playing on older browsers, 272–273
- <audio> tags, 271–272
 - <source> element, 272
- Audio Video Interleave (.avi) (multimedia container format), 263
- autoplay attribute (<video> tags), 267
- autostart attribute (<embed> tag), 272

B

- background-color style rule (<div> tag), 195
- background-image style (<div> tag), 195
- background-repeat= attribute (<body> tag), 69
- backgrounds, 64–70
 - colors, setting, 65
 - image files, using as, 67–70
- bandwidth and preloading multimedia, 267
- <bgsound> tag, 317
- BigNoseBird.Com, 258
- blinking text, readability and, 116
- blink (text-decoration attribute), 116
- block-level element, 134
- <blockquote> tags, 42–44
 - <p> tags and, 43
- <body> tag, 21–23
- bold text, 33–35
- border attribute (<table> tags), 224–226
- border-color attribute (<table> tags), 226
- border-color attribute (<td> tags), 226
- borders (graphics), 151–152
- borders (paragraphs), 129–134
 - attributes, setting, 132
 - color, 131
 - default style for, 131
 - order of, in style attribute, 132

- padding, 130
- padding attribute, 130
- sides, formatting individually, 131–132
- width, 131
- borders (tables), 224–232
 - attributes, applying with, 224–226
 - styles, applying to, 226–232
- border-style attribute (<table> tags), 226
- border-style attribute (<td> tags), 226
- border-width attribute (<table> tags), 226
- border-width attribute (<td> tags), 226
- bottom attribute (borders), 131
- breadcrumbs, 311
-
 tag, 24
 - XHTML and, 24
- tags, 33–35
- bulleted lists, 50–56
 - tags, 50
- bulletin boards, 258
 - phpBB, 258
 - ProBoards, 258
- Button Generator, 170
- buttons, 169–170

C

- cascading style sheets. *See* CSS (cascading style sheets)
- cellpadding attribute (<table> tags), 237
- cellpadding attribute (<td> tags), 237
- cellspacing attribute (<table> tags), 237
- cells (tables), 235–239
 - alignment, setting, 238–239
 - padding, 237
 - spacing, 237
- ¢ (cent) entity, 59
- The CGI Resource Index, 258
- CGIs (Common Gateway Interface), 257–258
 - security and, 257
- character based formatting, 106–124
 - bold, including in styles, 114–115
 - font families, specifying, 106–109
 - italics and, 114–115
 - size/color, specifying, 109–113
 - spacing (letter/word), adjusting, 120–123
 - strikethrough formatting, 116–118
 - underlining and, 116–118
- character encoding
 - meta tags and, 27

checkboxes in forms, 249
 checked attribute (<input> tags), 250
 Chrome (Google)
 audio formats supported by, 272
 <audio>/<video> tag support in, 262
 tables and, 210
 cite="URL" attribute (<blockquote> tag), 42
 class attribute (style tags), 95
 clear style, 146–148
 Code pane (Expression Web), 286
 Code tab (Expression Web), 277
 <code> tags, 37
 colors
 accessibility and, 314
 basic names for, 62
 hexadecimal values of, 65
 RGB (red-green-blue) values of, 64–65
 web-safe, 65
 cols attribute (<textarea> tags), 244
 colspan attribute (<td> tags), 213
 Common Gateway Interface (CGI) script, 257–258
 Comprehensive Perl Archive Network, 257
 controls attribute (<video> tags), 267
 copyright symbol (entity), 59
 Courier font, 38
 Crystal Button (button generation program), 170
 CSS (cascading style sheets), 87–104
 bold, including in styles, 114–115
 character based formatting, 105–124
 class attribute (style tags), 95
 classes, 95–98
 division-based layouts and, 179
 divisions, formatting with, 195–198
 external style sheets, 100–102
 <figure> tag and, 157
 font families, specifying, 109–113
 font size/color, specifying, 109–113
 font-weight attribute, 114
 formatting of, 89–92
 hyperlink formatting and, 98–100
 IDs, 95–98
 inline spans, creating, 118–120
 italics, including in styles, 114–115
 linking to external, 100
 nested tags and, 93–95
 padding styles, overriding, 134
 paragraphs, formatting with, 125–140
 positioning divisions in, 188
 spacing (letter/word), adjusting, 120–123
 style rules, defining, 88–89

table borders, applying with, 226–232
 tag formatting vs., 105
 underlining/strikethrough formatting, specifying, 116–118
 unintended consequences of global tag changes, 129
 .css file extension, 100
 CSS Properties tab (Expression Web), 294
 CSS templates (Expression Web), 285–288
 changing document type of, 286
 {} curly braces, 88

D

†(dagger) entity, 59
 <dd> tags, 57
 default (webpage), 72
 definition list, 56–58
 <dd> tags, 57
 <dl> tags, 56
 <dt> tags, 56
 indenting, 126
 ° (degree) entity, 59
 tag, 116
 Design pane (Expression Web), 287
 Design tab (Expression Web), 278
 device independence, 318
 division-based layouts
 accessibility and, 179
 creating divisions, 182–184
 positioning, 187–194
 divisions
 absolute positioning, accessibility and, 188
 background images, applying in Expression Web, 290
 formatting in Expression Web, 297–299
 <div> tags, 179
 background-color style, 195
 background-image style, 195
 float style rule, 187–188
 id attribute, 182–184
 positioning attribute, 188
 position style rule, 188
 <section> tags vs., 180
 DivX (video encoding format), 263
 <dl> tag, 56
 <!DOCTYPE> tag, 20
 XHTML and, 20

document structure

- document structure, 19–28
 - body section, 23–26
 - metatags, adding, 26–28
 - page title, specifying, 26–28
 - required sections of, 21–23
- document type, specifying, 20
- Dreamweaver (web development application), 275
- <dt> tag, 56

E

- e-mail address (as hyperlink), 76–79
 - avoiding/dealing with spam, 76
 - default subject line, setting, 76
- <embed> tags
 - audio, playing with, 272
 - loop attribute, 273
 - older browsers and, 262
 - video files, playing with, 268–269
- ems (font size), 110
- enctype attribute (<form> tags), 242
- entities, 58–60
- EOS Development, 169
- error pages, custom, 175–177
- Expression Web (Microsoft), 275–304
 - autocoding in, 293
 - background images, adding, 290–292
 - CSS templates, creating pages with, 285–288
 - divisions, formatting, 297–299
 - file type, specifying, 283
 - hyperlinks, inserting, 299–302
 - image folders, importing, 288–289
 - images, placing on the page, 289–290
 - interface, 276–281
 - new website, creating with, 281–285
 - opening an existing page in, 277–281
 - text, formatting, 292–297
 - trial version, source for, 275
 - websites, creating, 281–288
- extended names (of colors), 65
- Extensible Markup Language (XML). *See* XML (Extensible Markup Language)

F

- Favorites bar, adding folders to, 11–12
- <figcaption> tag, 156
- <figure> tag, 156–158
- tag vs., 143

- File Explorer (Windows 8)
 - favorites bar, adding folders to, 11
 - vs. Windows Explorer, 10
- file extensions
 - default applications, working around, 9
 - forcing Windows to display, 9
 - various types, 7
- Firefox (Mozilla), 13, 16
 - audio formats supported by, 272
 - <audio>/<video> tag support in, 262
- first-line indent, 126
- Flash (Adobe)
 - audio objects and, 272
 - <embed> tags and, 268
 - testing video content for, 271
- flashing text, 116
- Flash Video (.flv or .f4v) (multimedia container format), 263
- float style rule (<div> tags), 187–188
- focus (hyperlink pseudo-class), 99
- Folder List pane (Expression Web), 286
- font families, 106–109
- fonts
 - absolute size of, 109
 - colors, specifying in a CSS, 111
 - tag, 116
 - ems, as measurement of, 110
 - <ins> tag, 116
 - relative size of, 110
- font-style attribute (<p> tags), 114
- tag, 29
- font-weight attribute (<p> tags), 114
- <footer> tags, 180
- foregrounds, 64–70
 - colors, setting, 66
- formatting, tags vs. CSS, 105
- formatting text. *See* text, formatting
- forms, 241–260
 - buttons, creating, 244–245
 - CGI and, 257–258
 - check boxes, 249–252
 - creating, 242–248
 - e-mail/URL specific boxes, 244
 - menu lists, 252–256
 - option buttons, 249–252
 - security concerns with sending as e-mail, 242
 - sliders, 256
 - spin boxes, 256
 - text areas, creating, 244
 - text boxes, 243

- <form> tags, 242
 - enctype attribute, 242
 - method attribute, 242
- 404 error message, 176
- frame attribute (<table> tags), 225
- frames, 310

G

- GIF (graphic file type)
 - animations and, 142
 - JPG/PNG file types vs., 142
 - transparent images and, 195
- GoDaddy.com, 176
- Google
 - meta tags and, 26
 - WebM and, 263
- Google Chrome (web browser), 13
 - <audio>/<video> tag support in, 262
 - tables and, 210
- graphics, 141–160. *See also* tags
 - accessibility and, 155
 - adding in Expression Web, 289
 - alternate text for, 155–156
 - borders, removing, 151
 - captions, adding, 156–158
 - common file types for, 142
 - editing, 153
 - horizontal rules, clearing, 146–148
 - hyperlink, using as, 151–152
 - inserting, 143–146
 - padding, controlling, 148
 - resizing, 148–151
 - resolution, 142
 - size, site performance and, 142
 - thumbnails, creating, 153–155
 - wrapping text around, 144
- > (greater than), 59
- guest books, 258

H

- H.264 (video encoding format), 263
- Handbrake (video encoding program), 266
- <header> tags, 180
- headings, 30–33
 - indenting, 126
- <head> section, 26
 - style sheets and, 87
 - <style> tag, 88–89

- <head> tag, 21–23. *See also* <head> section
- height attribute (<table> tags), 205
- <hgroup> tags, 31
- horizontal lines/rules, 60–64
 - clear style, attributes of, 146–148
 - graphics and, 146–148
- hover (hyperlink pseudo-class), 99
- href= attribute (<a> tags), 72
- <hr> tag, 60–64
- <h> tags, 30–33
 - margin attribute, 126
 - padding attribute, 126
 - text-indent attribute, 126
- .htm (file extension), 6
- HTML5
 - <a> tag, 71–84
 - <audio> tags, 271–273
 - audio/video in, 262
 - color names recognized by, 62
 - division-based layout tags, 180–181
 - entities, 58–60
 - <figcaption> tag, 156–158
 - <figure> tag, 156–158
 - tag, 29
 - hyperlinks, creating, 71–84
 - tag, 143–146
 - <nav> tag, 166
 - sliders, 256
 - tags, 118–120
 - spin boxes, 256
 - units of measurement accepted by, 109
 - <video> tags, 266–271
- HTML documents
 - background colors, defining, 65
 - background/foreground colors, choosing, 64–70
 - body section, 23
 - changes, viewing "on the fly", 16–17
 - compatibility and older browsers, 13
 - document type, specifying, 20
 - generally, 3–6
 - lists, defining, 49–58
 - metatags, 26–28
 - opening/editing in Notepad, 6–12
 - page title, 26–28
 - previewing, 13–16
 - required sections of, 21–23
 - structure of, defining, 19–28
 - templates, creating, 21
 - text, formatting, 29–48
- .html (file extension), 6

HTML (HyperText Markup Language)

HTML (HyperText Markup Language), 4. *See also* HTML5; *See also* HTML documents; *See also* tags (HTML)

<html> tag, 21–23

HTML tags. *See* tags (HTML)

http-equiv attribute (<meta> tags), 174

hyperlinks, 71–84

- anchors and, 79–81
- e-mail addresses as, 76–79
- filenames, 72–73
- generally, 5
- graphical, 151–152
- non-html content, pointing to, 82
- partial paths, 72–73
- paths, relative/absolute, 73–74
- ScreenTips, 77
- subfolders and, 73
- target windows, setting, 74–75
- viewers for non-html content, linking to, 82
- webpages, pointing to, 72–75

I

id attribute (style tags), 96

images. *See also* graphics

- accessibility and, 155
- borders, removing, 151
- padding, controlling, 148
- programs to edit, 153
- resizing, 148–151
- size, modifying through attributes, 148

 tags, 143–146

- alt attribute, 155
- border style, 151
- <figcaption> tag and, 156
- float style, attributes for, 144
- height attribute, 148–151
- history of, 4–5
- margin style, 149
- padding attribute, 149
- src attribute, 143
- width attribute, 148–151

index.htm(l), 72

inline spans, 118–120

<input> tags, 243

- checked attribute, 250
- value attribute, 249

Insert Hyperlink dialog box (Expression Web), 299

<ins> tag, 116

Internet Explorer 10, 13–16

- audio formats supported by, 272
- <audio>/<video> tag support in, 262
- default browser, setting as, 14
- extended color names and, 65
- MP4 files created in VLC Media Player and, 266
- refreshing webpages, 16
- <th> tags and, 209
- view settings, configuring, 44–47

Internet Explorer app, 13–16

Internet Explorer (older versions)

- HTML 5 features, support in, 13
- refreshing webpages, 16

iPhone browser

- <audio>/<video> tag support in, 262
- email/URL text boxes and, 244

<i> tags, 33–35

italic text, 33–35

J

JPEG file. *See* JPG (graphic file type)

JPG (graphic file type), 142

- transparency and, 195

.jsp (file extension), 7

K

<kbd> tag, 37

L

-left attribute (borders), 131

< (less than) entity, 59

letter spacing, 121

line breaks, adding, 24

line-height attribute (<p> tags), 137

lines, horizontal, 60–64

line-through (text-decoration attribute), 116

link (hyperlink pseudo-class), 98

lists, 49–58

- definition list, 56–58
- items, indenting, 126
- tags, 50
- nested, 51–52
- numbered/bulleted, 50–56

 tags, 50

- value="n" attribute, 53

literal text, 3
 lobby pages, 165
 loop attribute (<embed> tags), 273
 loop attribute (<video> tags), 267

M

margin attribute (<p> tags), 126
 arguments for, 127
 units of measurement for, 127
 margin indent, 126
 margin style attribute
 tag, 149
 Matroska (.mkv) (multimedia container format), 263
 Matt's Script Archive, 257
 maxlength attribute (<input> tags), 243
 menu lists, breaking into groups, 253
 <meta> tags, 26–28
 page redirection and, 174–175
 search engines and, 26
 XHTML and, 27
 method attribute (<form> tags), 242
 Microsoft Download Center, 82
 Microsoft Office viewers, 82
 Microsoft Visio, 164
 monitors, supporting multiple, 13
 MPEG-4 Audio Layer 3 (MP3) (audio encoding format), 264
 MPEG-4 Part 14 (.mp4) (multimedia container format), 263
 multimedia, 261–274
 audio, embedding, 271–273
 codecs for, 263–264
 formats, choosing, 264
 formats/containers for, 263
 plug-ins for content and, 262
 providing alternatives for accessibility, 314
 video, embedding, 266–271
 video, encoding, 264–266

N

<nav> tag, 166
 name attribute (<a> tags), 79
 name attribute (<input> tags), 243
 navigational aids, 163–178
 accessibility and, 320
 breadcrumbs, 311
 custom error pages, 175–177

graphical navigation bars, 169–173
 planning site organization, 164–165
 text-based navigation bars, 166
 url redirection, 173–175
 navigation bars
 buttons and, 169–170
 drop menus, 164
 graphical, 169–173
 laying out with tables, 216–220
 text based, 166–169
 navigation buttons, 169
 transparency and, 195
 navigation with hyperlinked graphics, 152
 (nonbreaking space) entity, 59, 168
 Netscape (web browser), 16
 none (text-decoration attribute), 116
 Notepad (Microsoft)
 file extensions, changing, 6
 and HTML document creation/editing, 6–12
 HTML documents, opening with, 7–12
 launching, 7
 Word Wrap feature, 21
 numbered lists, 50–56
 tag, 50

O

<object> tags, 262
 Ogg (.ogg) (multimedia container format), 263
 tag, 50
 start="n" attribute, 53
 Opera (web browser)
 audio formats supported by, 272
 <audio>/<video> tag support in, 262
 previewing files in, 13
 testing design in, 16
 option buttons, 250
 <optgroup> tags, 253
 <option> tags, 252
 underline (text-decoration attribute), 116

P

padding attribute (borders), 130
 padding attribute (tags), 149
 padding attribute (<p> tags), 126
 arguments for, 127
 units of measurement for, 127
 padding (images), 149–151

padding indent

- padding indent, 126
- padding (table cells)
 - defined, 235
 - setting, 237
- page layout, 179–198
 - accessibility concerns with, 315
 - CSS and, 195–198
 - division-based, 181–184
 - division-based vs. tables, 179
 - divisions, naming, 182–184
 - divisions, positioning, 187–194
 - semantic, HTML5, 184–186
 - tables and, 216–220
 - wrapping text around images, 144
- Paint Shop Pro, 153
- paragraphs
 - borders, applying, 129–134
 - creating, 23–26
 - formatting with CSS, 125–140
 - indenting, 126–129
 - margins vs. padding, 126
 - setting alignment, horizontal, 134–136
 - vertical space, setting, 137–138
- paths
 - relative vs. absolute, 73–74
 - subfolders, linking to documents in, 143
- performance, graphic size and, 142
- Photoshop, 153
- phpBB open-source bulletin board package, 258
- .php (file extension), 7
- pixels, 109
- placeholder attribute (<input> tags), 246
- plug-ins for multimedia content, 262
- ± (plus or minus) entity, 59
- PNG (graphics file type), 142
 - transparency and, 195
- position: absolute (<div> tag position attribute), 188
- position: fixed (<div> tag position attribute), 188
- position: relative (<div> tag position attribute), 188
- position style (<div> tags), 188–194
- £ (pound) entity, 59
- # (pound sign), 80
- practice files, navigating to, 8
- preformatted text, 38
- preload attribute (<video> tags), 267
- <pre> tag, 38
- ProBoards (bulletin board host), 258
- pseudo-classes, for hyperlinks, 98–100
 - recommended order of, when defining, 99
- <p> tags, 23–26

- <blockquote> tags and, 43
- color style, setting, 111
- font-family, setting, 107
- font-style attribute, 114
- font-weight attribute, 114
- line-height attribute, 137
- margin attribute, 126
- padding attribute, 126
- text-align attribute, 134–136
- XHTML and, 23

Q

- <q> tag, 42
- quirks mode (of web browsers), 20
- quotations
 - indenting, 126
 - tags for, 42–44

R

- radio buttons, 250
- redirecting to new URL, 27
- registered trademark (entity), 59
- relative paths, 73–74
- Reset button for forms, 245
- resolution (of graphics), 142
- RGB (red-green-blue) value, 64–65
- right attribute (borders), 131
- rows attribute (<textarea> tags), 244
- rowspan attribute (<td> tags), 213
- rules. *See* style rules
- rules attribute (<table> tags), 226

S

- Safari (Apple web browser), 13
 - audio formats supported by, 272
 - <audio>/<video> tag support in, 262
- <samp> tags, 37
- screen readers
 - alternate text for graphics and, 155
 - <h> tags and, 31
 - ScreenTips and, 77
 - tables and, 199
 - title attribute (<a> tags) and, 77
- ScreenTips, 77
- search engines, meta tags and, 26
- <section> tags, 180

- <div> tag vs., 180
- <select> tag, 252
- semantic layouts, 184–186
 - tags for, 180–181
- semantic tags, 30
 - tags, 33–35
 - <code> tags, 37
 - <h> tags, 30–33
 - <i> tags, 33–35
 - <kbd> tag, 37
 - positioning blocks of, 187–194
 - <pre> tag, 38
 - <samp> tags, 37
 - <sup> tag, 36
- site design
 - custom error pages, 175–177
 - lobby pages, 165
 - navigation aids, 163–178
- Site menu (Expression Web), 281
- size attribute (<input> tags), 243
- size attribute (<select> tags), 253
- sliders, 256
- SmartArt (Microsoft), 164
- <source> element
 - <audio> tags and, 272
 - <video> tag and, 268
- spacing attribute (table cells), 236
- spacing between letters, adjusting, 120–123
- tags, 118–120
- special characters, 58–60
- spin boxes, 256
- Split tab (Expression Web), 278
- src attribute (<embed> tag), 272
- src attribute (tag), 143
- <s> tags, 116
- standards mode (of web browsers), 20
- start="n" attribute (tags), 53
- stie design
 - consistent design and templates, 309–310
- <strike> tags, 116
- strikethrough, 116
- style attribute
 - <hr> tags, 61
 - tags, 52
 - tags, 52
- style="background-color: color" attribute (<body> tag), 65
- style="background-image: url(image)" attribute (<body> tag), 69
- style="color: color" attribute (<body> tag), 66

- style rules
 - constructing, 89–92
 - defined, 88–89
 - defining multiple, for one style, 90
 - multiple tags, applying to, 90
 - overriding, 100
- Styles pane (Expression Web), 286, 296
- styles (<p> tags)
 - font-style attribute, 114
 - font-weight attribute, 114
 - text-decoration attribute, 116
- <style> tag (header)
 - classes, defining in, 95
 - IDs, defining in, 96
 - style sheets and, 88–89
- Submit button for forms, 244
- subscript format, 35–36
- superscript format, 35–36
- <sup> tag, 36

T

- tables, 199–222
 - accessibility concerns with, 316–317
 - borders, 224–232
 - cells, configuring, 235–239
 - columns, specifying width of, 209–213
 - creating, 200–204
 - foregrounds, applying, 232–235
 - formatting, 223–240
 - headers for, 202
 - merging cells, 213–216
 - page layout with, 216–220
 - size of, specifying, 204–209
 - spanning cells, 213–216
- <table> tags, 200
 - background image for, setting, 233
 - border attribute, 224–226
 - border-color attribute, 226
 - border-style attribute, 226
 - border-width attribute, 226
 - cellpadding attribute, 237
 - cellspacing attribute, 237
 - frame attribute, 225
 - height attribute, 205
 - rules attribute, 226
 - <td> tags, 200
 - <th> tags, 202
 - <tr> tags, 200

<table> tags, *continued*

- width attribute, 204
- tags (HTML), 4–6
 - attributes of, 5
- target attribute (<a> tag), 74
- target windows, 74–75
- taskbar, creating shortcuts in, 8
- <td> tags, 200
 - align attribute, 238
 - border-color attribute, 226
 - border-style attribute, 226
 - border-width attribute, 226
 - cellpadding attribute, 237
 - colspan attribute, 213
 - rowspan attribute, 213
 - valign attribute, 238
 - width attribute, 209
- templates, creating, 21–23
- testing
 - cross-browser compatibility, 16
 - previewing in different browsers, 13
 - tables, 210
- text-align attribute (horizontal rules), 92
- text-align attribute (<p> tags), 134–136
- <textarea> tags, 244
- text boxes, 243
 - clearing, 244
 - default text, adding, 245–246
 - e-mail/URL specific, 244
 - placeholder text, 245–246
 - security concerns with, 249
- text-decoration attributes (<p> tags), 116
- text, formatting, 29–48. *See also* fonts; *See also* CSS (cascading style sheets); *See also* <p> tags
 - block quotes, 42–44
 - bold/italics, 33–35
 - CSS and, 105–124
 - headings, 30–33
 - <h> tags, 30–33
 - monospace/preformatted text, 36–42
 - subscript, 35–36
 - superscript, 35–36
 - superscript/subscript tags, 35–36
 - tags vs. CSS, 105
 - text-decoration attributes, 116
 - web browser settings and, 44–47
- text-indent attribute (<p> tags), 126
- Theora (video encoding format), 263
- <th> tags, 202
 - Internet Explorer and, 209
- title attribute (<a> tags), 77

- <title> tag, 26–28
- top attribute (borders), 131
- trademark symbol (entity), 59
- <tr> tags, 200
 - background colors, applying, 233
 - borders and, 229
- <tt> tag, 37
- tumbnails, 153–155
- type attribute (<input> tags), 243
 - checkbox, 249
 - email value of, 244
 - radio, 250
 - URL value of, 244

U

- tags, 50
- uMenu (drop menu generator), 164
- underline (text-decoration attribute), 116
- URL redirection, 173–175
- usability, 307–312
 - major components of, 307–308
 - page content and, 310
 - planning for, 308–309
 - site organization, planning, 309
 - testing, 311–312
 - visual consistency, 309–310
- Usability.gov, 307
- user behavior and site design, 308
- <u> tags, 116

V

- valign attribute (<td> tags), 238
- value attribute (<input> tags), 249
- value attribute (<option> tags), 253
- value="n" attribute (tags), 53
- video
 - accessibility and, 267
 - embedding, 266–271
 - encoding, 264–266
 - VLC Media Player, encoding with, 265–266
- <video> tags, 266–268
 - attribute list for, 267
 - multiple sources for, 268
 - <source> element and, 268
- Vincent Flanders's Web Pages That Suck, 307
- visited (hyperlink pseudo-class), 98

Visual Studio (web development application), 275
 VLC Media Player, 264
 audio files, encoding with, 271
 video files, encoding with, 265–266
 Vorbis (audio encoding format), 264
 VP8 (video encoding format), 263

W

W3C
 technologies and accessibility, 319
 Web Accessibility Initiative, 313
 web browsers, 13
 accessibility, interim solutions for, 318–319
 audio content, playing in older, 272
 audio support in, 272
 compatibility and older browsers, 13
 email hyperlinks, support for, 76
 extended names (of colors) and, 65
 horizontal rules, support for attributes of, 92
 plug-ins for multimedia content and, 263
 semantic tags, implementation of, 30
 semantic tags vs. divisions, 180
 standards vs. quirks mode, 20
 superscript/subscript formats and, 35
 support for <audio> and <video> tags in, 262
 table behavior of, testing, 210
 Web Content Accessibility Guidelines (WCAG)
 2.0, 313
 WebM (multimedia container format), 263
 webpage files. *See* HTML documents
 web-safe colors, 65
 web sites
 defined, 281
 moving, relative paths and, 73
 width attribute (<table> tags), 204
 width attribute (<td> tag), 209
 Windows 8
 desktop shortcuts, creating, 10
 Internet Explorer and, 13
 monitors, support for multiple, 13
 taskbar shortcuts, creating, 8
 Windows Explorer (Windows 8)
 vs. File Explorer, 10
 Windows Media Player (Microsoft), 263
 word spacing, 121

X

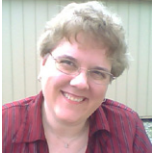
XHTML, 6

 tag, 24
 <!DOCTYPE> tag and, 20
 <hr> tag and, 60–64
 <html>, <head>, and <body> tags, 21
 <meta> tag and, 27
 <p> tag and, 23
 target attribute in <a> tags and, 74
 XML (Extensible Markup Language), 6
 closing tags, requirement for, 6

Y

¥ (yen), 59

About the Author



FAITHE WEMPEN, M.A., is a Microsoft Office Master Instructor, an A+ certified PC technician, and the author of over 120 books on computer hardware and software. She is an adjunct instructor of Computer Technology at Indiana University/Purdue University at Indianapolis (IUPUI), and in her spare time owns and operates a small bed and breakfast in rural Indiana.