

Microsoft®



# Step by Step

# HTML5

Build *exactly* the skills you need.  
Learn at the pace *you* want.

*Faihe Wempen*



Includes  
**practice files** and  
**online edition**  
of this book

See back

# HTML5

Step by Step

Faithe Wempen

Copyright © 2011 Faithe Wempen

Complying with all applicable copyright laws is the responsibility of the user. All rights reserved. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without express written permission of Microsoft Press, Inc.

Printed and bound in Canada.

2 3 4 5 6 7 8 9 10 11 TG 6 5 4 3 2 1

Microsoft Press titles may be purchased for educational, business or sales promotional use. Online editions are also available for most titles (<http://my.safaribooksonline.com>). For more information, contact our corporate/institutional sales department: (800) 998-9938 or send comments to [mispinput@microsoft.com](mailto:mispinput@microsoft.com).

Microsoft, Microsoft Press, ActiveX, Excel, FrontPage, Internet Explorer, PowerPoint, SharePoint, Webdings, Windows, and Windows 7 are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other product and company names mentioned herein may be the trademarks of their respective owners.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail 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 author, Microsoft Corporation, nor their respective resellers or distributors, will be held liable for any damages caused or alleged to be caused either directly or indirectly by such information.

**Acquisitions and Development Editors:** Russell Jones and Kim Spilker

**Production Editor:** Kristen Borg

**Production Services:** Octal Publishing, Inc.

**Technical Reviewer:** Joydip Kanjilal

**Indexing:** Lucie Haskins

**Cover:** Karen Montgomery

**Compositor:** Octal Publishing, Inc.

**Illustrator:** Robert Romano

978-0-735-64526-4

*To Margaret*



# Contents

Acknowledgments .....	xiii
Introduction .....	xv
What Is HTML? .....	xix
Understanding HTML Tags .....	xix
Understanding Cascading Style Sheets .....	xxi
Why Learn HTML in Notepad? .....	xxi
Choosing an HTML Version .....	xxii
Why Code in HTML5? .....	xxiii
Minimum System Requirements .....	xxiv
Using the Practice Files .....	xxv
Getting Help .....	xxviii
Conventions and Features in This Book .....	xxix
What Next? .....	xxx

## Part 1 **Getting Started with HTML**

<b>1</b>	<b>Editing and Viewing HTML Files</b>	<b>3</b>
	Opening a Web Page in Notepad .....	3
	Adding the Data File Location to the Favorites List .....	5
	Opening a File from Windows Explorer .....	7
	Previewing a Web Page in a Web Browser .....	7
	Making, Saving, and Viewing Changes .....	10
	Key Points .....	11

---

**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](https://microsoft.com/learning/booksurvey)

<b>2</b>	<b>Setting Up the Document Structure</b>	<b>13</b>
	Specifying the Document Type . . . . .	14
	Creating the HTML, Head, and Body Sections . . . . .	15
	Creating Paragraphs and Line Breaks . . . . .	17
	Specifying a Page Title and Metatags . . . . .	19
	Publishing a File to a Server. . . . .	22
	Key Points . . . . .	23
<b>3</b>	<b>Formatting Text by Using Tags</b>	<b>25</b>
	Creating Headings. . . . .	26
	Applying Bold and Italic Formatting . . . . .	29
	Applying Superscript and Subscript Formatting . . . . .	31
	Using Monospace and Preformatted Text. . . . .	32
	Formatting a Block Quotation. . . . .	37
	Configuring View Settings in Internet Explorer . . . . .	40
	Key Points . . . . .	45
<b>4</b>	<b>Using Lists and Backgrounds</b>	<b>47</b>
	Creating Bulleted and Numbered Lists . . . . .	48
	Nesting Lists . . . . .	49
	Changing the Bullet or Number Character . . . . .	50
	Specifying the Start of a Numbered List . . . . .	51
	Creating Definition Lists . . . . .	54
	Inserting Special Characters. . . . .	55
	Inserting Horizontal Lines . . . . .	57
	Choosing Background and Foreground Colors . . . . .	60
	Specifying Colors. . . . .	60
	Applying a Background Color . . . . .	61
	Applying a Foreground Color . . . . .	61
	Specifying a Background Image File . . . . .	62
	Key Points . . . . .	65
<b>5</b>	<b>Creating Hyperlinks and Anchors</b>	<b>67</b>
	Hyperlinking to a Web Page . . . . .	68
	Using Partial Paths and Filenames . . . . .	68
	Using Relative and Absolute Paths . . . . .	69
	Setting a Target Window . . . . .	70

Hyperlinking to an E-Mail Address . . . . .	71
Creating and Hyperlinking to Anchors . . . . .	75
Hyperlinking to Other Content . . . . .	78
Key Points . . . . .	81

## Part 2 **Style Sheets and Graphics**

### **6 Introduction to Style Sheets 85**

Understanding Styles . . . . .	86
Constructing Style Rules . . . . .	87
Creating Styles for Nested Tags . . . . .	91
Creating Classes and IDs for Applying Styles . . . . .	93
Applying Styles to Hyperlinks . . . . .	96
Creating and Linking to External Style Sheets . . . . .	98
Key Points . . . . .	100

### **7 Formatting Text by Using Style Sheets 103**

Specifying a Font Family . . . . .	104
Specifying a Font Size and Color . . . . .	107
Applying Bold and Italics . . . . .	111
Applying Strikethrough and Underlining . . . . .	114
Creating Inline Spans . . . . .	117
Adjusting Spacing Between Letters . . . . .	119
Key Points . . . . .	123

### **8 Formatting Paragraphs by Using Style Sheets 125**

Indenting Paragraphs . . . . .	126
Applying a Border to a Paragraph . . . . .	130
Specifying a Border Style . . . . .	130
Setting Border Padding . . . . .	131
Specifying Border Width and Color . . . . .	131
Formatting Border Sides Individually . . . . .	132
Setting All Border Attributes at Once . . . . .	133
Specifying the Horizontal Alignment of a Paragraph . . . . .	135
Specifying Vertical Space within a Paragraph . . . . .	137
Key Points . . . . .	139



## 9 Displaying Graphics 141

Selecting a Graphics Format . . . . .	142
Preparing Graphics for Web Use . . . . .	143
Inserting Graphics . . . . .	144
Arranging Elements on the Page . . . . .	147
Controlling Image Size and Padding . . . . .	149
Hyperlinking from Graphics . . . . .	153
Using Thumbnail Graphics . . . . .	155
Including Alternate Text for Graphics . . . . .	158
Adding Figure Captions . . . . .	158
Key Points . . . . .	161

## Part 3 Page Layout and Navigation

### 10 Creating Navigational Aids 165

Planning Your Site's Organization . . . . .	166
Creating a Text-Based Navigation Bar . . . . .	167
Creating a Graphical Navigation Bar . . . . .	171
Creating an Image Map . . . . .	174
Redirecting to Another URL . . . . .	180
Key Points . . . . .	183

### 11 Creating Division-Based Layouts 185

Understanding HTML5 Semantic Tags . . . . .	186
Beginning to Think in Divisions . . . . .	187
Creating Divisions . . . . .	188
Creating an HTML5 Semantic Layout . . . . .	190
Positioning Divisions . . . . .	192
Floating a Division to the Right or Left . . . . .	192
Positioning a Division on the Page . . . . .	193
Formatting Divisions . . . . .	197
Key Points . . . . .	203

### 12 Creating Tables 205

Creating a Simple Table . . . . .	207
Specifying the Size of a Table . . . . .	211
Specifying the Width of a Column . . . . .	216

Merging Table Cells . . . . .	220
Using Tables for Page Layout . . . . .	224
Key Points . . . . .	229
<b>13 Formatting Tables</b>	<b>231</b>
Applying Table Borders . . . . .	232
Applying Borders by Using Attributes . . . . .	233
Applying Borders by Using Styles . . . . .	235
Applying Background and Foreground Fills . . . . .	241
Changing Cell Padding, Spacing, and Alignment . . . . .	245
Setting Cell Padding . . . . .	246
Setting Cell Spacing . . . . .	246
Setting Horizontal and Vertical Alignment . . . . .	247
Key Points . . . . .	249
<b>14 Creating User Forms</b>	<b>251</b>
Creating a Basic Form . . . . .	252
Creating a Text Box . . . . .	253
Special Field Types for E-Mail and Web Addresses . . . . .	254
Creating a Text Area . . . . .	255
Creating a Submit or Clear Button . . . . .	255
Adding Default or Placeholder Text . . . . .	256
Creating Check Boxes and Option Buttons . . . . .	259
Creating Lists . . . . .	262
Additional Input Types in HTML5 . . . . .	267
Understanding CGI and Other Advanced Tools . . . . .	268
Key Points . . . . .	269
<b>15 Incorporating Sound and Video</b>	<b>271</b>
What's New with Audio and Video in HTML5? . . . . .	272
HTML Multimedia Basics . . . . .	273
Multimedia Formats and Containers . . . . .	273
Codecs: Decoding the Video and Audio . . . . .	274
Which Format to Choose? . . . . .	275
File Size and Quality . . . . .	275
Encoding Video . . . . .	276

Embedding Video Clips . . . . .	277
Introducing the <video> Tag . . . . .	277
The <embed> Tag: Your Fallback Plan . . . . .	279
Placing a Video Clip on a Web Page . . . . .	279
Incorporating Audio on a Web Page . . . . .	282
Playing Audio with the <audio> Tag . . . . .	282
Playing Audio in Older Browsers . . . . .	283
Placing an Audio Clip on a Web Page . . . . .	283
Key Points . . . . .	285

**16 Including JavaScript and External Content 287**

Introducing the Canvas. . . . .	287
JavaScript, Briefly . . . . .	288
Including JavaScript on Your Web Page. . . . .	289
Your First JavaScript Web Page . . . . .	289
JavaScript Events and jQuery. . . . .	292
Obtaining jQuery. . . . .	293
Getting Ready for jQuery . . . . .	294
Selecting Elements with jQuery. . . . .	295
Calling Functions with JavaScript . . . . .	297
Responding to Events with jQuery and JavaScript. . . . .	298
Using the HTML5 <canvas> Tag . . . . .	303
Including External Content in Web Pages. . . . .	310
Key Points . . . . .	311

**Part 4 Other Ways to Create HTML Code**

**17 HTML and Microsoft Expression Web 315**

Exploring the Expression Web Interface . . . . .	316
Creating Web Sites and Web Pages . . . . .	320
Create a Page by Using a CSS Template . . . . .	325
Insert Graphics . . . . .	328
Import an Images Folder . . . . .	328
Place Images on a Page . . . . .	330
Add a Background Image to a Division . . . . .	331
Formatting Text . . . . .	333
Formatting a Division . . . . .	338

Inserting Hyperlinks .....	340
Key Points .....	343

## Part 5 **Appendixes**

### **A Designing for Usability** **347**

Understanding Usability .....	347
Planning for Usability .....	348
Sketching the Site Organization .....	349
Designing a Consistent Page Template .....	349
Designing the Content of Individual Pages .....	350
Performing Usability Testing .....	351

### **B Designing for Accessibility** **353**

Guideline 1: Provide Equivalent Alternatives to Auditory and Visual Content .....	354
Guideline 2: Don't Rely on Color Alone .....	354
Guideline 3: Use Markup and Style Sheets, and Do So Properly .....	355
Guideline 4: Clarify Natural Language Usage .....	356
Guideline 5: Create Tables that Transform Gracefully .....	356
Guideline 6: Ensure that Pages Featuring New Technologies Transform Gracefully ..	357
Guideline 7: Ensure User Control of Time-Sensitive Content Changes .....	357
Guideline 8: Ensure Direct Accessibility of Embedded User Interfaces .....	358
Guideline 9: Design for Device Independence .....	358
Guideline 10: Use Interim Solutions .....	359
Guideline 11: Use W3C Technologies and Guidelines .....	359
Guideline 12: Provide Context and Orientation Information .....	360
Guideline 13: Provide Clear Navigation Mechanisms .....	360
Guideline 14: Ensure that Documents are Clear and Simple .....	361

### **C Tags Added and Removed in HTML5** **363**

Tags Added in HTML5 .....	363
Tags Removed in HTML5 .....	364

Glossary .....	365
----------------	-----

Index .....	369
-------------	-----

About the Author .....	385
------------------------	-----



# Acknowledgments

Thank you to the wonderful editorial staff at O'Reilly Media for guiding this book smoothly through the editorial and production process. This is my first book for O'Reilly, and I certainly hope that it won't be the last. It was a pleasure working with you all.



# Introduction

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

Don't let the phrase "markup language" intimidate you. A markup language annotates or "marks up" plain text, letting a browser know how to format that text so it looks good on a Web page. It's easy to get started—in fact, you can create a simple Web page in just a few minutes. While full-featured What You See Is What You Get (WYSIWYG) tools exist that can help speed up the process of writing Web pages, all you really need is an ordinary text-editing program such as Microsoft Notepad. You don't need special software or extensive training.

In this introduction, you'll learn some basics about HTML. You'll find out how they turn plain text into attractive formatting, how they incorporate graphics and hyperlinks, and how anyone can create Web content in virtually any program that edits text. This introduction explains what cascading style sheets (CSS) are, and how they make formatting consistent across large Web sites. You'll also discover the differences between HTML4, XHTML, and HTML5, so you can make the important decision about which version of HTML you want your code to conform to. Finally, you'll learn about the conventions used in this book for pointing out special helps like notes, tips, cautions, and references to the data files.

## How to Access Your Online Edition Hosted by Safari


The voucher bound in to the back of this book gives you access to an online edition of the book. (You can also download the online edition of the book to your own computer; see the next section.)

To access your online edition, do the following:

1. Locate your voucher inside the back cover, and scratch off the metallic foil to reveal your access code.
2. Go to <http://microsoftpress.oreilly.com/safarienabled>.
3. Enter your 24-character access code in the Coupon Code field under Step 1:



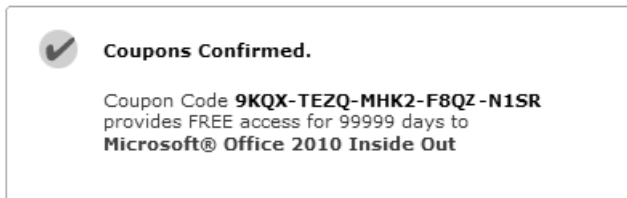
## Step 1

Coupon Code:  

(Please note that the access code in this image is for illustration purposes only.)

4. Click the CONFIRM COUPON button.

A message will appear to let you know that the code was entered correctly. If the code was not entered correctly, you will be prompted to re-enter the code.



5. In this step, you'll be asked whether you're a new or existing user of Safari Books Online. Proceed either with Step 5A or Step 5B.
  - 5A. If you already have a Safari account, click the EXISTING USER – SIGN IN button under Step 2.

## Step 2

OR

- 5B. If you are a new user, click the NEW USER – FREE ACCOUNT button under Step 2.
  - You'll be taken to the "Register a New Account" page.
  - This will require filling out a registration form and accepting an End User Agreement.
  - When complete, click the CONTINUE button.
6. On the Coupon Confirmation page, click the My Safari button.
7. On the My Safari page, look at the Bookshelf area and click the title of the book you want to access.


### 0-Slot Bookshelf



You are using 0 out of 0 available Bookshelf slots.

You have full access to the content of the titles you place in your Bookshelf Folder. Any title placed in your Bookshelf Folder must remain on your Bookshelf for 30 days.

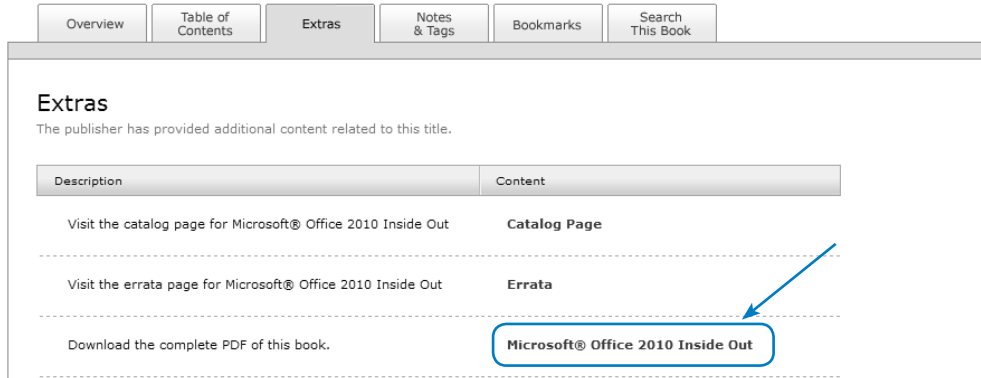
1-1 of 1 Items < Prev | 1 | Next >

	Title	Published	Added To Folder		
1.	 <b>Microsoft® Office 2010 Inside Out</b> Ed Bott; Carl Siechert Slots: 0 <input type="button" value="ADD TO FOLDER"/>	29-SEP-2010	27-OCT-2010	0 Notes 0 Bookmarks	May Remove On 10-AUG- 2284

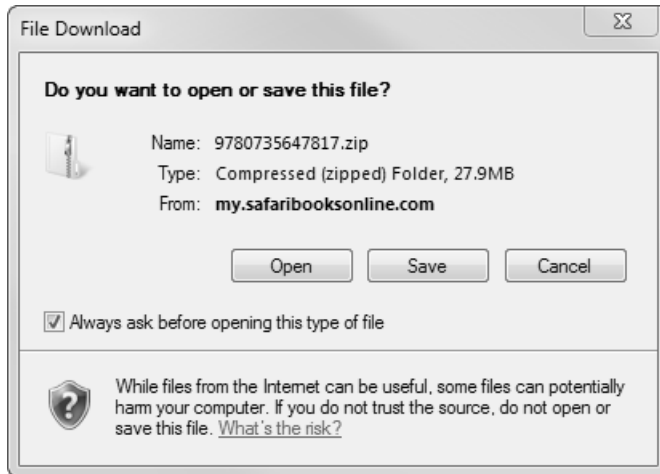
## How to Download the Online Edition to Your Computer

In addition to reading the online edition of this book, you can also download it to your computer. First, follow the steps in the preceding section. After Step 7, do the following:

1. On the page that appears after Step 7 in the previous section, click the Extras tab.
2. Find "Download the complete PDF of this book," and click the book title:



A new browser window or tab will open, followed by the File Download dialog box:



3. Click Save.
4. Choose Desktop and click Save.
5. Locate the .zip file on your desktop. Right-click the file, click Extract All, and then follow the instructions.

**Note** If you have a problem with your voucher or access code, please contact [mspinput@microsoft.com](mailto:mspinput@microsoft.com). or call 800-889-8969, where you'll reach O'Reilly Media, distributor of Microsoft Press books.

# What Is HTML?

In simple terms, a *Web page* (or *HTML document*) is a plain text file that has been encoded using *Hypertext Markup Language (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 Web pages 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.

**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 Web pages resides in files. Each time your Web browser opens a Web page, it processes the HTML code within the file.

## Understanding HTML Tags

The code within an HTML file consists of text surrounded by *tags*. These tags indicate where the 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 be italicized, like this:

*Everything* is on sale.

In HTML, there's no *Italics* button to click, like there is in a word-processing program. Therefore, you have to "tag" the word that you want to be italicized. The code to turn on italics is `<i>`, and the code to turn italics off is `</i>`. Your HTML code would look something like this:

```
<i>Everything</i> is on sale.
```

That's an example of a *two-sided tag*, which encloses text between opening and closing tags, in this case `<i>` and `</i>`. Note the forward slash in the closing tag (`</i>`). 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 olden days of the Internet, nearly everyone connected to it by using a dial-up modem, 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 Web pages 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 one a visitor to a site 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 `<img>` 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, "Displaying Graphics.")

The `<img>` tag is different in several ways from the `<i>` 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 `<img>` tag, you have to specify a source, abbreviated *src*. Here's an example:

```

```

This `<img>` 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.

With HTML, you can also create *hyperlinks* from one page to another. When a visitor to a Web site 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 file *index.htm* when clicked, the coding would look like this:

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

There's a lot more to HTML, of course, but that's basically how it works. Plain text is marked up with tags that indicate where elements such as formatting, hyperlinks, and graphics should be applied, and a Web browser interprets those tags and displays the page in its formatted state. The trick, of course, is to know which tags to use, and where they're appropriate, and what attributes they need. And that's the subject of this book.

## Understanding Cascading Style Sheets

Web designers who worked with early versions of HTML to create large Web sites were often frustrated by the amount of repetition involved in their jobs. Suppose a Web site has 200 pages, all using the same basic layout and design. To make a design change to the entire site, a designer would have had to go in and manually edit each of those 200 pages.

Later versions of HTML have gotten around this by supporting *cascading style sheets*. Based on the same principle as style templates in a word-processing or page-layout program, Web designers use cascading style sheets to specify the formatting for a particular tag type—usually in a separate style sheet document—and then apply that style sheet to multiple pages. Need to make a change to the style? Simply make it in the style sheet, and the change is applied automatically to all pages.

Although you can still format documents by using older methods—and you'll learn how to do a little of that in this book—most Web designers rely almost exclusively on cascading style sheets for formatting these days, and XHTML all but demands that you do so. It might seem intimidating at first, but if you are creating a multi-page site, the extra trouble involved in setting up a cascading style sheet will pay for itself many times over.

## Why Learn HTML in Notepad?

This book teaches beginner-level HTML coding, but it teaches it in a rather fundamental way: by creating plain text files in Notepad. There are so many good Web site 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 Web site 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.

## Choosing an HTML Version

Different versions of HTML use different tags for some types of content, although they more similar than different overall, especially at the beginner level covered in this book. Here's a quick comparison of the HTML versions you may encounter:

- *HTML4* A very stable, universally accepted code set, which is also fairly forgiving of small coding errors. Using HTML4 codes is desirable when compatibility with all browsers is important.
- *XHTML* A strict, standards-based implementation of HTML4 created with XML (eXtensible Markup Language). XHTML coding uses the same codes as HTML4, so it is compatible with the same browsers as HTML4. (See the sidebar about XML on the next page for more information.)
- *HTML5* A revised code set that builds upon HTML4 to add new capabilities. HTML5 offers many dramatic improvements in the areas of application handling and multimedia, but a lot of those features are beyond the scope of this book. In terms of basic coding, which is what this book teaches, the biggest difference is that there are new specific codes for different types of content that were previously handled with more general codes. For example, HTML5 has `<audio>` and `<video>` tags for inserting multimedia content, whereas HTML4 inserts all types of multimedia content via a generic `<embed>` tag.

Since this is a book about HTML5, it might seem like an obvious decision to do your coding using HTML5 tags, but it is not quite as simple as that in real-world situations.

A good Web browser should ideally support every tag and every version of HTML it can, because the various HTML version differences should be completely invisible to the Web site visitor. However, HTML5 is so new that not all browsers have caught up to it yet, and people who use older computers may not have the latest version of a browser even if an HTML5 compatible version is available.

**Tip** Here's a site that lists what HTML5 features are supported by each version of each of the popular Web browsers: <http://caniuse.com>.

The code you will create as you work through the exercises in this book is based on HTML5, but I will also show you some workarounds in situations where HTML5 codes might cause problems in some browsers. You'll learn both ways of creating a certain effect, so that you can make the call of which codes to use in your real-life work as the situations arise.

### What are XML and 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. The W3C did just that: they re-created the entire HTML language in XML, and called it Extensible HTML (XHTML). Version 1.0 was released in 2001; the current version is XHTML 2.0, released in 2004.

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.

At one point, it was thought that XHTML would eventually replace HTML4 as its successor, but due to interoperability problems, that has not happened; instead HTML5 is poised to succeed HTML4. This book doesn't explicitly cover XHTML, but most of what you will learn can be applied to XHTML coding.

## Why Code in HTML5?

The short answer is: you should code in HTML5 because it's an investment in the future. Within a few years, it will be the standard on which nearly all Web sites are based.



A slightly longer answer is because it enables cleaner, easier-to-write code. Web page technology has grown by leaps and bounds, mostly due to the increase of the average person's Internet connection speed, but also because users, designers, and programmers increasingly demand more functionality from their Web pages, such as more precise control of fonts and layout, better rendering on devices that vary wildly in size from mobile phones to huge desktop monitors, better images, more interactivity, video, audio, animations, and better support for various image and file formats. Because most people have fast connections, they don't have to wait a long time for pages to load that contain large audio and video files, which means more and more sites are including audio and video content.

HTML was not originally designed for the rigors of multimedia content delivery, so more and more high-end professional sites have moved to other languages and technologies that piggyback on HTML to deliver that content, such as JavaScript, Java, and Active Server Pages (ASP).

HTML5 adds some important new tags to make audio, video, and application integration smoother and more reliable. You'll learn about many of these new tags in Chapters 15 and 16, including `<audio>`, `<video>`, and `<canvas>`.

HTML5 removes support for some of the older tags. For example, an old way (pre-HTML4) of specifying a font was the `<font>` tag. Today, most people use cascading style sheets to define fonts, so the `<font>` tag has not been used by many Web designers in a long time anyway. HTML5 formally removes it from the language.

One of the biggest things that HTML5 removes is the ability to create multi-framed Web sites with the `<frame>` and `<frameset>` commands. You can still create Web sites with multiple sections, but they're handled much more capably using tables or divisions. Chapter 11 covers divisions—the newer way, preferred by most professional Web designers. Chapters 12 and 13 cover tables, still an acceptable way, and preferred by many casual Web page designers who are familiar with tables from programs like Word.

## Minimum System Requirements

There are no minimum system requirements for developing HTML; you can do it in any text editing program with any type of computer and any operating system. That's the beauty of HTML! This book uses Notepad as the text editor, but you can use any editor that you like.

For testing your work, you will need an HTML5-compliant Web browser application. The latest versions of Google Chrome and Firefox (both freely available online) will work fine for this, as will Internet Explorer 9 or higher.

# Using the Practice Files

Each exercise in the lessons is preceded by a paragraph or paragraphs that list the files needed for that exercise and explain any file preparation you need to take care of before you start working through the exercise. The practice files are available for download from <http://aka.ms/645264/files>. When you unzip them from the download file, separate folders will be created for each chapter, and separate folders within each of those for each exercise.

The following table lists the practice file folders for each chapter and the subfolders you'll find within them. The practice file folder for each chapter also includes a Solutions subfolder containing finished versions of the practice files used in that chapter.

<b>Chapter</b>	<b>Folder</b>	<b>Subfolder</b>
Chapter 1: HTML and XHTML Basics	01Editing	no subfolders
Chapter 2: Setting Up the Document Structure	02Structure	CreatingParagraphs PublishingFiles SpecifyingKeywords SpecifyingTitle
Chapter 3: Formatting Text by Using Tags	03Format	ApplyingBold ApplyingSuperscript ConfiguringSettings CreatingHeadings FormattingQuotes UsingMonospace
Chapter 4: Using Lists and Backgrounds	04Lists	ChoosingColors CreatingGlossary InsertingCharacters InsertingLines NestingLists SpecifyingImages
Chapter 5: Creating Hyperlinks and Anchors	05Links	CreatingAnchors CreatingHyperlinks LinkingEmail LinkingOther

<b>Chapter</b>	<b>Folder</b>	<b>Subfolder</b>
Chapter 6: Introduction to Style Sheets	06Styles	ConstructingRules CreatingClasses CreatingExternal CreatingNested StylingHyperlinks
Chapter 7: Formatting Text by Using Style Sheets	07Text	AdjustingSpacing ApplyingBold ApplyingStrike CreatingSpan SelectingFont SelectingSize
Chapter 8: Formatting Paragraphs Using Style Sheets	08Paragraphs	AddingBorders AdjustingHeight Indenting SettingAlignment
Chapter 9: Displaying Graphics	09Graphics	CaptioningFigures ClearingImages CreatingHyperlinks InsertingImages SizingImages UsingAlt UsingThumbnails
Chapter 10: Creating Navigational Aids	10Navigation	CreatingGraphicBar CreatingImageMap CreatingTextBar Redirecting
Chapter 11: Creating Division-Based Layouts	11Divisions	CreatingDivisions FormattingDivisions PositioningDivisions UsingSemantic
Chapter 12: Creating Tables	12Tables	CreatingTable SettingWidth SpanningCells SpecifyingSize UsingTables

<b>Chapter</b>	<b>Folder</b>	<b>Subfolder</b>
Chapter 13: Formatting Tables	13FmtTables	ApplyingBackground ApplyingBorders ChangingPadding
Chapter 14: Creating User Forms	14Forms	CreatingButtons CreatingForms CreatingLists
Chapter 15: Incorporating Sound and Video	15AudioVideo	
Chapter 16: Including JavaScript and External Content	16Canvas	
Chapter 17: HTML and Microsoft Expression Web	17Expression	ViewingPage

# Getting Help

Every effort has been made to ensure the accuracy of this book. If you do run into problems, please contact the sources listed in the following topics.

## Getting Help with This Book

If your question or issue concerns the content of this book or its practice files, please first consult the book's page, which can be accessed at:




*<http://aka.ms/645264/files>*

This page provides information about known errors and corrections to the book. If you do not find your answer on the errata page, send your question or comment to:

*[mspinput@microsoft.com](mailto:mspinput@microsoft.com)*

# Conventions and Features in This Book

You can save time when you use this book by understanding how the *Step by Step* series shows special instructions, keys to press, buttons to click, and so on.

Convention	Meaning
 SET UP	These words are found at the beginning of paragraphs preceding step-by-step exercises. They point out items you should check or actions you should carry out before beginning an exercise.
Use Open	These words are found within the SET UP paragraphs that precede step-by-step exercises. They draw your attention to practice files that you'll need to use in the exercise.
 CLEAN UP	These words are found at the beginning of paragraphs following step-by-step exercises. They give instructions for closing open files or programs before moving on to another topic.
1.	Numbered steps guide you through hands-on exercises in each topic.
	A round bullet indicates an exercise that has only one step.
Troubleshooting	These paragraphs show you how to fix a common problem that might prevent you from continuing with the exercise.
Tip	These paragraphs provide a helpful hint or shortcut that makes working through a task easier.
Important	These paragraphs point out information that you need to know to complete a procedure.
Note	These paragraphs provide supplementary or related information.
Compatibility	These paragraphs explain alternate coding you can use for greater backward compatibility
<b>Ctrl+C</b>	A plus sign (+) between two key names means that you must hold down the first key while you press the second key. For example, "press <b>Ctrl+C</b> " means "hold down the <b>Ctrl</b> key while you press the <b>c</b> key."
<b>user interface elements</b>	In exercises, the names of program elements such as buttons, commands, and dialog boxes.
<b>user input</b>	Anything you are supposed to type.
<i>glossary terms</i>	Terms explained in the glossary at the end of the book.

## What Next?

To get started, turn the page to Chapter 1 and start reading and working through the exercises. The lessons are designed to be tackled in the order they appear in the book, but feel free to skip around if you just need to fill in some holes in your HTML knowledge.

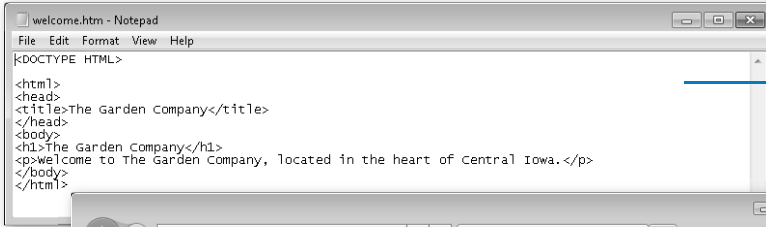
Part 1

# Getting Started with HTML

<b>1</b>	<b>Editing and Viewing HTML Files . . . . .</b>	<b>3</b>
<b>2</b>	<b>Setting Up the Document Structure. . . . .</b>	<b>13</b>
<b>3</b>	<b>Formatting Text by Using Tags . . . . .</b>	<b>25</b>
<b>4</b>	<b>Using Lists and Backgrounds. . . . .</b>	<b>47</b>
<b>5</b>	<b>Creating Hyperlinks and Anchors . . . . .</b>	<b>67</b>



# Chapter at a Glance

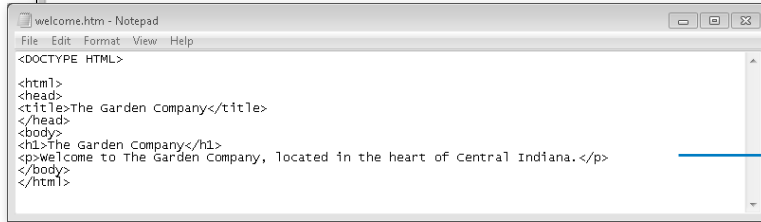


```
welcome.htm - Notepad
File Edit Format View Help
<DOCTYPE HTML>
<html>
<head>
<title>The Garden Company</title>
</head>
<body>
<h1>The Garden Company</h1>
<p>welcome to The Garden Company, located in the heart of Central Iowa.</p>
</body>
</html>
```

Open a Web page in Notepad, **page 2**



Preview a Web page in Internet Explorer, **page 7**



```
welcome.htm - Notepad
File Edit Format View Help
<DOCTYPE HTML>
<html>
<head>
<title>The Garden Company</title>
</head>
<body>
<h1>The Garden Company</h1>
<p>welcome to The Garden Company, located in the heart of Central Indiana.</p>
</body>
</html>
```

Make, save, and view changes, **page 10**



# 1 Editing and Viewing HTML Files

---

## In this chapter, you will learn how to

- ✓ Open a Web page in Notepad.
  - ✓ Preview a Web page in Internet Explorer.
  - ✓ Make, save, and view changes.
- 

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

**See Also** Do you need only a quick refresher on the topics in this chapter? See the Key Points section at the end of this chapter.

**Practice Files** 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 "Using the Practice Files" at the beginning of this book for more information.

## Opening a Web Page in Notepad

Notepad is included with all versions of Windows, and you'll find it in the All Programs (or Programs)/Accessories folder on the Start menu. 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 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 Web pages (which have .htm or .html extensions).

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

In this exercise, you will open a Web page in Notepad and examine its text and tags.



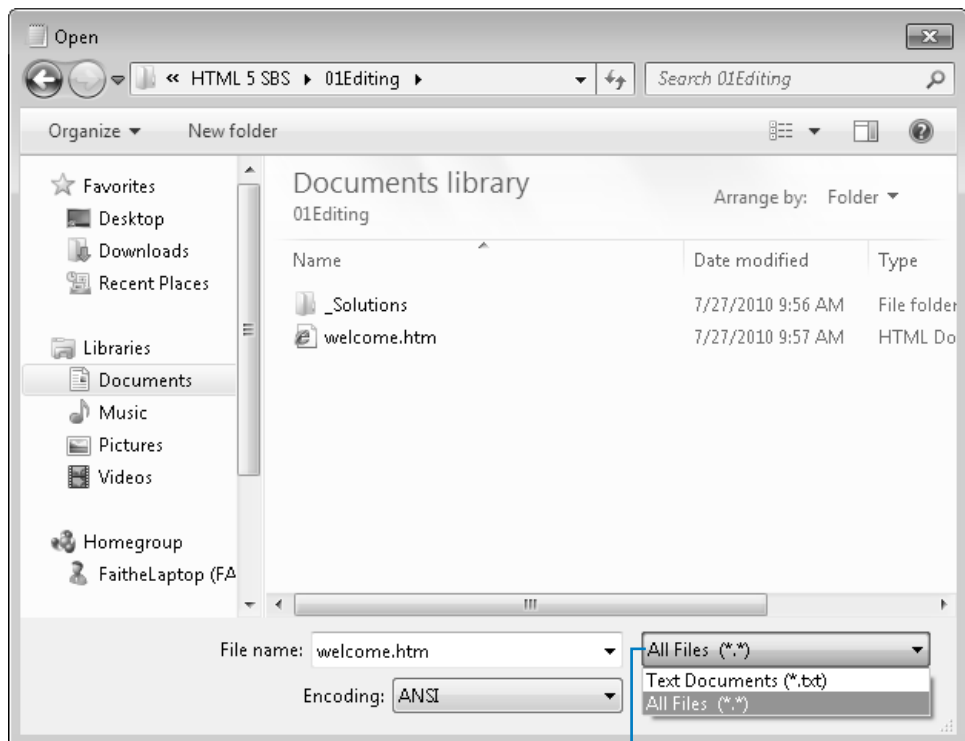
**SET UP** Use the *welcome* file in the practice file folder for this topic. This practice file is located in the Documents\Microsoft Press\HTML5 SBS\01Editing folder.

1. From the **Start** menu, select **All Programs | Accessories | Notepad**.
2. In the untitled Notepad window, select **File | Open**.
3. Navigate to the folder containing the practice files for this chapter.

On the Places bar, click Documents (or My Documents if you are using Windows XP). In the Open dialog box, double-click Microsoft Press, HTML5 SBS, and then 01Editing.

**Note** You won't see any files in the list at this point. The only thing that you should see is just 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** down arrow, and then click **All Files**.

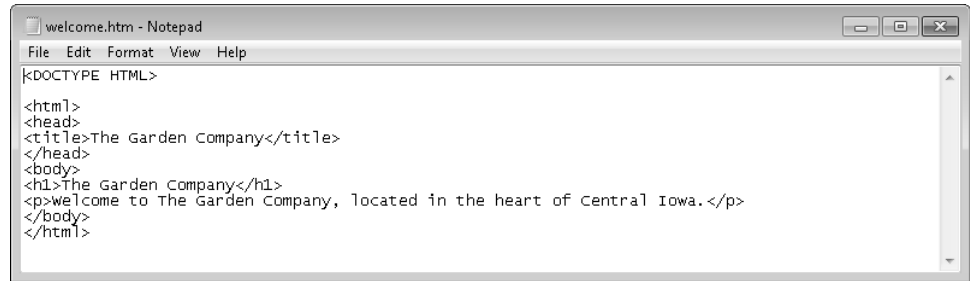


Click here to open Files of Type list

- 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 Computer (or My Computer), and on the Tools menu (press the Alt key for the menu bar if you don't see it), 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>The Garden Company</title>
</head>
<body>
<h1>The Garden Company</h1>
<p>welcome to The Garden Company, located in the heart of Central Iowa.</p>
</body>
</html>

```

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

These tags signify the beginning and end of the HTML code.

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

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

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

These tags signify the beginning and end of a paragraph.

**✖ CLEAN UP** Leave the page open in Notepad for later use.

## Adding the Data File Location to the Favorites List

In the course of 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 (HTML5 SBS), you might want to add that folder to your Favorites bar in the Open dialog box for easy access to the data files.

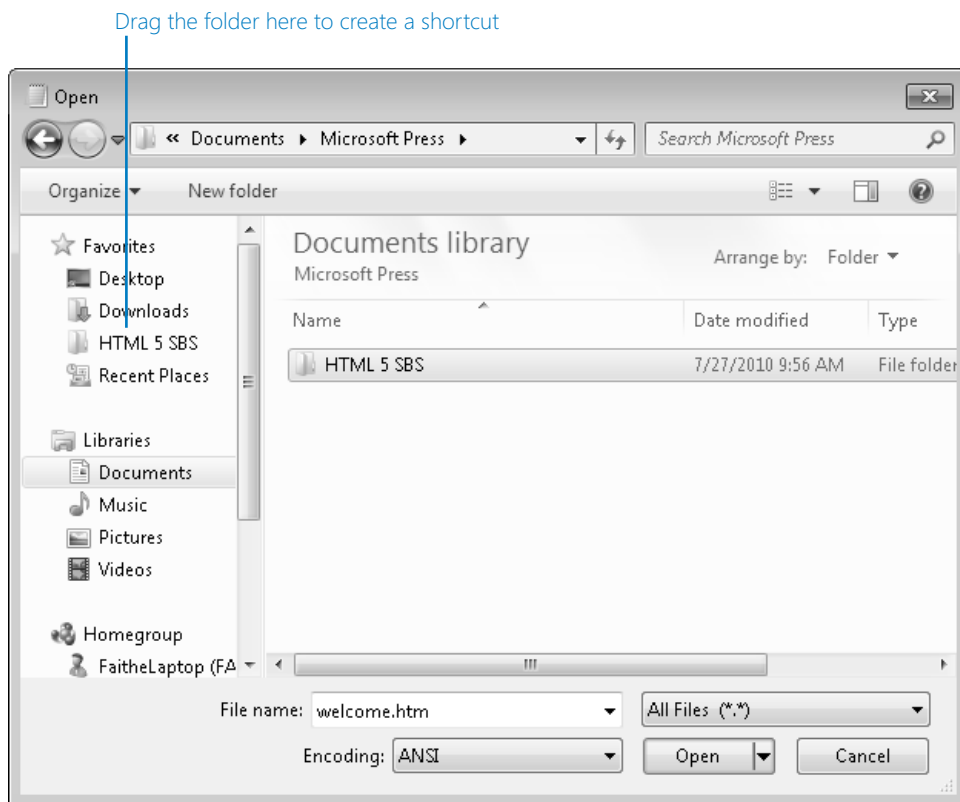
In this exercise, you will add to the Favorites bar a shortcut that brings you directly to the HTML5 SBS folder.

## ➔ SET UP Open Notepad.

1. Select **File | Open**.
2. Navigate to the folder containing the practice files for this chapter.

On the Places bar, click Documents (or My Documents if you are using Windows XP). In the Open dialog box, double-click Microsoft Press. The HTML5 SBS folder appears as an icon.

3. Drag the **HTML5 SBS** icon to the Favorites list on the left side of the window.  
A shortcut for it appears on the Favorites list.

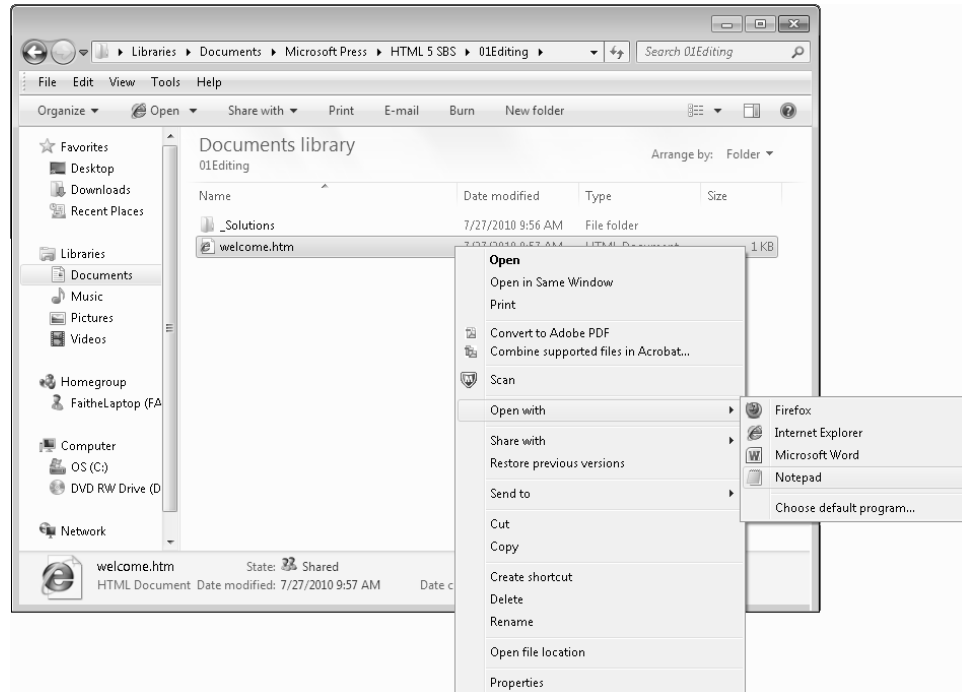


## ✘ CLEAN UP Close the Windows Explorer window

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

## Opening a File from Windows Explorer

A quick way to open most file types in their default applications is to double-click them from any Windows Explorer window. However, the problem with doing that for HTML files is that the default application is your Web browser, not Notepad; thus, instead of the file opening in Notepad, it opens in your Web browser. One way to get around this is to right-click a file in Windows Explorer, choose Open With from the contextual menu, and then click Notepad. This opens Notepad and loads the file.



## Previewing a Web Page 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 affect the finished product. To work around this, most Web page designers keep a browser window open next to Notepad.

You can preview your work in any browser; you do not need to use Internet Explorer 9 (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 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 the most popular browser—the one your target audience is most likely to be using. Other popular browsers include Google Chrome, Firefox, Safari, and Opera.

**Caution** Versions of Internet Explorer prior to version 9 do not support some of the HTML5 features. You will probably want to test your Web pages in an earlier version to make sure that people who use them will be able to view your page. But don't use an earlier 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 could 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 even more.

In this exercise, you will display an HTML file in Internet Explorer. To see the displayed file and the underlying code at the same time, open the practice files from this exercise and the previous exercise in separate windows and arrange them so both are visible.



**SET UP** Use the *welcome* file from the previous exercise, or use the one in the practice file folder for this topic. The practice file is located in the Documents\Microsoft Press\HTML5 SBS\01Editing folder.

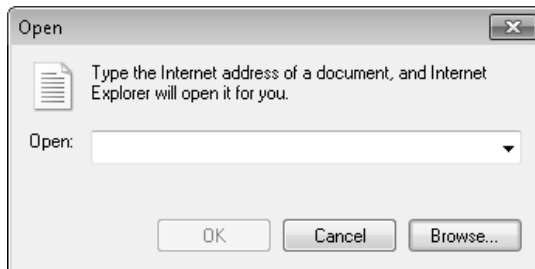
**1.** Select **Start | Internet Explorer**.

**Note** Depending on your system and your default browser, Internet Explorer might not be pinned to the top of your Start menu. If it is not, click **Start | All Programs | Internet Explorer**.

**2.** Select **File | Open**.

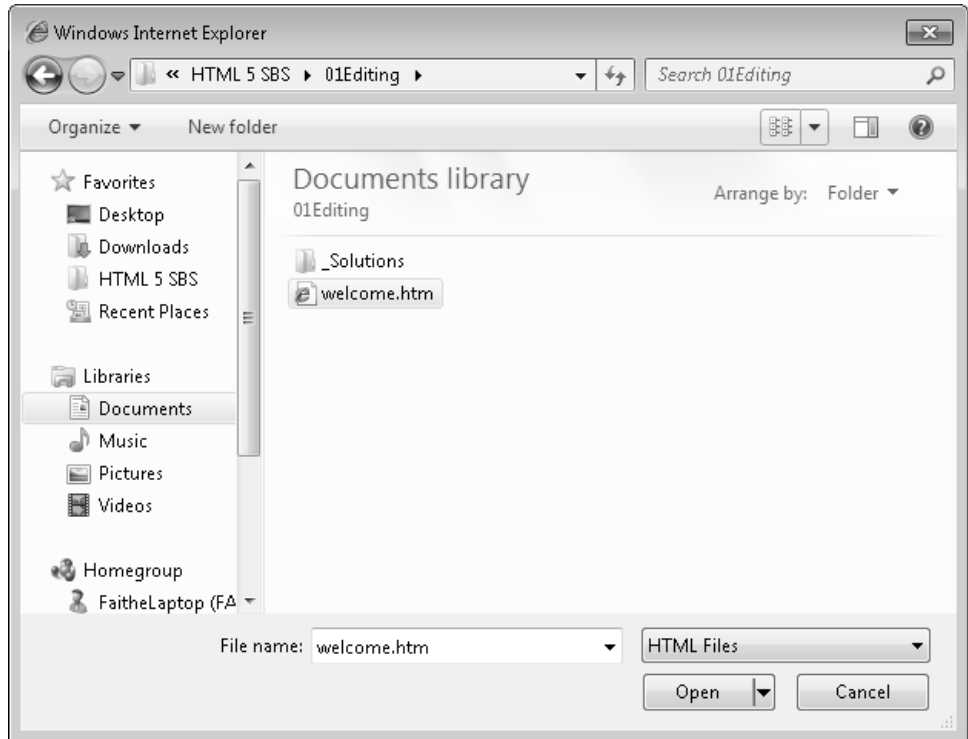
The Open dialog box appears.

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



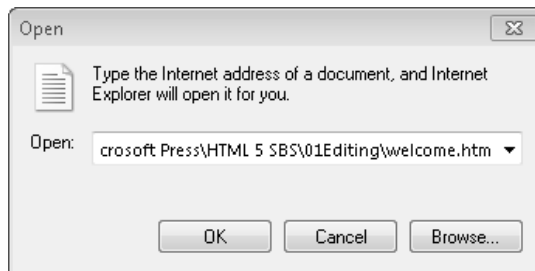
3. Click the **Browse** button, and then browse to Documents\Microsoft Press\HTML5 SBS\01Editing.

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



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

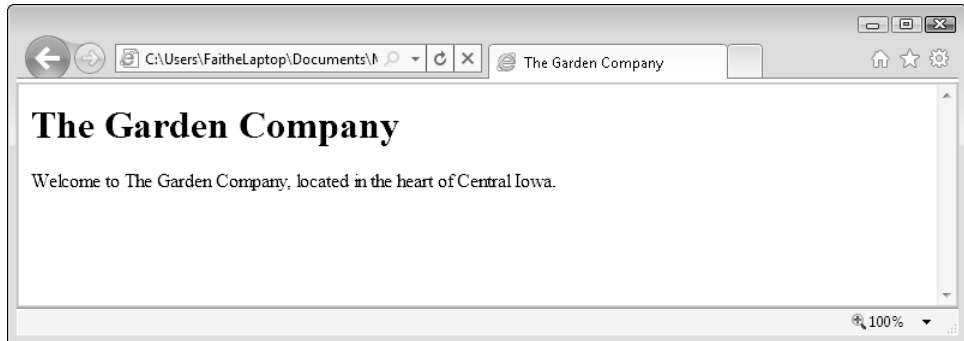
The path to the file appears in the Open dialog box.



5. Click **OK**.



The file opens in Internet Explorer.



### ✘ CLEAN UP Leave Internet Explorer open for the next exercise.

The method you just learned works especially well when you already have your Web browser open, in which case you can skip step 1. An alternate method is to browse to the storage location by using Windows Explorer, and double-click the file. Remember that you can't just double-click a Web page to edit it; you must right-click it. By default, the double-click operation is reserved for opening the page in your Web browser. What was a hardship only a few pages ago is now a convenience!

**Tip** Not all Web browser software displays pages exactly 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, Chrome, and Opera. These are available as free downloads from [www.firefox.com](http://www.firefox.com), [www.google.com/chrome](http://www.google.com/chrome), and [www.opera.com](http://www.opera.com), respectively.

## Making, Saving, and Viewing Changes

After you've made a change to a Web page, you will probably want to preview the result of that change. 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.

In this exercise, you will change "Iowa" to "Indiana" 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. You can use this procedure throughout the rest of the book to preview your work from each exercise.

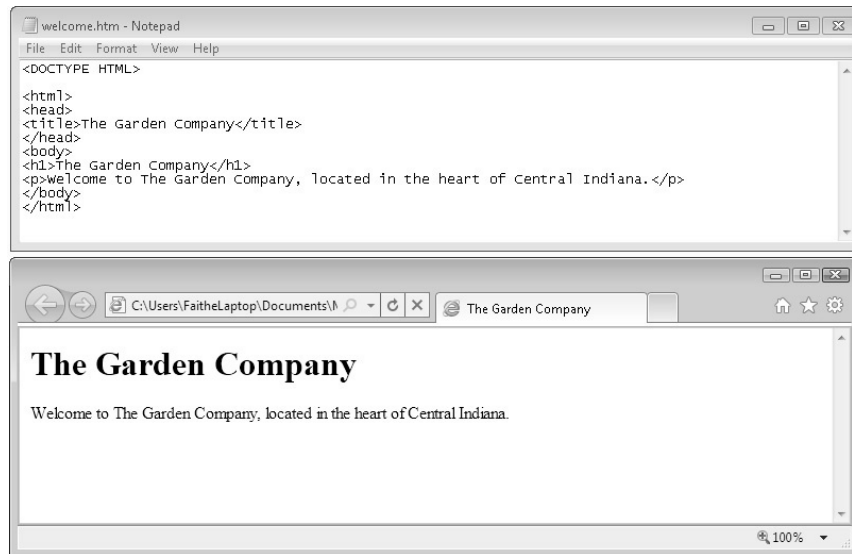
➔ **SET UP** Be sure to have the `welcome` file open in Notepad and in Internet Explorer before beginning this exercise. Use the `welcome` file from the previous exercise, or use the one in the practice file folder for this topic. The practice file is located in the `Documents\Microsoft Press\HTML5 SBS\01Editing` folder.

1. In Notepad, locate the word *Iowa*, and change it to **Indiana**, as shown in bold text in the following code:

```
<p>Welcome to the Garden Company, located in the heart of Central Indiana.</p>
```

2. Save your work (**File | Save**).
3. On the Internet Explorer toolbar, click the **Refresh** button.

Notice that the Web page shown in Internet Explorer now reads “Indiana,” too.



**✘ CLEAN UP** Close the *welcome* file, and then exit Notepad and Internet Explorer.

## Key Points

- Any plain text editor, including Notepad, can be an HTML editor.
- Most Web pages have an .htm or .html extension. You can open them 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 Web page in Notepad is to right-click it in Windows Explorer, select Open With from the contextual menu, and then click Notepad.
- To preview a page in a Web browser, select File | Open from the browser’s menu.
- You can double-click an .htm or .html file in Windows Explorer to open it automatically in your default Web browser.
- To see changes you make in Notepad reflected in your Web browser, save your work in Notepad, and then click Refresh in the browser window.

# Chapter at a Glance

Create the HTML, head, and body sections, **page 15**

```
Untitled - Notepad
File Edit Format View Help
<!DOCTYPE html>
<html>
<head>
</head>
<body>
</body>
</html>
```

Create paragraph and line breaks, **page 17**

Welcome to The Garden Company! We hope you will find our site a useful resource for becoming a better gardener.

Our main store is located at:  
108 Ponting Street  
Macon, IN 46062

Please stop by and browse our extensive inventory from Monday through Saturday, 7:00 a.m. to 8:00 p.m.

Specify a page title, **page 19**

The Garden Company

Welcome to The Garden Company! We hope you will find our site a useful resource for becoming a better gardener.

Our main store is located at:  
108 Ponting Street  
Macon, IN 46062

Please stop by and browse our extensive inventory from Monday through Saturday, 7:00 a.m. to 8:00 p.m.

# 2 Setting Up the Document Structure

---

## In this chapter, you will learn how to

- ✓ Specify the document type.
  - ✓ Create the HTML, Head, and Body sections.
  - ✓ Create paragraphs and line breaks.
  - ✓ Specify a page title and keywords.
  - ✓ Publish a file to a server.
- 

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, or 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 Web pages properly. That means the document must contain certain tags that delineate its major sections and 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 and Body sections. You’ll also learn how to create paragraph and line breaks, specify a page title, enter hidden keywords by which your page can be found in search engines, and publish a test page to a Web server.

**See Also** [Do you need only a quick refresher on the topics in this chapter? See the Key Points section at the end of this chapter.](#)

**Practice Files** 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 “Using the Practice Files” at the beginning of this book for more information.

## Specifying the Document Type

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 assumes that there is something quirky about the page, and 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. In 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.

Earlier versions of HTML used more complex *DOCTYPE* tags. If you're using HTML Version 4.01, the syntax for the tag is:

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

If you're using XHTML, the syntax for the tag is:

```
<!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 of your HTML coding—except the *DOCTYPE* tag—should be placed within the two-sided `<html>` tag. Recall from the Introduction that when a tag is two-sided, it requires a corresponding closing tag that is identical to the opening tag but contains a slash: `</html>`. The tags `<html>` and `</html>` serve as a “wrapper” around all the other tags in the document.

In addition, your document should have two sections: a Head and a Body. The *Head* section is defined by the two-sided tag `<head>`. The Head section contains the *page title*, which is the text that will appear in the title bar of the Web browser and on the Microsoft Windows taskbar button. It also includes information about the document that is not displayed, such as its `<meta>` tags (which you’ll learn about on page 19). You can also include lines of code that run scripts, like Javascript.

The *Body* section is defined by the two-sided tag `<body>`, and it contains all the information that appears 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, “Editing and Viewing HTML Files”). Here’s how the opening `<html>` tag should look in an XHTML document: `<html xmlns="http://www.w3.org/1999/xhtml">`.

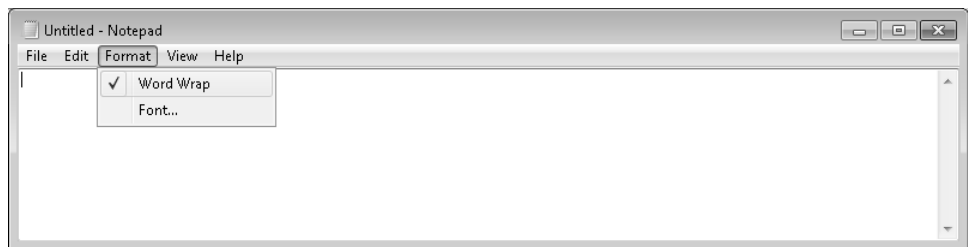
In this exercise, you will create an HTML5 template file that you can reuse later for your own work.



**SET UP** Start Microsoft Notepad before beginning this exercise.

1. In Notepad, open 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.



- In the Notepad window, type the following:

```
<!DOCTYPE html>
```

- Press **Enter**, and then type:

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

- Press **Enter** two or three times to add some blank lines, and then type:

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

- Press **Enter** two or three times to add some blank lines, and then type:

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



```

Untitled - Notepad
File Edit Format View Help
<!DOCTYPE html>
<html>
<head>

</head>
<body>

</body>
</html>

```

- 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, or you can store them anywhere you like.

## ✘ CLEAN UP Close the Notepad window.

You now have a template for creating any HTML documents you like. You can reopen this template file and save it under different names, which will save time re-creating these basic tags.

**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 select 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 appears and a Save As dialog box prompts you to save a copy of it with a new name.

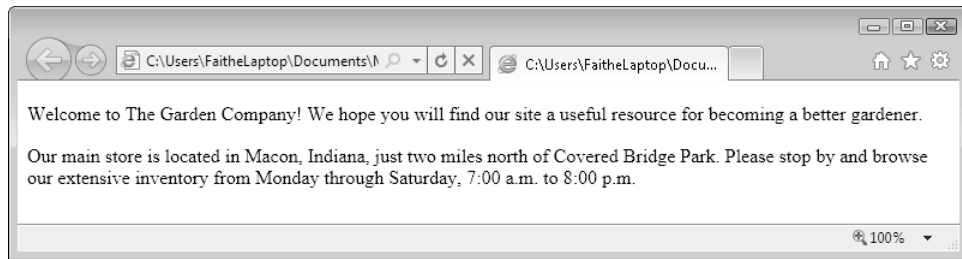
## Creating Paragraphs and Line Breaks

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

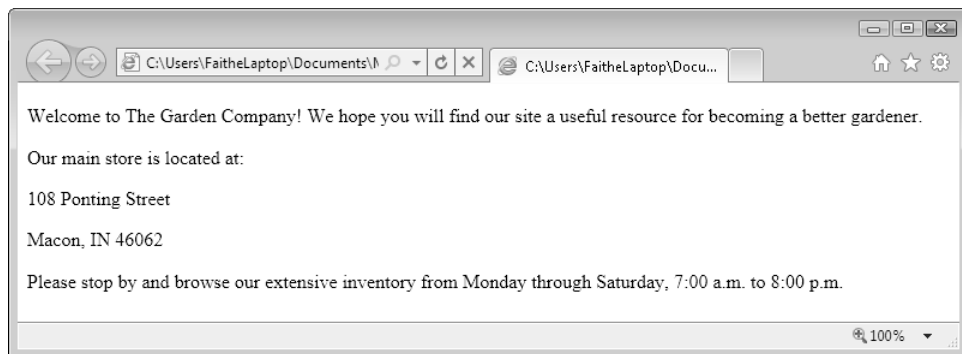
The most basic paragraph type is the body paragraph, indicated by the `<p>` tag. It is a two-sided tag, so the paragraph text is placed between a `<p>` and a `</p>`.

**Note** In HTML, the code will still work even if the `</p>` is omitted; in XHTML, it won't. However, 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 Web page, it inserts vertical white space between paragraphs:



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



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

```
<p>David Jaffe<br>
317-555-8882</p>
```



**Note** In XHTML, the line break tag is `<br />`. The end slash (and the space) 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 this exercise, you will add text to an HTML file template, and then preview it in Microsoft Internet Explorer.



**SET UP** Use the *HTML5.htm* file from the previous exercise or in the practice file folder for this topic. This practice file is located in the Documents\Microsoft Press\HTML5 SBS\02Structure\CreatingParagraphs folder. Open the *HTML5* file in Notepad.

1. Save the *HTML5* file in the Documents\Microsoft Press\HTML5 SBS folder as **index.htm**.

**Note** It is customary to name the opening page of a Web site *index.htm*, *index.html*, *default.htm*, or *default.html*. When users type a URL in their Web browsers but omit the file name (for example, 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.

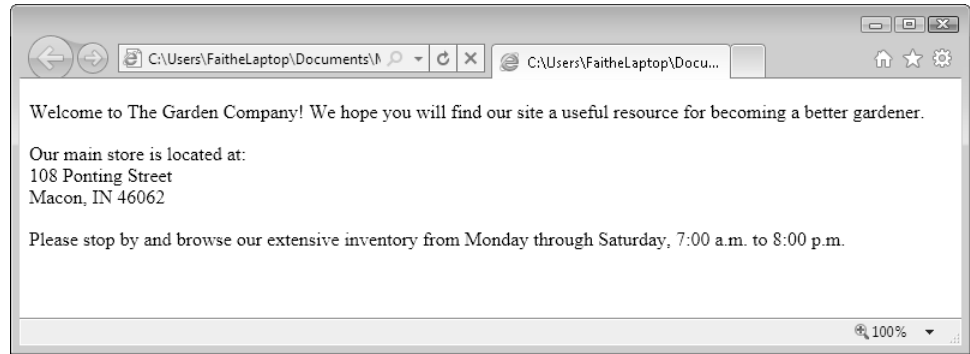
2. Open the *index* file in Internet Explorer and arrange the Notepad and Internet Explorer windows so that both are visible.

The index file displayed in Internet Explorer is currently blank.

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

```
<p>Welcome to The Garden Company! We hope you will find our site a useful
resource for becoming a better gardener.</p>
<p>Our main store is located at:<br>
108 Ponting Street<br>
Macon, IN 46062</p>
<p>Please stop by and browse our extensive inventory from Monday through
Saturday, 7:00 a.m. to 8:00 p.m.</p>
```

4. Save your work, and then press **F5** or click the **Refresh** button at the right side of the Address bar to refresh the display in Internet Explorer to see the result of the changes. Leave both windows open for the next exercise.



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

**✘ CLEAN UP** Close the Notepad and Internet Explorer windows.

## Specifying a Page Title and Metatags

Perhaps you noticed in the preceding exercise that the complete path to the file appeared in the title bar of Internet Explorer. Usually when you view a Web page, a friendly, descriptive title appears in that spot instead. That text is specified by a `<title>` tag that is placed in the `<head>` section (also called the *header*). Here's an example:

```
<head>
<title>The Garden Company</title>
</head>
```

**Troubleshooting** Make sure you place the `<title>` tag in the `<head>` section, and not 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 such as MSN. 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 Garden Company's Web site would be useful to people who are searching for information about all types of gardening problems, such as pests, weeds, and fungus, and about growing flowers and vegetables. Perhaps all these topics are not mentioned on the main page, but you want people who search for those words to be directed to the main page anyway. You could place the following in the `<head>` section:

```
<meta name="keywords" content="pests, weeds, fungus, plants, flowers,
vegetables">
```

Notice that the `<meta>` tag in the above 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 slash (/) at the end of a `<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 Web site, 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 this exercise, you will add a page title and some keywords to the `index.htm` page you created in the preceding exercise.



**SET UP** Use the *index.htm* file from the previous exercise or in the practice file folder for this topic. This practice file is located in the Documents\Microsoft Press\HTML5 SBS\02Structure\SpecifyingTitle folder. Open the *index* file in Notepad.

1. Between the `<head>` and `</head>` tags, type the following to create the page title:

```
<title>The Garden Company</title>
```

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

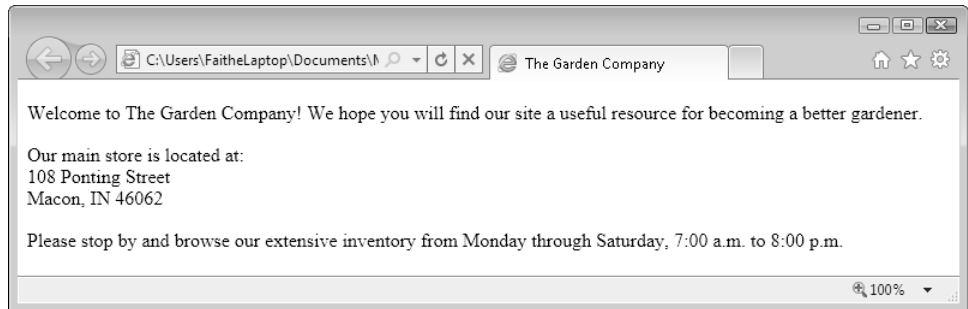
```
<meta name="keywords" content="pests, weeds, fungus, plants, flowers, vegetables">
```

3. Press **Enter** to start a new line, and 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 appear on the Web page itself.



**CLEAN UP** Close the Notepad and Internet Explorer windows.

## Publishing a File to a Server

Throughout most of this book's exercises, you will save pages to your own hard drive. That way they don't get into the public's hands before they are completed. When a page is finalized, however, you will want to transfer it to a publicly accessible Web server (that is, to publish it) so that others can view it.

There are several ways to transfer files to a server. The company or individual in charge of the server should be able to advise you about your options. Here are some of the possibilities that might be available:

- **Uploading through an FTP connection by using Internet Explorer.** You do this by entering the address of an FTP server (which will start with *ftp://*) in the Address bar of Internet Explorer. A dialog box prompts you for your user name and password for that server. If you enter those correctly, a Windows Explorer-like file-management window appears, just as though you were browsing any folder on your hard disk. You can then transfer the files by dragging them into that window, or copying them and pasting them into the FTP window.
- **Uploading through an FTP connection by using FTP software.** There are many third-party FTP applications available that make it simple to transfer files. These utilities have some advantages over the Internet Explorer transfer method, such as the ability to restart uploads that are interrupted due to communication errors. Some examples include FileZilla ([www.filezilla-project.org](http://www.filezilla-project.org)) and BulletProof FTP ([www.bpftp.com](http://www.bpftp.com)).
- **Saving directly to a Web folder.** Most Web development tools, such as Microsoft Expression Web, let you to save directly to a Web server by typing the URL of the site into the Save As dialog box. That's very convenient! Unfortunately, you can't do that in Notepad,.

This book doesn't include an exercise for practicing transferring files to a server because the process details differ depending on many factors, including the site you are saving to, the availability of FTP software, and the version of Windows you are using. If you have questions about how to upload your files, ask the network administrator or tech support staff for advice.

## Key Points

- To specify HTML5 as the document type, add `<!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 `<br>` 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 `<meta>` tags in the `<head>` section to indicate keywords and the document encoding language.
- Use the `<title>` and `</title>` tags to enclose the text that should appear in the browser's title bar. Place these in the `<head>` section of the file.
- To publish pages directly to a server, you can use an FTP utility or the FTP capability built into Windows, or (with some tools) you can save files directly to a server. However, Notepad does not offer this capability.

# Index

## Symbols

- ¢ (cent) 56
- £ (pound) 56
- ¥ (yen) 56
- & (ampersand)
  - displaying 180
  - entities and 55, 56, 366
- © (copyright) 56
- { } (curly braces) 86, 88
- † (dagger) 56
- ° (degree) 56
- ! (exclamation point) 14
- / (forward slash)
  - in tags *xix*
  - referencing folder structure 144
  - XHTML requirements 253
- > (greater than sign) 56
- < (less than sign) 56
- ± (plus or minus) 56
- # (pound sign) 76
- ® (registered trademark) 56
- ;(semicolon)
  - inserting special characters 56
  - redirect operations and 181
  - separating attributes 64
  - separating style rules 88, 109
- ™ (trademark) 56

## A

- <a> tag
  - about *xxi*
  - creating anchors 75
  - graphical navigation bars and 172
  - href attribute *xxi*, 68
  - hyperlinking graphics 153
  - inserting graphics 328
  - target attribute 70
- AAC format 274, 275, 277
- <abbr> tag 356
- absolute paths
  - defined 69, 365
  - hyperlinks using 69
- absolute size (font) 107
- accessibility
  - common issues 353
  - defined 353, 365
- accesskey attribute 358
- <acronym> tag 356, 364
- action attribute 252
- active pseudo-class 97
- Active Server Pages (ASP) 69
- ADA (Americans with Disabilities Act) 353
- .addClass() function 301
- adding
  - data file locations to Favorites list 5
  - figure captions 158–162
- Adobe Flash player
  - multimedia support 272
  - WCAG guidelines 357
- Adobe Reader, downloading 78
- align attribute
  - about 247
  - deprecated 58
- alignment
  - changing for table cells 245–248
  - defined 135, 245, 365
  - for paragraphs 135–137
  - tables and 214
- alt attribute 158, 354
- Americans with Disabilities Act (ADA) 353
- ampersand (&)
  - displaying 180
  - entities and 55, 56, 366
- anchors
  - defined 75, 365
  - hyperlinks to 75
- Android browser 272
- animations
  - <canvas> tag and 303
  - graphics support 142
- <applet> tag 364
- <area> tag 178
- arguments. *See* attributes
- <article> tag
  - about 186, 363
  - division-based layouts and 185, 187
- <aside> tag
  - about 186
  - division-based layouts and 185, 187

**ASP (Active Server Pages) 69****attributes.** *See also* **specific attributes**

- applying bold formatting 30
- applying italic formatting 30
- applying table borders 233–235
- controlling size of graphics 143
- defined xx, 365
- deprecated 58
- separating with semicolon 64
- sizing graphics 144
- styling lists 50

**audio clips.** *See* **multimedia****<audio> tag**

- about 363
- autoplay attribute 282
- controls attribute 282
- loop attribute 282
- multimedia support 273
- playing audio with 282
- preload attribute 282

**autoplay attribute**

- <audio> tag 282
- <video> tag 277, 278

**autostart attribute 283****.avi file extension 273****B****<b> tag**

- applying formatting 29–45
- nesting 35
- style sheets and 111

**background-attachment attribute 64****background color**

- applying 61
- applying to tables 241–244
- background images and 64
- changing with jQuery 296
- design considerations 60
- formatting divisions 197
- horizontal lines and 58
- setting 108
- Web form validation and 301

**background-color attribute 58, 61, 109****backgroundColor property 291, 301****background-color style rule 197, 241****background-image attribute 241****background images**

- adding to divisions 331–333
- applying to table cells 242
- background color and 64
- defined 62, 365
- specifying on Web pages 62–65
- transparency of 142

**background-repeat attribute 64****<basefont> tag 364****baseline**

- defined 365
- superscript/subscript formatting 31

**beveled borders 234****<bgsound> tag 357****BigNoseBird.Com 268****<big> tag 364****blink element 358****blinking text 115****block-level element**

- defined 135, 365
- inserting images in 145

**<blockquote> tag**

- cite attribute 38
- formatting text 37–40
- WCAG guidelines 356

**<body> section (HTML documents)**

- applying colors 61
- creating 15–16
- creating line breaks 17–19
- creating paragraphs 17–19
- defined 365
- <script> tag 291

**bold style, applying formatting 29–45, 111–114****border attribute**

- setting all sides at once 133, 235
- tables and 208, 233–235

**border-bottom attribute 235****border-bottom-width attribute 239****border-color attribute 131, 132, 235****border-left attribute 235****border-right attribute 235****borders**

- applying to paragraphs 130–135
- applying to tables 232–240
- applying with attributes 233–235
- applying with styles 235–240
- beveled 234
- Expression Web example 327
- formatting sides individually 132
- setting all attributes at once 133
- setting padding 131
- specifying color 131
- specifying styles 130
- specifying width 131
- table defaults for 207
- vertical-only 235

**border-style attribute**

- about 132, 235
- beveled borders 234
- selecting line types 130

**border-top attribute 235****border-width attribute 131, 132, 235**



**breadcrumbs**

- defined 365
- designing for usability 351

**browsers.** *See* **Web browsers****<br> tag** 17–19**bulleted lists**

- changing characters 50
- creating 48–53

**BulletProof FTP** 22**button-creation programs** 172, 365**buttons**

- Clear 255
- command 253, 365
- creating graphical navigation bars 171–173
- formatting divisions and 198
- free download sites 172
- option 259–262, 260
- radio 260
- Reset 255
- Submit 255

**C****<canvas> element**

- about 287–288
- additional information 309

**<canvas> tag**

- about 287–288, 363
- JavaScript support 303

**cascading style sheet.** *See* **CSS (cascading style sheet)****CDN (Content Delivery Network)** 293**cellpadding attribute** 246**cellspacing attribute** 246**cells (table)**

- applying background color 242
- applying background images 242
- applying borders 232, 236
- changing alignment 245–248
- changing padding 245–248
- changing spacing 233, 245–248
- defined 205, 365
- merging 220–224
- nonbreaking spaces in 217
- WCAG guidelines 356

**cent (¢)** 56**<center> tag** 364**CGI Resource Index** 268**CGI scripts**

- drawbacks of 268
- sending form contents to 253

**character-based formatting**

- adding spacing between letters 119–122
- applying bold 111–114

applying italics 111–114

applying strikethrough 114–116

applying underlining 114–116

creating inline spans 117–119

defined 103

specifying font color 107–111

specifying font family 104–107

specifying font size 107–111

**character encoding schemes** 20**check boxes**

- creating 259–262
- defined 259

**checked attribute** 260**child folders**

- defined 70, 365
- inserting graphics into 146

**Chrome browser**

- audio support 272, 282
- displaying table borders 234
- video support 272, 279

**circular hotspots** 175**cite attribute** 38**class attribute** 94**classes**

- creating for applying styles 93–96
- defined 93, 365
- pseudo-classes 96, 367
- removing from elements 301

**Clear button** 255**.click() function** 298, 307**closing tags** xix**codec (compression/decompression)**

- defined 274, 365

**<code> tag** 33**color.** *See also* **background color; foreground color**

- additional information 61
- dithered 60, 365
- extended names 61
- formatting divisions and 197
- graphics preparation 144
- specifying 60
- universally accepted 59, 60
- WCAG guidelines 354
- Web-safe 60, 368

**color attribute**

- about 108
- applying styles 58
- deprecated 58

**colspan attribute** 221**columns attribute** 255**column width in tables** 216–220**command buttons** 253, 365**<command> tag** 363**compiled programming languages** xix, 365

[Comprehensive Perl Archive Network](#) 268  
[compression algorithms](#) 275, 276  
[compression schemes](#) 142, 143  
[container formats](#) 273  
[content attribute](#) 181  
[Content Delivery Network \(CDN\)](#) 293  
[controls attribute](#)  
   [<audio> tag](#) 282  
   [<video> tag](#) 277, 278  
[copyright symbol \(©\)](#) 56  
[Crystal Button](#) 172  
[CSS \(cascading style sheet\)](#)  
   [applying table borders](#) 233, 235  
   [creating Web pages](#) 325–328  
   [defined](#) xxi, 85, 365  
   [horizontal lines](#) 58  
   [selecting elements via](#) 296  
   [styling lists](#) 50  
   [WCAG guidelines](#) 357  
[.css file extension](#) 98  
[curly braces {}](#) 86, 88

## D

[dagger \(†\)](#) 56  
[<datagrid> tag](#) 363  
[<datalist> tag](#) 363  
[<datatemplate> tag](#) 363  
[date pickers](#) 267  
[<dd> tag](#) 54, 365  
[default.aspx file](#) 69  
[default.css file](#) 194, 335  
[default text](#) 256  
[definition description](#) 54, 365  
[definition lists](#)  
   [creating](#) 54  
   [defined](#) 54, 365  
[definition term](#) 54, 365  
[degree \(°\)](#) 56  
[<del> tag](#) 115  
[deprecated tags](#)  
   [attributes and](#) 51  
   [defined](#) 365  
   [horizontal lines](#) 58  
   [in HTML5](#) 364  
   [strikethrough/underlining](#) 115  
[descriptive tags](#) 365  
[design. See navigational aids](#)  
[<details> tag](#) 363  
[device independence](#) 358  
[<dialog> tag](#) 363  
[<dir> tag](#) 364  
[display size, video clips](#) 275  
[dithered color](#) 60, 365

[div element](#)  
   [backgroundColor property](#) 291  
   [fade out example](#) 298  
   [id attribute](#) 295  
[division-based layouts](#)  
   [about](#) 187  
   [advantages](#) 185  
   [creating divisions](#) 188–189  
   [creating semantic layouts](#) 190–191  
   [defined](#) 185  
   [formatting divisions](#) 197–202  
   [positioning divisions](#) 192–197  
   [semantic tags](#) 186  
   [table-based design and](#) 206  
   [WCAG guidelines](#) 355  
[divisions](#)  
   [adding background images to](#) 331–333  
   [applying borders to](#) 130  
   [creating](#) 188–189  
   [defined](#) 185, 187  
   [Expression Web example](#) 326  
   [formatting](#) 197–202  
   [formatting with Expression Web](#) 338–340  
   [positioning](#) 192–197  
   [vertical navigation bars and](#) 224  
[<div> tag](#)  
   [applying borders to](#) 130  
   [division-based layouts and](#) 185, 187  
   [id attribute](#) 188  
   [<nav> tag and](#) 189  
   [<section> tags and](#) 186  
   [semantic tags and](#) 193  
[DivX format](#) 274  
[<dl> tag](#) 54, 365  
[DOCTYPE tag](#)  
   [about](#) 14  
   [quirks mode and](#) 14, 367  
   [standards mode and](#) 14, 367  
[DOM \(Document Object Model\)](#) 298  
[downloading](#)  
   [Adobe Reader](#) 78  
   [free graphical buttons](#) 172  
   [jQuery](#) 293  
   [Microsoft Download Center and](#) 78  
   [online edition of book](#) xvii–xviii  
   [Web browsers](#) 10  
[<dt> tag](#) 54, 365

## E

[ECMA-262 specification](#) 289  
[editing Web pages](#) 10–11  
[elements](#)  
   [arranging on Web pages](#) 147–149

- block-level 135, 145, 365
  - removing classes from 301
  - selecting with jQuery 295–297
  - e-mail addresses**
    - cautions displaying 72
    - hyperlinking to 71–75
    - input field types and 254
    - subject attribute 72
  - e-mail programs**
    - delivering user forms via 252
    - delivery speeds 259
  - embedded style sheets**
    - about 87
    - applying table borders 236
    - external styles sheets and 99
  - embedding graphics 141**
  - <embed> tag**
    - about 363
    - audio/video support 272, 279
    - autoplay attribute 283
    - height attribute 310
    - including external content 310
    - loop attribute 283
    - src attribute 283, 310
    - type attribute 310
    - width attribute 310
  - em (font size) 365**
  - <em> tag 30**
  - enctype attribute 252**
  - end slash 18**
  - entities 55, 366. See also special characters**
  - entity names**
    - common symbols 56
    - defined 366
    - inserting special characters 56
  - entity numbers**
    - common symbols 56
    - defined 366
    - inserting special characters 56
  - event handling 292, 298**
  - <eventsource> tag 310, 363**
  - exclamation point (!) 14**
  - Expression Web**
    - about 315
    - applying bold/italic formatting 30
    - Code View 317
    - creating pages using CSS templates 325–328
    - creating Web pages 320–324
    - creating Web sites 320–324
    - Design View 318
    - exploring the interface 316–320
    - formatting divisions 338–340
    - formatting text 333–338
    - HTML5-compliant code and 322
    - inserting graphics 328–333
    - inserting hyperlinks 340–343
    - installing 316
    - opening 316
    - WYSIWYG mode 318
  - extended names 366**
  - Extensible HTML. See XHTML (Extensible HTML)**
  - Extensible Markup Language (XML)**
    - about xxiii, 366
    - WCAG guidelines 357
  - external style sheets**
    - creating 87, 98–100
    - defined 98, 366
    - division-based layouts and 185
    - embedded style sheets and 99
    - linking to 98–100
- ## F
- .fadeOut() function 296**
  - Favorites list 5**
  - ffmpeg2theora 277**
  - figure captions 158–160**
  - <figurecaption> tag 144, 159**
  - <figure> tag**
    - about 363
    - adding figure captions 158
    - inserting graphics 144
  - file extensions**
    - defined 4
    - Expression Web default 324
    - known file types 5
  - file locations, adding to Favorites list 5**
  - files**
    - hyperlinks to 78–80
    - opening from Windows Explorer 7
    - publishing to servers 22
  - file size**
    - defined 143, 366
    - video clips 275
  - FileZilla 22**
  - fillRect function 304**
  - Firefogg plug-in 277**
  - Firefox browser**
    - audio support 272, 275, 282
    - <canvas> element and 306
    - displaying table borders 234
    - NoScript add-on 292
    - video support 272, 275, 279
  - fixed-width tables 214**
  - Flanders, Vincent 347**
  - float style rule**
    - left attribute 145, 192
    - positioning divisions 192
    - right attribute 145, 192
  - .flv file extension 273**

**focus pseudo-class** 97

**font color** 107–123

**font family**

- defined 104, 366
- formatting text and 26
- specifying 104–107

**font-family attribute** 105

**font size**

- em multiplier 365
- heading tags and 27
- monospace fonts and 32, 367
- proportional fonts and 32, 367
- specifying 107–111

**font-style attribute** 112

**<font> tag** xxiv, 25

**font-weight attribute** 111

**<footer> tag**

- about 186, 363
- division-based layouts and 187

**foreground color**

- applying 61
- applying to tables 242
- color attribute and 108
- defined 366

**formatting text.** *See also* **character-based**

**formatting; paragraph formatting**

- applying bold styles 29–31
- applying italic styles 29–31
- applying subscript 31
- applying superscript 31
- applying with styles 25, 103
- block quotations 37–40
- via cascading style sheets xxi
- configuring views in Internet Explorer 40–44
- creating headings 26–29
- Expression Web example 333–338
- monospaced text 32–37
- nonbreaking space and 56
- preformatted text 32–37
- semantic tags and 26

**<form> tag**

- about 252
- action attribute 252
- enctype attribute 252
- method attribute 252

**forward slash (/)**

- in tags xix
- referencing folder structure 144
- XHTML requirements 253

**fps (frames per second)** 276

**frame attribute** 234

**frame, defined** 366

**frameset, defined** 366

**<frameset> tag** xxiv, 364

**frames per second (fps)** 276

**<frame> tag** xxiv, 364

**FTP software** 22

**function arguments** 297

**functions**

- calling with JavaScript 297
- defined 297
- hierarchical 296
- return values 297

## G

**getContext() function** 304

**getElementById() method** 291

**getElementsbyTagName() method** 291

**GIF (Graphics Interchange Format)**

- format comparisons 143
- graphical navigation bars and 171
- transparency support 198

**graphics (images)**

- adding figure captions 158–160
- alternate text for 158
- arranging elements on Web pages 147–149
- controlling image size/padding 149–153
- creating in navigation bars 171–173, 172
- embedding 141
- hyperlinking from 153–155, 174
- inserting 144–162, 328–333
- preparing for Web use 143
- resolution of 143, 367
- thumbnail 155–157
- WCAG guidelines 354, 361

**graphics formats**

- for navigation bars 171
- selecting 142–143

**Graphics Interchange Format (GIF)**

- format comparisons 143
- graphical navigation bars and 171
- transparency support 198

**greater than sign (>)** 56

## H

**H.264 format** 274, 275, 277

**Handbrake software** 277

**hanging (indent)** 366

**<header> tag**

- about 186, 363
- creating semantic layouts 190
- division-based layouts and 187

**headings**

- about 26
- constructing style rules 87
- creating 26–29

- defining font sizes 108
  - WCAG guidelines 355
  - <head> section (HTML documents)**
    - constructing style rules 87
    - creating 15–16
    - CSS code in 85
    - defined 366
    - redirecting operations and 181
    - <script> tag 291
    - <style> section in 86
  - height attribute**
    - applying styles 58
    - controlling image size 150
    - <embed> tag 310
    - multimedia support 277, 278
    - specifying table size 212, 215
  - hexadecimal color values 60, 108**
  - <hgroup> tag 27**
  - hierarchical functions 296**
  - horizontal alignment**
    - for table cells 247
  - horizontal alignment**
    - for paragraphs 135–137
  - horizontal lines**
    - inserting 57–59, 169
    - moving 148
    - text-based navigation bars 167
  - hotspots 174**
  - .hover() function 298**
  - hover pseudo-class 97**
  - href attribute**
    - hyperlinks to e-mail addresses 72
    - hyperlinks to Web pages 68
  - <hr> tag 58**
  - .html file extension 324**
  - HTML4**
    - about xxii
    - DOCTYPE tag and 14
  - HTML5**
    - about xxii
    - additional information xxii
    - coding in xxiii
    - deprecated tags 58
    - DOCTYPE tag and 14
    - Expression Web and 322
    - tags added in 363
    - tags removed in 364
  - HTML documents. See also <body> section (HTML documents); <head> section (HTML documents); Web pages**
    - defined xix
    - editing 10–11
    - line breaks in 17–19
    - metatags and 15, 19–21, 367
    - page title 15, 19–21, 366, 367
    - paragraphs in 17–19
    - placing forms in 252
    - setting up structure 13–23
    - specifying type 14
  - HTML (Hypertext Markup Language)**
    - choosing versions xxii
    - defined xv, xix, 366
    - multimedia support 273–277
    - system requirements xxiv
  - <html> tag**
    - about 15–16
    - WCAG guidelines 356
  - hyperlinks**
    - absolute paths and 69
    - activating 67
    - to anchors 75
    - applying styles to 96–98
    - breadcrumbs 365
    - defined xx, 366
    - to e-mail addresses 71–75
    - to external style sheets 98–100
    - to files 78–80
    - from graphics 153–155, 174
    - inserting 340–343
    - multimedia support 273
    - navigation bars and 165, 166, 367
    - to non-HTML content 78–80
    - partial paths/filenames 68
    - to pictures 78
    - relative paths and 69
    - ScreenTips for 72
    - setting target windows 70
    - underlining and 115
    - visited 96, 368
    - WCAG guidelines 354
    - to Web pages 68–71
  - Hypertext Markup Language. See HTML (Hypertext Markup Language)**
- I**
- id attribute**
    - about 94
    - creating divisions 188
    - div element 295
    - image maps 177
  - ID (identifier)**
    - creating for applying styles 93–96
    - defined 93, 366
  - if conditional 301**
  - image maps**
    - creating 174–180
    - defined 174, 366
    - WGAC guidelines 354
  - images. See graphics (images)**

**images folder, importing** 328–333**<img> tag**

- about xx
- adding figure captions 158
- alt attribute 158
- graphical navigation bars and 172
- height attribute 150
- hyperlinking graphics 153
- src attribute xx
- usermap attribute 178
- width attribute 150

**importing images folder** 328–333**indentation**

- defined 126, 366
- keyboard shortcuts 52
- nonbreaking space and 56
- paragraph formatting 126–129
- readability and 49
- removing unwanted 129
- WCAG guidelines 356

**index.htm page** 68**inline quotations** 38**inline spans**

- creating 117–119
- defined 117, 366

**<input> tag**

- checked attribute 260
- max attribute 267
- maxlength attribute 254
- min attribute 267
- name attribute 253, 260
- placeholder attribute 256
- required attribute 254
- size attribute 254
- type attribute 253, 254, 259
- value attribute 255, 256, 259, 267

**Insert | Hyperlink command** 340**inserting**

- graphics 144–147, 328–333
- horizontal lines 57–59, 169
- hyperlinks 340–343
- nonbreaking spaces 56, 169
- special characters 55–57

**<ins> tag** 115**Internet Explorer**

- audio/video support 272, 274, 275, 279, 282
- configuring views in 40–44
- displaying table borders 234
- extended colors 61
- FTP connections 22
- previewing Web pages in 8–10
- relative font sizes and 108
- text wrapping in tables 217
- tooggling status bar 74
- WCAG guidelines 356

**interpreted programming languages** xix, 366**iPhone browser**

- audio/video support 272
- e-mail addresses and 254

**<isindex> tag** 364**<i> tag**

- applying formatting 29–31
- style sheets and 111

**italic style, applying formatting** 29–45, 111–123**J****Java applets** 358**JavaScript**

- about 288
- additional information 303
- calling functions with 297
- canvas element and 288
- <canvas> tag and 303
- event handling and 292, 298
- fillRect function 304
- getContext() function 304
- getElementById() method 291
- getElementsByTagName() method 291
- showAlert() function 297
- strokeRect function 304
- style() function 291
- using on Web pages 289–292
- validating Web forms 300–303
- WCAG guidelines 357
- Web browsers and 292

**JPEG (Joint Photographic Experts Group)**

- format comparisons 143
- graphical navigation bars and 171
- transparency considerations 198

**jQuery**

- about 289, 292
- .addClass() function 301
- additional information 303
- .click() function 298, 307
- downloading 293
- event handling and 298
- .fadeOut() function 296
- .hover() function 298
- preparing for 294
- .ready() function 294, 298, 302
- .removeClass() function 301
- selecting elements 295–297
- .submit() function 298, 303
- .val() function 301
- validating Web forms 300–303

## K

[<kbd> tag](#)  
 about 33  
 nesting tags 35  
[keyboard shortcuts](#). *See shortcuts*  
[keywords](#). *See <meta> tag*

## L

[label attribute](#) 360  
[lang attribute](#) 356  
[large \(relative font size\)](#) 107  
[leading](#)  
     defined 137, 366  
     specifying 137–138  
[left attribute](#) 145  
[less than sign \(<\)](#) 56  
[letter spacing](#) 119  
[line breaks](#)  
     creating line breaks 17–19  
     CSS considerations 86  
[line-height style](#) 107, 138  
[link pseudo-class](#) 96  
[list items](#) 48, 366  
[lists](#)  
     bulleted 48–53  
     creating 262–266  
     defined 54, 262  
     nesting 49, 52, 356  
     numbered 48–53  
[list-style-type: type attribute](#) 50  
[<li> tag](#) 48, 366  
[load event](#) 294  
[logical tags](#). *See descriptive tags*  
[longdesc attribute](#) 360  
[loop attribute](#)  
     <audio> tag 282  
     <embed> tag 283  
     <video> tag 277, 278  
[lossless compression](#) 142, 143  
[lossy compression](#) 142, 143

## M

[<map> tag](#)  
 about 177  
 id attribute 177  
 name attribute 177  
[margin attribute](#)  
 about 150  
 default actions 127  
 indenting paragraphs 126  
[<mark> tag](#) 363

[marquee element](#) 358  
[Matt's Script Archive](#) 268  
[max attribute](#) 267  
[maxlength attribute](#) 254, 255  
[medium \(relative font size\)](#) 107  
[menus](#). *See lists*  
[merging table cells](#) 220–224  
[<meta> tag](#)  
     defined 367  
     <head> section and 15  
     redirecting operations and 181  
     search engines and 20  
     specifying 19–21  
[<meter> tag](#) 363  
[method attribute](#) 252  
[Microsoft Bing Maps](#) 292  
[Microsoft Download Center](#) 78  
[Microsoft Expression Web](#). *See Expression Web*  
[Microsoft Notepad](#). *See Notepad*  
[Microsoft Office, default fonts](#) 42  
[Microsoft Windows Media Player\)](#) 273  
[Microsoft Word, embedding graphics](#) 141  
[mkv file extension](#) 273  
[monospace fonts](#)  
     defined 32, 367  
     formatting text 32–37  
[MooTools framework](#) 295  
[MP3 format](#) 274, 282  
[mp4 container](#) 275  
[.mp4 file extension](#) 273  
[MPEG-4 Audio Layer 3 \(codec\)](#) 274  
[multimedia](#)  
     encoding video 276–277  
     formats and containers 273  
     HTML basics 273–277  
     including in Web pages 310  
     incorporating audio on Web pages 282–284  
     placing video clips on Web pages 279–282  
     playing audio 282–284  
     WCAG guidelines 354, 357, 361

## N

[name attribute](#)  
     creating image maps 177  
     defining anchors 75  
     option buttons and 260  
     text boxes and 253  
[natural language usage](#) 356  
[navigational aids](#)  
     creating graphical navigation bars 171–173  
     creating image maps 174–180  
     creating text-based navigation bars 167–171  
     planning site organization 166–167  
     redirecting to other URLs 180–182

**navigation bars**

- defined 165, 367
- formatting differently 190
- graphical 171–173, 172
- hyperlinks and 165, 166, 367
- modifying via division-based layouts 185
- planning site organization 166–167
- positioning divisions 192
- text-based 167–171, 172
- usability planning 349
- vertical 224–228
- WCAG guidelines 360
- width adjustments 198

**<nav> tag**

- about 168, 189, 363
- defining navigation bars differently 190
- division-based layouts and 187
- positioning divisions 192

**nesting**

- defined 367
- lists 49, 52, 356
- rules for tags 29, 35, 91–93

**<nest> tag 363****Netscape browser 234****<noframes> tag 364****nonbreaking space**

- in empty cells 217
- inserting 56, 169

**none keyword 115****<noscript> tag 292****Notepad**

- about xv
- opening Web pages in 3–7

**numbered lists**

- changing characters 50
- creating 48–53
- defined 48, 367
- specifying start of 51–53

**O****<object> tag 272, 273****oblique font style 112****Ogg container 274, 275****.ogv file extension 273****<ol> tag**

- about 48, 367
- applying attributes 50
- start attribute 51
- value attribute 51

**one-sided tags**

- as self-closing 20
- defined xx, 367
- XHTML and 18

**online edition**

- accessing xv–xvii
- downloading to computer xvii–xviii

**onload event 294****opening**

- Expression Web 316
- files from Windows Explorer 7
- Web pages in Notepad 3–11

**opening tags xix****Opera browser**

- audio support 272, 275, 282
- video support 272, 275

**option buttons 259, 260****<optgroup> tag 264****<option> tag 263****ordered lists**

- creating 48–53
- defined 48, 367
- paragraph alignment and 49

**<output> tag 363****overriding style rules 236****P****padding**

- applying to all element sides 127
- changing for borders 131
- changing for table cells 245–248
- defined 126, 245
- element sides individually 132
- paragraph formatting and 150

**padding attribute**

- about 126, 131, 150
- formatting sides individually 127, 132

**page layout**

- creating user forms 251–269
- designing consistent templates 349
- division-based 185–203
- formatting tables 231–249
- incorporating sound/video 271–285
- table-based 205–229
- WCAG guidelines 361

**page title (HTML documents)**

- defined 366, 367
- Expression Web example 326
- location of 15
- specifying 19–21

**Paint Shop Pro 176****paragraph formatting**

- applying borders 130–135
- defined 125, 367
- indentation 126–129
- specifying paragraph alignment 135–137
- specifying vertical space 137–138



**paragraphs.** *See also* <p> tag  
 aligning 135–137, 365  
 applying borders to 130–135  
 creating in HTML documents 17–19  
 indenting 126–129  
 removing unwanted indentation 129  
 specifying vertical space 137–138

**parent folders** 70, 367

**phpBB bulletin board package** 268

**pixels**  
 image resolution in 143, 367  
 specifying table size with 214  
 as unit of measurement 107  
 video clip display size 275

**placeholder attribute** 256

**placeholder text** 256

**planning site organization** 166–167

**player (program)** 367

**plug-ins** 78, 367

**plus or minus ( $\pm$ )** 56

**PNG (Portable Network Graphics)**  
 format comparisons 143  
 transparency support 198

**poly hotspots** 174, 175

**Portable Network Graphics (PNG)**  
 format comparisons 143  
 transparency support 198

**positioning divisions** 192–197

**position style rule** 193–197

**pound sign (#)** 76

**pound symbol (£)** 56

**preformatted text** 32–37

**preload attribute**  
 <audio> tag 282  
 <video> tag 277, 278

**<pre> tag** 33, 34

**previewing Web pages** 7–11

**profiles, defined** 274

**programming languages**  
 compiled xix, 365  
 interpreted xix, 366

**<progress> tag** 363

**proportional fonts**  
 defined 32, 367  
 formatting text 32–37

**prototypeJS** 295

**pseudo-classes** 96, 367

**<p> tag.** *See also* paragraphs  
 block quotations and 39  
 creating in HTML documents 17–19  
 removing unwanted indentation 129  
 XHTML cautions xxiii

**publishing files to servers** 22

**px (unit of measure)** 107

## Q

**<q> tag**  
 about 38  
 WCAG guidelines 356

**quirks mode**  
 defined 14, 367  
 DOCTYPE tag and 14, 367

**quotation marks, font names and** 105

## R

**radio buttons** 260

**RDF (Resource Description Framework)** 360

**.ready() function** 294, 298, 302

**rectangular hotspots** 174

**redirecting Web pages**  
 canceling operation 181  
 defined 367  
 to another URL 180–182  
 via metatags 20  
 WCAG guidelines 358

**registered trademark (®)** 56

**relative paths**  
 defined 69, 367  
 hyperlinks using 69

**relative size (font)** 107

**.removeClass() function** 301

**required attribute** 254

**Reset button** 255

**resolution**  
 defined 143, 367  
 sample 276

**Resource Description Framework (RDF)** 360

**return values** 297

**RGB color values** 60, 108

**right attribute** 145

**rows attribute** 255

**rowspan attribute** 221

**rules.** *See* style rules

**rules attribute** 234, 235

**<rule> tag** 363

## S

**<s> tag** 115, 364

**Safari browser**  
 audio support 272, 275, 282  
 video support 272, 275, 279

**sample resolutions** 276

**samples** 276, 367

**sampling rates** 276, 367

**<samp> tag** 33

[saving changes in Web pages](#) 10–11

[screen reader programs](#) 27

[ScreenTips](#)

- for hyperlinks 72
- title attribute and 178
- usefulness of 180

[<script> tag](#)

- about 289
- src attribute 289

[search engines, metatags and](#) 20

[<section> tag](#)

- about 186, 363
- division-based layouts and 187

[security](#)

- CGI scripts and 268
- JavaScript validation and 300

[<select> tag](#) 263

[self-closing tags. \*See one-sided tags\*](#)

[semantic tags](#)

- about 186
- creating layouts with 190–191
- defined 26, 186, 367
- division-based layouts and 187
- <div> tags and 193

[semicolon \(;\)](#)

- inserting special characters 56
- redirect operations and 181
- separating attributes with 64
- separating style rules 88, 109

[shading, table defaults for](#) 207

[shortcuts](#)

- adding file locations to Favorites list 5
- indenting code 52
- specifying arguments for border sides 132

[showAlert\(\) function](#) 297

[size attribute](#)

- deprecated 58
- list height 263
- text box widths 254

[sliders](#) 267

[small \(relative font size\)](#) 107

[<source> tag](#) 363

[spacing](#)

- adding between letters 119–122
- changing for table cells 245–248
- defined 245
- option buttons and 260

[spam](#) 72, 367

[spanning table cells](#) 220

[<span> tag](#)

- creating inline spans 118
- WCAG guidelines 356

[special characters](#)

- ampersand as 180
- defined 55, 367

- entities as 366
- inserting 55–57

[spin boxes](#) 267

[src attribute](#)

- <embed> tag 283, 310
- <img> tag xx
- <script> tag 289

[standards mode](#)

- defined 14, 367
- DOCTYPE tag and 14, 367

[start attribute](#) 51

[status bar, toggling on/off](#) 74

[<strike> tag](#) 115, 364

[strikethrough formatting](#) 114–116

[strokeRect function](#) 304

[<strong> tag](#) 30

[style attribute](#)

- adjusting spacing between letters 120
- applying bold/italics 111
- applying colors 61
- applying special characters 86
- CSS coding and 86
- defining font color 108
- horizontal line and 58
- overriding style rules 236

[style\(\) function](#) 291

[style rules](#)

- applying table borders 235
- applying to tags 86
- constructing 87–90
- curly braces and 86, 88
- defined 367
- inserting graphics 145
- nested tags 29, 35, 91–93
- overriding 236
- placement considerations 194
- positioning divisions 193–197
- pseudo-classes and 97
- separating with semicolons 88, 109

[styles. \*See also CSS \(cascading style sheet\)\*](#)

- applying bold/italic 29–31
- applying borders 235–240
- applying table borders 233
- applying to hyperlinks 96–98
- border 130
- classes/IDs for applying 93–96
- constructing rules 87–90
- defined 86, 367
- horizontal lines 58
- for nested tags 91–93
- text formatting via 103

[<style> section \(HTML documents\)](#)

- about 86
- constructing style rules 87–90

**style sheets.** *See also* [character-based formatting](#);

[CSS \(cascading style sheet\)](#); [paragraph formatting](#)

- adding spacing between letters 119–122
- applying bold 111–114
- applying borders to paragraphs 130–135
- applying italics 111–114
- applying strikethrough 114–116
- applying styles to hyperlinks 96–98
- applying underlining 114–116
- classes for applying styles 93–96
- constructing style rules 87–90
- creating inline spans 117–119
- embedded 87
- external 87, 98–100, 185, 366
- IDs for applying styles 93–96
- indenting paragraphs 126–129
- specifying font color 107–111
- specifying font size 107–111
- specifying paragraph alignment 135–137
- specifying table size 212
- specifying vertical space 137–138
- styles for nested tags 91–93
- text-based navigation bars 168
- WCAG guidelines 355

**subfolders.** *See* [child folders](#)

**?subject attribute** 72

**Submit button** 255

**.submit() function** 298, 303

**subscript formatting** 31

**superscript formatting** 31

## T

**tabindex attribute** 358

**tables.** *See also* [cells \(table\)](#)

- applying background color 241–244
- applying borders 232–240
- applying foreground color 242
- creating 207–211
- defined 205, 367
- division-based layouts and 206
- fixed-width 214
- merging cells 220–224
- organizing form fields 252
- as page-layout tool 206, 224–228
- specifying column width 216–220
- specifying size of 211–216
- text wrapping in 217
- vertical navigation bars and 224–228
- WCAG guidelines 356

**<table> tag**

- applying table borders 235
- applying vertical borders 235
- background-image attribute 241

- border attribute 208, 233
- cellpadding attribute 246
- cellspacing attribute 247
- creating tables 207–211
- height attribute 212, 215
- <td> tag and 207
- troubleshooting table widths 219
- <tr> tag and 207
- width attribute 211

**tags.** *See also* [specific tags](#)

- applying styles to 86
- closing xix
- defined 367
- defining font sizes 108
- descriptive 365
- forward slash in xix
- hyperlinks and xx
- nesting 29, 35, 91–93
- one-sided xx, 18, 20, 367
- opening xix
- overview xix–xxi
- two-sided xix, 17, 367
- WCAG guidelines 355
- XHTML requirements 15

**target attribute** 70, 155

**<td> tag**

- about 207
- align attribute 247
- applying table borders 235, 236
- colspan attribute 221
- rowspan attribute 221
- specifying column width 216
- table columns and 208
- troubleshooting table widths 219
- WCAG guidelines 356

**templates**

- creating Web pages 325–328
- designing consistent 349
- read-only 16

**testing, usability** 351

**text-align attribute** 135, 153, 247

**text areas**

- adding default text 256–259
- adding placeholder text 256–259
- creating 255

**<textarea> tag**

- about 255
- columns attribute 255
- maxlength attribute 255
- rows attribute 255

**text-based navigation bars** 167–171, 172

**text boxes**

- adding default text 256–259
- adding placeholder text 256–259
- creating 253
- validating 300

[text-decoration attribute](#) 115  
[text formatting](#). *See* [formatting text](#)  
[text-indent attribute](#) 126  
[themes, defined](#) 367  
[Theora format](#) 274, 275  
[<th> tag](#)  
   about 209  
   WCAG guidelines 356  
[thumbnail graphics](#) 155–157  
[time-sensitive content](#) 357  
[<time> tag](#) 363  
[title attribute](#)  
   ScreenTips for hyperlinks 72  
   ScreenTips for image maps 178  
   WCAG guidelines 360  
[<title> tag](#) 19, 366, 367  
[trademark \(™\)](#) 56  
[transferring files to servers](#) 22  
[transparency](#) 142, 198  
[troubleshooting table widths](#) 219  
[<tr> tag](#)  
   about 207  
   border-bottom-width attribute 239  
[<tt> tag](#) 33, 364  
[two-sided tags](#)  
   defined xix, 367  
   XHTML and 17  
[type attribute](#)  
   creating check boxes 259  
   creating text boxes 253  
   e-mail addresses and 254  
   <embed> tag 310  
   multimedia support 279

## U

[<u> tag](#) 115, 364  
[<ul> tag](#)  
   about 48, 367  
   applying attributes 50  
   specifying 49  
[underlining formatting](#)  
   applying 114–116  
   hyperlinks and 115  
[unit of measurement](#)  
   pixels as 107  
   WCAG guidelines 355  
[unordered lists](#)  
   applying styles 86  
   defined 48, 367  
[url attribute](#) 181  
[URLs, redirecting Web pages to](#) 180–182

[usability](#)  
   additional resources 348  
   defined 347, 368  
   designing consistent templates 349  
   designing individual page content 350  
   performing testing 351  
   planning for 348  
   qualities of 347  
   sketching site organization 349  
[Usability.gov](#) 347  
[user forms](#)  
   additional input types 267  
   creating 252–259  
   creating check boxes 259–262  
   creating lists 262–266  
   creating option buttons 259–262  
   e-mail delivery and 252  
   validating 300–303  
   WCAG guidelines 359  
[usermap attribute](#) 178

## V

[.val\(\) function](#) 301  
[validating Web forms](#) 300–303  
[valign attribute](#) 247  
[value attribute](#)  
   creating check boxes and 259  
   default text and 256  
   default values 267  
   displaying text on buttons 255  
   ordered lists and 51  
   reporting form results 263  
[vertical-align attribute](#) 247  
[vertical alignment](#) 247  
[video cards, dual monitor connections](#) 8  
[video clips](#). *See* [multimedia](#)  
[video encoding formats](#) 274  
[<video> tag](#)  
   about 272, 277–282, 363  
   autoplay attribute 277, 278  
   controls attribute 277, 278  
   height attribute 277, 278  
   loop attribute 277, 278  
   multimedia support 273  
   preload attribute 277, 278  
   type attribute 279  
   Web browser support 275  
   width attribute 277, 278  
[viewing changes in Web pages](#) 10–11  
[view settings, configuring in Internet Explorer](#) 40–44  
[visited hyperlinks](#) 96, 368  
[visited pseudo-class](#) 96

**VLC media player** 277, 282

**Vorbis format**

about 275

MP3 format and 282

Ogg containers and 274

WebM format and 274

**VP8 format** 274, 275

## W

**W3C (World Wide Web Consortium)**

defined 368

WCAG guidelines 359

Web Accessibility Initiative 353

**WCAG (Web Content Accessibility Guidelines)**

about 353

accessibility of embedded interfaces 358

color considerations 354

content delivery 357

content information 360

creating tables 356

device independence 358

document clarity 361

interim solutions 359

natural language usage 356

navigation mechanisms 360

orientation information 360

providing equivalent alternatives 354

time-sensitive content 357

W3C guidelines and 359

**Web addresses** 254

**Web browsers.** *See also specific browsers*

<canvas> element and 303

customizing 26

default text and 256

displaying table borders 234

downloading 10

<eventsource> tag and 310

formatting text and 26

historical background xx

HTML support xxii, xxiv

hyperlink support xx

input field types and 254

interpreting color 60

JavaScript considerations 292

multimedia support 272, 274, 283

previewing Web pages 7–10

responding to events 298

scaling size of graphics 143

special characters and 56

standardization considerations 14

text-based navigation bars 168

text wrapping in tables 217

**Web Content Accessibility Guidelines.** *See WCAG (Web Content Accessibility Guidelines)*

**Web design**

for accessibility 353–361

for usability 347–351

**WebM format** 274, 275, 277

**Web pages.** *See also divisions; HTML documents; page layouts*

adding canvas elements 288

arranging elements on 147–149

button-creation programs 365

controlling size of graphics 143

creating using CSS templates 325–328

creating with Expression Web 320–324

defined xix, 368

history of formatting xx

hyperlinks to 68–71

including external content in 310

incorporating audio on 282–284

inserting graphics on 144–147

inserting horizontal lines 57–59

inserting special characters 55–57

JavaScript support 289–292

opening in Notepad 3–7

placing audio clips on 283

placing images on 330

placing video clips on 279–282

positioning divisions on 193–197

previewing in Web browsers 7–10

redirecting 20, 180–182, 358, 367

saving changes in 10–11

specifying background images 62–65

specifying colors 60–62

viewing changes in 10–11

**Web Pages That Suck site** 347

**Web-safe colors** 60, 368

**Web servers**

publishing files to 22

saving directly to 22

**Web sites**

creating with Expression Web 320–324

defined 320

designing for usability 347–351

**width attribute**

applying styles 58

controlling graphics size 150

deprecated 58

<embed> tag 310

multimedia support 277, 278

specifying for table columns 216–220

specifying table size 211

**Windows environment, default fonts** 42

**Windows Explorer, opening files from** 7

**word spacing** 119

**World Wide Web Consortium (W3C)**

- defined 368
- WCAG guidelines 359
- Web Accessibility Initiative 353

## X

- x coordinates** 304
- XHTML 1.0 Transitional** 322, 326
- XHTML (Extensible HTML)**
  - about xxii, xxiii
  - definition lists 54
  - DOCTYPE tag and 14
  - forward slash and 253
  - horizontal lines 58
  - line breaks in 18
  - nesting tags 29, 35
  - required tags 15
  - target attribute and 70
  - two-sided tags and 17

- x-large (relative font size)** 107
- XML (Extensible Markup Language)**
  - about xxiii, 366
  - WCAG guidelines 357
- <xmp> tag** 364
- x-small (relative font size)** 107
- xx-large (relative font size)** 107
- xx-small (relative font size)** 107

## Y

- Yahoo! User Interface (YUI) library** 295
- y coordinates** 304
- yen symbol (¥)** 56
- YouTube site** 272

# About the Author

## **Faithe Wempen, M.A.**



Faithe is an adjunct instructor of Computer Technology at Indiana University/Purdue University at Indianapolis and the author of many books on PC hardware and software. She also develops Web sites for businesses and non-profit organizations, and writes and teaches online computer training courses for corporate clients.

# What do you think of this book?

We want to hear from you!

To participate in a brief online survey, please visit:

[microsoft.com/learning/booksurvey](https://microsoft.com/learning/booksurvey)

Tell us how well this book meets your needs—what works effectively, and what we can do better. Your feedback will help us continually improve our books and learning resources for you.

Thank you in advance for your input!

**Microsoft**  
Press

## Stay in touch!

To subscribe to the *Microsoft Press® Book Connection Newsletter*—for news on upcoming books, events, and special offers—please visit:

[microsoft.com/learning/books/newsletter](https://microsoft.com/learning/books/newsletter)