

“‘实战手册’系列就是最具智慧和最有用的指南……”

—— Kevin Kelly, 《Wired》杂志联合创办人

CSS3

CSS3实战手册（影印版）

the missing manual®

The book that should have been in the box®

全面修订
第三版
涵盖HTML5



O'REILLY® 东南大学出版社

David Sawyer McFarland 著

CSS3实战手册 (影印版)

CSS3: the missing manual®

David Sawyer McFarland 著

O'REILLY®

arnham | Köln | Sebastopol | Tokyo

Media, Inc. 授权东南大学出版社出版

南京 东南大学出版社

图书在版编目 (CIP) 数据

CSS3 实战手册: 第3版: 英文 / (美) 麦克法兰
(McFarland, D.S.) 著. —影印本. —南京: 东南大学出版社,
2013.5

书名原文: CSS3: The Missing Manual, 3E

ISBN 978-7-5641-4199-8

I. ① C… II. ① 麦… III. ① 网页制作工具—英文
文 IV. ① TP393.092

中国版本图书馆CIP数据核字 (2013) 第 097344 号

江苏省版权局著作权合同登记

图字: 10-2013-126 号

©2012 by O'Reilly Media, Inc.

Reprint of the English Edition, jointly published by O'Reilly Media, Inc. and Southeast University Press, 2013. Authorized reprint of the original English edition, 2013 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

英文原版由 O'Reilly Media, Inc. 出版 2012。

英文影印版由东南大学出版社出版 2013。此影印版的出版和销售得到出版权和销售权的所有者——O'Reilly Media, Inc. 的许可。

版权所有, 未得书面许可, 本书的任何部分和全部不得以任何形式重制。

CSS3 实战手册 第三版 (影印版)

出版发行: 东南大学出版社

地 址: 南京四牌楼2号 邮编: 210096

出 版 人: 江建中

网 址: <http://www.seupress.com>

电子邮件: press@seupress.com

印 刷: 扬中市印刷有限公司

开 本: 787 毫米 × 980 毫米 16 开本

印 张: 40.75

字 数: 798 千字

版 次: 2013 年 5 月第 1 版

印 次: 2013 年 5 月第 1 次印刷

书 号: ISBN 978-7-5641-4199-8

定 价: 82.00 元 (册)

本社图书若有印装质量问题, 请直接与营销部联系。电话 (传真): 025-83791830

The Missing Credits

ABOUT THE AUTHOR



David Sawyer McFarland is president of Sawyer McFarland Media, Inc., a web development and training company in Portland, Oregon. He's been building websites since 1995, when he designed his first website: an online magazine for communication professionals. He's served as the webmaster at the University of California at Berkeley and the Berkeley Multimedia Research Center, and he has helped build, design, and program numerous websites for clients including

Macworld.com.

In addition to building websites, David is also a writer, trainer, and instructor. He's taught web design at the UC Berkeley Graduate School of Journalism, the Center for Electronic Art, the Academy of Art College, Ex'Pressions Center for New Media, and the Art Institute of Portland. He currently teaches in the Multimedia Program at Portland State University. He's written articles about web design for *Practical Web Design*, *Macworld* magazine, and *CreativePro.com*.

David is also the author of *Dreamweaver: The Missing Manual* and *JavaScript: The Missing Manual*.

He welcomes feedback about this book by email: missing@sawmac.com. (If you're seeking technical help, however, please refer to the sources listed in Appendix B.)

ABOUT THE CREATIVE TEAM

Nan Barber (editor) has worked with the Missing Manual series since the previous century. She lives in Massachusetts with her husband and iMac. Email: nanbarber@oreilly.com.

Holly Bauer (production editor) lives in Ye Olde Cambridge, Mass., where she is an avid home cook, prolific DIYer, and mid-century modern design enthusiast. Email: holly@oreilly.com.

Nancy Reinhardt (proofreader) lives in the Midwest, where she enjoys summer weekends at the lake, boating, swimming, and reading voraciously. Nan is not only a freelance copyeditor and proofreader, but she's also a published romance novelist. Check out her work at www.nanreinhardt.com. Email: nanleigh1@gmail.com.

Nancy A. Guenther (indexer) indexed this book on behalf of Potomac Indexing, LLC, an international indexing partnership at www.potomacindexing.com. She has been a full-time freelance indexer since 1983, specializing in computer software, American studies, and business. Her website is www.guenther.bizland.com.

Daniel J. Quinn (technical reviewer) is a freelance web developer at DQuinn.net. For the past five years, he has worked as a senior UI engineer at award-winning digital agency Genuine Interactive, specializing in WordPress and content strategy for brands like Sam Adams, MassMutual, and Children's Hospital Boston. Today, Daniel serves as web developer for Harvard University's Digital Communications department. He regularly works with a team of local designers and can be reached at daniel@dquinn.net.

Jason Arnold (technical reviewer) lives in Santa Rosa, California, with his wife and three daughters. He works at Healdsburg District Hospital as a Telemetry Technician and teaches Kenpo Karate to children. He is currently working toward his nursing degree at Santa Rosa Junior College. In his free time, he does Kenpo Karate with his daughters and is always on the lookout for an extra tech project to stay busy.

ACKNOWLEDGEMENTS

Many thanks to all those who helped with this book, including my students, who always help me see complex concepts through beginners' eyes. Thanks to my technical editors, Daniel Quinn and Jason Arnold, who saved me from any embarrassing mistakes, and Zoe Gillenwater whose valuable advice for the first edition of this book lives on. Also, we all owe a big debt of gratitude to the many web designers who have broken new ground by using CSS in creative ways and shared their discoveries with the web design community.

Finally, thanks to David Pogue, whose unflagging enthusiasm and endurance is inspiring; Nan Barber for refining my writing, fixing my mistakes, and keeping me on track; my wife, Scholle, for her love and support; my son, Graham, who suggested that I'd get this book done a lot faster if I just typed "Blah, blah, blah, blah, BOO!" for each chapter; my wonderful daughter, Kate, whose smile is always a great pick-me-up; and to my family: Mom, Doug, Mary, David, Marisa, Tessa, Phyllis, Les, Del, Patricia, and Mike.

—David Sawyer McFarland

THE MISSING MANUAL SERIES

Missing Manuals are witty, superbly written guides to computer products that don't come with printed manuals (which is just about all of them). Each book features a handcrafted index and cross-references to specific pages (not just chapters). Recent and upcoming titles include:

Access 2010: The Missing Manual by Matthew MacDonald

Access 2013: The Missing Manual by Matthew MacDonald

Adobe Edge Animate: The Missing Manual by Chris Grover

Buying a Home: The Missing Manual by Nancy Conner

Creating a Website: The Missing Manual, Third Edition, by Matthew MacDonald

David Pogue's Digital Photography: The Missing Manual by David Pogue

Dreamweaver CS6: The Missing Manual by David Sawyer McFarland

Droid 2: The Missing Manual by Preston Gralla

Droid X2: The Missing Manual by Preston Gralla

Excel 2010: The Missing Manual by Matthew MacDonald

FileMaker Pro 12: The Missing Manual by Susan Prosser and Stuart Gripman

Flash CS6: The Missing Manual by Chris Grover

Galaxy S II: The Missing Manual by Preston Gralla

Galaxy Tab: The Missing Manual by Preston Gralla

Google+: The Missing Manual by Kevin Purdy

Google SketchUp: The Missing Manual by Chris Grover

HTML5: The Missing Manual by Matthew MacDonald

iMovie '11 & iDVD: The Missing Manual by David Pogue and Aaron Miller

iPad: The Missing Manual, Fourth Edition by J.D. Biersdorfer

iPhone: The Missing Manual, Fifth Edition by David Pogue

iPhone App Development: The Missing Manual by Craig Hockenberry

iPhoto '11: The Missing Manual by David Pogue and Lesa Snider

iPod: The Missing Manual, Tenth Edition by J.D. Biersdorfer and David Pogue

JavaScript & jQuery: The Missing Manual, Second Edition by David Sawyer McFarland

Kindle Fire: The Missing Manual by Peter Meyers

Living Green: The Missing Manual by Nancy Conner

Mac OS X Mountain Lion: The Missing Manual by David Pogue

Microsoft Project 2010: The Missing Manual by Bonnie Biafore

Motorola Xoom: The Missing Manual by Preston Gralla

Netbooks: The Missing Manual by J.D. Biersdorfer

NOOK HD: The Missing Manual by Preston Gralla

Office 2010: The Missing Manual by Nancy Conner, Chris Grover, and Matthew MacDonald

Office 2011 for Macintosh: The Missing Manual by Chris Grover

Office 2013: The Missing Manual by Nancy Conner and Matthew MacDonald

Personal Investing: The Missing Manual by Bonnie Biafore

Photoshop CS6: The Missing Manual by Lesa Snider

Photoshop Elements 10: The Missing Manual by Barbara Brundage

PHP & MySQL: The Missing Manual, Second Edition by Brett McLaughlin

QuickBooks 2012: The Missing Manual by Bonnie Biafore

Switching to the Mac: The Missing Manual, Lion Edition by David Pogue

Windows 7: The Missing Manual by David Pogue

Windows 8: The Missing Manual by David Pogue

Your Body: The Missing Manual by Matthew MacDonald

Your Brain: The Missing Manual by Matthew MacDonald

Your Money: The Missing Manual by J.D. Roth

For a full list of all Missing Manuals in print, go to www.missingmanuals.com/library.html.

Contents

The Missing Credits	vii
----------------------------------	------------

Introduction	1
---------------------------	----------

Part One: **CSS Basics**

CHAPTER 1: HTML for CSS	17
--------------------------------------	-----------

HTML: Past and Present	17
------------------------------	----

Writing HTML for CSS	20
----------------------------	----

The Importance of the Doctype	30
-------------------------------------	----

Making Sure Internet Explorer Is Up-to-Date	32
---	----

CHAPTER 2: Creating Styles and Style Sheets	35
--	-----------

Anatomy of a Style	35
--------------------------	----

Understanding Style Sheets	38
----------------------------------	----

Internal Style Sheets	40
-----------------------------	----

External Style Sheets	41
-----------------------------	----

Tutorial: Creating Your First Styles	43
--	----

CHAPTER 3: Selectors: Identifying What to Style	55
--	-----------

Tag Selectors: Page-Wide Styling	56
--	----

Class Selectors: Pinpoint Control	57
---	----

ID Selectors: Specific Page Elements	60
--	----

Styling Groups of Tags	62
------------------------------	----

Styling Tags Within Tags	63
--------------------------------	----

Pseudo-Classes and Pseudo-Elements	68
--	----

Attribute Selectors	72
---------------------------	----

Child Selectors	74
-----------------------	----

Siblings	79
----------------	----

The :not() Selector	80
----------------------------	----

Tutorial: Selector Sampler	81
----------------------------------	----

CHAPTER 4: Saving Time with Style Inheritance	93
--	-----------

What Is Inheritance?	93
----------------------------	----

How Inheritance Streamlines Style Sheets	95
--	----

The Limits of Inheritance	95
---------------------------------	----

Tutorial: Inheritance	97
-----------------------------	----

CHAPTER 5:	Managing Multiple Styles: The Cascade	103
	How Styles Cascade	104
	Specificity: Which Style Wins	108
	Controlling the Cascade	111
	Tutorial: The Cascade in Action	117

Part Two: **Applied CSS**

CHAPTER 6:	Formatting Text	127
	Using Fonts	127
	Using Web Fonts	132
	Discovering Google Web Fonts	146
	Adding Color to Text	154
	Changing Font Size	157
	Formatting Words and Letters	162
	Adding Text Shadow	166
	Formatting Entire Paragraphs	167
	Styling Lists	173
	Tutorial: Text Formatting in Action	178

CHAPTER 7:	Margins, Padding, and Borders	193
	Understanding the Box Model	193
	Controlling Space with Margins and Padding	195
	Adding Borders	202
	Coloring the Background	206
	Creating Rounded Corners	207
	Adding Drop Shadows	210
	Determining Height and Width	212
	Wrapping Content with Floating Elements	220
	Tutorial: Margins, Backgrounds, and Borders	225

CHAPTER 8:	Adding Graphics to Web Pages	239
	Discovering CSS and the Tag	239
	Adding Background Images	240
	Controlling Repetition	244
	Positioning a Background Image	245
	Using Background Property Shorthand	255
	Using Multiple Background Images	257
	Utilizing Gradient Backgrounds	259
	Applying Easy Gradients with Colorzilla	269
	Tutorial: Enhancing Images	272
	Tutorial: Creating a Photo Gallery	278
	Tutorial: Using Background Images	283

CHAPTER 9:	Sprucing Up Your Site's Navigation	293
	Selecting Which Links to Style	293
	Styling Links	297
	Building Navigation Bars	303
	CSS-Style Preloading Rollovers	313
	Styling Particular Types of Links	315
	Tutorial: Styling Links	318
	Tutorial: Creating a Navigation Bar	324
CHAPTER 10:	CSS Transforms, Transitions, and Animations	335
	Transforms	335
	Transitions	345
	Animations	354
	Tutorial	366
CHAPTER 11:	Formatting Tables and Forms	375
	Using Tables the Right Way	375
	Styling Tables	378
	Styling Forms	384
	Tutorial: Styling a Table	390
	Tutorial: Styling a Form	395
Part Three:	CSS Page Layout	
CHAPTER 12:	Introducing CSS Layout	405
	Types of Web Page Layouts	405
	How CSS Layout Works	408
	Layout Strategies	412
CHAPTER 13:	Building Float-Based Layouts	417
	Applying Floats to Your Layouts	421
	Overcoming Float Problems	425
	Tutorial: Multiple Column Layouts	440
CHAPTER 14:	Responsive Web Design	453
	Responsive Web Design Basics	454
	Setting Up a Web Page for RWD	455
	Media Queries	457
	Flexible Grids	465
	Fluid Images	470
	Responsive Web Design Tutorial	474

CHAPTER 15:	Positioning Elements on a Web Page	487
	How Positioning Properties Work	488
	Powerful Positioning Strategies	502
	Tutorial: Positioning Page Elements	507

Part Four: **Advanced CSS**

CHAPTER 16:	CSS for the Printed Page	519
	How Media Style Sheets Work	519
	How to Add Media Style Sheets	522
	Creating Print Style Sheets	523
	Tutorial: Building a Print Style Sheet	530
CHAPTER 17:	Improving Your CSS Habits	539
	Adding Comments	539
	Organizing Styles and Style Sheets	541
	Eliminating Browser Style Interference	548
	Using Descendent Selectors	553
	Trying Different CSS for Internet Explorer	558

Part Five: **Appendixes**

APPENDIX A:	CSS Property Reference	563
	CSS Values	563
	Text Properties	567
	List Properties	572
	Padding, Borders, and Margins	573
	Backgrounds	579
	Page Layout Properties	582
	Animation, Transform, and Transition Properties	588
	Table Properties	593
	Miscellaneous Properties	595
APPENDIX B:	CSS Resources	599
	References	599
	CSS Help	600
	CSS Tips, Tricks, and Advice	600
	CSS Navigation	601
	CSS Layout	602
	Showcase Sites	603
	CSS Books	604
	CSS Software	604
	Index	607

Introduction

Cascading Style Sheets—CSS for short—give you creative control over the layout and design of your web pages. Using CSS, you can dress up your site's text with eye-catching headlines, drop caps, and borders, just like the ones in glossy magazines. You can also arrange images with precision, create columns and banners, and highlight your links with dynamic rollover effects. You can even make elements fade in or out of view, move objects around the page, or make a button slowly change colors when a visitor mouses over it.

Anything that can do all that must be pretty complicated, right? Au contraire! The whole idea behind CSS is to streamline the process of styling web pages. In the next few pages, you'll learn about the basics of CSS. In Chapter 1, you'll get right to work creating a CSS-powered web page.

■ How CSS Works

CSS works with HTML, but it's not HTML. It's a different language altogether. While HTML provides structure to a document by organizing information into headers, paragraphs, bulleted lists, and so on, CSS works hand-in-hand with the web browser to make HTML *look* good.

For example, you might use HTML to turn a phrase into a top-level heading, indicating that it introduces the content on the rest of the page. However, you'd use CSS to format that heading with, say, big and bold red type and position it 50 pixels from the left edge of the window. In CSS, that text formatting comprises a style—a rule describing the appearance of a particular portion of a web page. A style *sheet* is a set of these styles.

You can also create styles specifically for working with images. For instance, a style can align an image along the right edge of a web page, surround the image with a colorful border, and place a 50-pixel margin between the image and the surrounding text.

Once you've created a style, you can apply it to text, images, headings, or other elements on a page. For example, you can select a paragraph of text and apply a style to instantly change the text's size, color, and font. You can also create styles for specific HTML tags, so, for example, all first-level headings (`<h1>` tags) in your site are displayed in the same style, no matter where they appear.

■ The Benefits of CSS

Before CSS, web designers were limited to the layout and styling options of HTML. And if you surfed the Web in 1995, then you understand the emphasis on *limited*. HTML still forms the foundation of all pages on the World Wide Web, but it's simply not a design tool. Sure, HTML provides basic formatting options for text, images, tables, and other web page elements, and patient, meticulous webmasters can make pages look pretty good using only HTML. But the result is often sluggish web pages laden with clunky code.

CSS, in contrast, offers the following advantages:

- Style sheets offer far more formatting choices than HTML. With CSS, you can format paragraphs as they appear in a magazine or newspaper (the first line indented and no space between each paragraph, for example) and control the *leading* (the space between lines of type in a paragraph).
- When you use CSS to add a background image to a page, you get to decide whether and how it *tiles* (repeats). HTML can't even begin to do that.
- Even better, CSS styles take up much less space than HTML's formatting options, such as the much-hated `` tag. You can usually trim a lot of kilobytes from text-heavy web pages by using CSS. As a result, your pages look great and load faster.
- Style sheets also make updating your site easier. You can collect all your styles into a single external style sheet that's linked to every page in your site. Then, when you edit a style, that change immediately ripples through your site *wherever* that style appears. You can completely change the appearance of a site just by editing a single style sheet.

■ What You Need to Know

This book assumes you've already got some knowledge of HTML (and maybe some CSS experience as well). Perhaps you've built a site or two (or at least a page or two) and have some familiarity with the sea of tags—`<html>`, `<p>`, `<h1>`, `<table>`,

and so on—that make up the Hypertext Markup Language. CSS can't do anything without HTML, so to move forward you need to know how to create a web page by using basic HTML.

If you've used HTML in the past to create web pages, but feel like your knowledge is a bit rusty, the next section provides a basic refresher.

TIP If you're just getting your feet wet learning HTML, then check out these free online tutorials: HTML Dog (www.htmldog.com/guides/htmlbeginner) and W3Schools (www.w3schools.com/html/). If you're a printed page fan, then you may want to pick up a copy of *Creating a Website: The Missing Manual*, Third Edition or *Head First HTML and CSS*, Second Edition (both O'Reilly).

■ HTML: The Barebones Structure

HTML (Hypertext Markup Language) uses simple commands called *tags* to define the various parts of a web page. For example, this HTML code creates a simple web page:

```
<!doctype html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Hey, I am the title of this web page</title>
  </head>
  <body>
    <p>Hey, I am a paragraph on this web page.</p>
  </body>
</html>
```

It may not be exciting, but this example has all the basic elements a web page needs. You'll notice something called a *doctype* declaration at the very beginning of the code, followed by `<html>` (with the brackets), a head, a body, and some stuff—the actual page contents—inside the body, ending in a final `</html>`.

■ Document Types

All web pages begin with a doctype—a line of code that identifies what flavor of HTML you used to write the page. Two doctypes have been used for years—HTML 4.01 and XHTML 1.0—and each of those doctypes has two styles: *strict* and *transitional*. For example, the HTML 4.01 transitional doctype looks like the following (the other doctypes for HTML 4.01 and XHTML 1.0 look similar):

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

NOTE For examples of all various doctypes, visit www.webstandards.org/learn/reference/templates.

If you look at the code for the sample HTML page in this section, you'll see that it uses a much more succinct doctype:

```
<!doctype html>
```

That's HTML5's new—and simpler—doctype. HTML5 is intended to be easier to use and more streamlined than its predecessors. This book uses the HTML5 doctype, which is supported by every popular browser (even the old Internet Explorer 6). And because HTML5 is the future of the Web, there's no reason to use the older doctypes any longer.

NOTE

Just because the HTML doctype works in older browsers doesn't mean that those browsers understand all HTML5 tags or features. Internet Explorer 8 and earlier, for example, don't recognize the new HTML5 tags. To style tags with CSS for those versions of IE, you have to employ a little JavaScript. You'll learn how to get older browsers up to speed on page 30.

But no matter which doctype you prefer, the important thing is to always use one. Without it, your pages will look different depending on your visitor's browser, since browsers display CSS differently if they don't have a doctype for guidance.

Each doctype requires you to write your HTML in a certain way. For example, the tag for a line break looks like this in HTML 4.01:

```
<br>
```

But in XHTML, it looks like this:

```
<br />
```

And there's another advantage of HTML5: it accepts either one.

■ How HTML Tags Work

In the simple HTML example on page 3, as in the HTML code of any web page, most commands appear in pairs that surround a block of text or other commands. Sandwiched between brackets, these *tags* are instructions that tell a web browser how to display the web page. Tags are the “markup” part of the Hypertext Markup Language.

The starting (*opening*) tag of each pair tells the browser where the instruction begins, and the ending tag tells it where the instruction ends. Ending or *closing* tags always include a forward slash (/) after the first bracket symbol (<).

On any web page, you'll usually find at least these four elements.

- The first line of a web page is the **DOCTYPE declaration**, discussed in the previous section.
- The **<html>** tag appears once at the beginning of a web page and again (with an added forward slash) at the end: **</html>**. This tag tells a web browser that

the information contained in this document is written in HTML, as opposed to some other language. All the contents of a page, including other tags, appear between the opening and closing `<html>` tags.

If you were to think of a web page as a tree, the `<html>` tag would be its roots. Springing from the trunk are two branches that represent the two main parts of any web page: the *head* and the *body*.

- The *head* of a web page contains the title of the page (“Izzie’s Mail-Order Pencils”). It may also include other, invisible information, like a page description, that browsers and search engines use. You surround the head section with opening and closing `<head>` tags.

In addition, the head section can include information that browsers use to format the page’s HTML and to add interactivity. As you’ll see, the `<head>` section can contain CSS code (like the kind you’ll learn to write in this book) or a link to another file containing CSS information.

- The *body*, as set apart by its surrounding `<body>` tags, contains all the content that appears inside a browser window—headlines, text, pictures, and so on.

Within the `<body>` tag, you commonly find tags like these:

- You tell a web browser where a paragraph of text begins with a `<p>` (opening paragraph tag), and where it ends with a `</p>` (closing paragraph tag).
- The `` tag emphasizes text. When you surround some text with it and its partner tag, ``, you get boldface type. The HTML snippet `Warning!` tells a web browser to strongly emphasize the word “Warning!”
- The `<a>` tag, or anchor tag, creates a *hyperlink* in a web page. When clicked, a hyperlink—or *link*—can lead anywhere on the Web. You tell the browser where the link points by putting a web address inside the `<a>` tags. For instance, you can type `Click here!`.

The browser knows that when your visitor clicks the words “Click here!” it should go to the Missing Manual website. The `href` part of the tag is called an *attribute*, and the URL (the Uniform Resource Locator or web address) is the *value*. In this example, `http://www.missingmanuals.com` is the *value* of the `href` attribute.

■ XHTML, Too

Like any technology, HTML has evolved over time. Although standard HTML has served its purpose well, it’s always been a somewhat sloppy language. Among other things, it lets you use uppercase, lowercase, and mixed-case letters in tags (`<body>` and `<BODY>` are both correct, for example) and permits unclosed tags (so you can use a `<p>` tag without a closing `</p>` tag to create a paragraph). While this flexibility may make page-writing easier, it also makes life more difficult for web browsers, smart phones, and other technologies that interact with data on the Web.

Additionally, HTML doesn't work with one of the hottest Internet languages: XML, or Extensible Markup Language.

To keep pace with the times, an improved version of HTML, called XHTML, was introduced back in 2000, and you'll find it used frequently on many sites (in fact, XHTML is just an "XML-ified" version of HTML). XHTML was seen as the future (back in 2000), but has since been supplanted by HTML5. While web browsers still understand XHTML (and probably will for a long time), this book doesn't include it.

■ HTML5: The Wheel Turns Again

When the W3C—the group responsible for many Internet technologies—introduced XHTML as an intermediate step in the transition to XML as the prime language of the Web, the web development community heralded it as the next big thing. History has shown that prediction to be a bit grandiose. As it turns out, the complexity of moving to XML kept browser manufacturers from following the XML path the World Wide Web Consortium laid down. Instead, the W3C closed down the XHTML working group in favor of HTML5—a new version of HTML that moves away from XML and back to HTML, albeit an enhanced version. With the support of all browser manufacturers, HTML5 is the new next big thing. Some browsers, including Google Chrome and Firefox, already recognize parts of HTML5, and even mainstream news outlets like Time, Newsweek, and CNN drop "HTML5" into their technology reporting.

HTML5 isn't some radically new technology. In fact, unlike XHTML, which was intended to foster a new way to build web pages, HTML5 is about making sure the Web continues to work the way it always has. Most of the basics of HTML are still in place. Also, HTML5 adds a few new elements, meant to support the way web designers currently build websites. For example, in HTML5, the `<header>` tag can contain the content you'd usually find at the top of a page, such as a logo and site-wide navigation links; the new `<nav>` tag encloses the set of links used to navigate a site; and the `<footer>` tag houses the stuff you usually put at the bottom of a page, like legal notices, email contacts, and so on.

In addition, HTML5 adds new tags that let you insert video and audio into a page, and new form tags that add sophisticated elements, like sliders and pop-up date pickers, as well as built-in browser support for form validation (which ensures visitors correctly fill out your forms). Unfortunately, browser support for these new features isn't consistent, and it's difficult to use the new tags without some pretty elaborate workarounds.

Although HTML5 may not be entirely ready for primetime, there's no reason not to start using the HTML5 doctype and even some basic HTML5 now. (You'll need to give Internet Explorer 8 and earlier a bit of help, though; see the box on page 24.) You'll find HTML5 throughout this book, especially in the next chapter.