

O'REILLY®

"The Missing Manual series is simply the most intelligent and usable series of guidebooks..."
—KEVIN KELLY, CO-FOUNDER OF WIRED

CSS

the missing manual®

The book that should have been in the box®

4th
Edition



David Sawyer McFarland

CSS: The Missing Manual

David Sawyer McFarland



Beijing • Cambridge • Farnham • Köln • Sebastopol • Tokyo

Special Upgrade Offer

If you purchased this ebook directly from oreilly.com, you have the following benefits:

- DRM-free ebooks—use your ebooks across devices without restrictions or limitations
- Multiple formats—use on your laptop, tablet, or phone
- Lifetime access, with free updates
- Dropbox syncing—your files, anywhere

If you purchased this ebook from another retailer, you can upgrade your ebook to take advantage of all these benefits for just \$4.99. [Click here](#) to access your ebook upgrade.

Please note that upgrade offers are not available from sample content.

The Missing Credits

About the Author



David McFarland (author) is a web developer, teacher and author. He's been building web sites since 1995, when he designed his first website: an online magazine for communication professionals.

He's taught web design at the UC Berkeley Graduate School of Journalism, the Center for Electronic Art, the Art Institute of Portland, and Portland State University.

He's currently a Teaching Team Leader at the online education site, Treehouse (<http://teamtreehouse.com>).

About the Creative Team

Nan Barber (editor) is associate editor for the Missing Manuals series. She lives in Massachusetts with her husband and various electronic devices. Email: nanbarber@gmail.com.

Melanie Yarbrough (production editor and compositor) works and plays in Cambridge, Massachusetts, where she bakes up whatever she can imagine and bikes around the city. Email: myarbrough@oreilly.com.

Molly Ives Brower (proofreader) is a freelance editor and proofreader who has loved the Internet since she got a BITNET address in 1990. These days, though, she can be found online at <http://www.mjibrower.com> and on Twitter, where she goes by @vintagereader. Email: molly@mjibrower.com.

Ron Strauss(indexer) specializes in the indexing of information technology publications of all kinds. Ron is also an accomplished classical violist and lives in Northern California with his wife and fellow indexer, Annie, and his miniature pinscher, Kanga. Email: rstrauss@mchsi.com.

Rich Koster (beta reader) bought his first Mac, a 17-inch MacBook Pro, in 2009, and has never looked back toward the Dark Side (PCs). Rich served as the tech editor of David Pogue's *iPhone: The Missing Manual*, 3rd Edition. He's a husband, a father,

and creator of the Disney Echo at <http://DisneyEcho.emuck.com>, which he has fun tending daily with his MacBook Pro!

Acknowledgments

Many thanks to all those who helped with this book, including the many students I've taught who always help me see complex concepts through beginner's eyes. Thanks to my technical editors, Daniel Quinn and Jennifer Davis, who saved me from embarrassing mistakes. 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.

Thanks to David Pogue, who got me started, many years ago on this long adventure. Thanks to Nan Barber for refining my writing, fixing my mistakes, and keeping me on track.

—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 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
- *CSS3: The Missing Manual, Third Edition* by David Sawyer McFarland
- *Dreamweaver CS6: The Missing Manual* by David Sawyer McFarland
- *Dreamweaver CC: The Missing Manual, Second Edition* by David Sawyer McFarland and Chris Grover
- *Excel 2013: The Missing Manual* by Matthew MacDonald
- *FileMaker Pro 13: The Missing Manual* by Susan Prosser and Stuart Gripman
- *Fire Phone: The Missing Manual* by Preston Gralla
- *Flash CS6: The Missing Manual* by Chris Grover
- *Galaxy Tab: The Missing Manual* by Preston Gralla
- *Galaxy S5: The Missing Manual* by Preston Gralla
- *Google+: The Missing Manual* by Kevin Purdy
- *HTML5: The Missing Manual, Second Edition* by Matthew MacDonald
- *iMovie: The Missing Manual* by David Pogue and Aaron Miller

- *iPad: The Missing Manual, Seventh Edition* by J.D. Biersdorfer
- *iPhone: The Missing Manual, Eighth Edition* by David Pogue
- *iPhone App Development: The Missing Manual* by Craig Hockenberry
- *iPhoto: The Missing Manual* by David Pogue and Lesa Snider
- *iPod: The Missing Manual, Eleventh Edition* by J.D. Biersdorfer and David Pogue
- *iWork: The Missing Manual* by Jessica Thornsby and Josh Clark
- *JavaScript & jQuery: The Missing Manual, Third Edition* by David Sawyer McFarland
- *Kindle Fire HD: The Missing Manual* by Peter Meyers
- *Living Green: The Missing Manual* by Nancy Conner
- *Microsoft Project 2013: The Missing Manual* by Bonnie Biafore
- *Motorola Xoom: The Missing Manual* by Preston Gralla
- *NOOK HD: The Missing Manual* by Preston Gralla
- *Office 2011 for Macintosh: The Missing Manual* by Chris Grover
- *Office 2013: The Missing Manual* by Nancy Conner and Matthew MacDonald
- *OS X Mavericks: The Missing Manual* by David Pogue
- *OS X Yosemite: The Missing Manual* by David Pogue
- *Personal Investing: The Missing Manual* by Bonnie Biafore
- *Photoshop CS6: The Missing Manual* by Lesa Snider
- *Photoshop CC: The Missing Manual, Second Edition* by Lesa Snider
- *Photoshop Elements 13: The Missing Manual* by Barbara Brundage
- *PHP & MySQL: The Missing Manual, Second Edition* by Brett McLaughlin
- *QuickBooks 2015: The Missing Manual* by Bonnie Biafore
- *Switching to the Mac: The Missing Manual, Mavericks Edition* by David Pogue
- *Windows 7: The Missing Manual* by David Pogue
- *Windows 8: The Missing Manual* by David Pogue
- *WordPress: The Missing Manual, Second Edition* by Matthew MacDonald
- *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.

Introduction

Cascading Style Sheets—CSS for short—give you creative control over the layout and design of your web pages. With CSS, dressing up your site’s text with eye-catching headlines, drop caps, and borders is just the beginning. 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.

What Is CSS?

CSS is a styling language. You use it to make HTML—the fundamental language of all web pages—look good. Well, hopefully, you’ll use CSS to make your web pages look *better* than good. After you read this book, you’ll be able to make your web pages beautiful, functional, and easy to use.

Think of HTML as the basic structure of your content, and CSS as a designer who takes your plain HTML and spruces it up with a fancy font, a border with rounded corners, or a bright red background.

But before you start learning about CSS, you need to understand HTML.

What You Need to Know

This book assumes you’ve already got some knowledge of HTML. 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>`—that make up the Hypertext Markup Language. CSS can’t do anything without HTML, so 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 *HTML5: 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 the HTML5 doctype. HTML5 is 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). There's no reason to use any doctype other than the simple HTML5 doctype.

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 in the box on [Getting IE8 to Understand HTML5](#).

The most important thing about a doctype, however, 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 [How HTML Tags Work](#), 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 root. 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 marks text as important. 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.

HTML5: More Tags to Choose From

HTML5—the current version of HTML—has been around for years now. Sometimes you’ll hear the name used to describe things other than HTML tags, like local storage (a way to save data from a website to a visitor’s computer), geolocation (a way to check where a visitor is in the world), and drawing to the web page using WebGL. Strictly speaking, those technologies aren’t part of HTML, but they’re new browser features that came onto the scene along with HTML5.

In this book, the term *HTML5* always refers to the HTML5 doctype as well as the new tags introduced as part of the HTML5 standard. HTML5 isn’t radically different from its predecessors—it was created to make sure the Web continues to work the way it always has. Most of the basics of HTML are the same as they’ve always been; HTML5 adds a few new elements meant to support the way web designers currently build websites. For example, in HTML5, the `<header>` tag contains the content you usually find at the top of a page, such as a logo and sitewide 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, new form tags that add sophisticated elements like sliders and pop-up date pickers, and built-in browser support for form validation (which ensures visitors correctly fill out your forms). You’ll see HTML5 used throughout this book, especially in the next chapter.

Software for CSS

To create web pages made up of HTML and CSS, you need nothing more than a basic

text editor like Notepad (Windows) or TextEdit (Mac). But after typing a few hundred lines of HTML and CSS, you may want to try a program better suited to working with web pages. This section lists some common programs, some free and some you have to buy.

NOTE

There are literally hundreds of tools that can help you create web pages, so the following isn't a complete list. Think of it as a greatest-hits tour of the most popular programs that CSS fans are using today.

Free Programs

There are plenty of free programs out there for editing web pages and style sheets. If you're still using Notepad or TextEdit, give one of these a try. Here's a short list to get you started:

- **Brackets (Windows, Mac, Linux; <http://brackets.io/>).** Spearheaded by Adobe, this free, open-source text editor has many tools for working with HTML and CSS. It's written specifically for web designers and developers.
- **Atom (Windows, Mac, Linux; <https://atom.io/>).** Another free, open-source text editor created by the people behind GitHub, the hugely popular code sharing and collaboration site. Like Brackets, this new text editor is aimed at web developers.
- **jEdit (Windows, Mac, Linux; <http://jedit.org>).** This free, Java-based text editor works on almost any computer and includes many features that you'd find in commercial text editors, like syntax highlighting for CSS.
- **Notepad++ (Windows; <http://notepad-plus.sourceforge.net>).** A lot of people swear by this fast text editor. It even has built-in features that make it ideal for writing HTML and CSS, like syntax highlighting—color-coding tags and special keywords to make it easier to identify the page's HTML and CSS elements.

Commercial Software

Commercial website development programs range from inexpensive text editors to complete website construction tools with all the bells and whistles:

- **EditPlus (Windows; www.editplus.com)** is an inexpensive (\$35) text editor that includes syntax highlighting, FTP, autocomplete, and other wrist-saving features.
- **skEdit (Mac; www.skedit.com)** is an inexpensive (\$30) web page editor, complete with FTP/SFTP, code hints, and other useful features.
- **Coda2 (Mac; www.panic.com/coda)** is a full-featured web development toolkit (\$99). It includes a text editor, page preview, FTP/SFTP, and graphic CSS-creating tools for creating CSS.
- **Sublime Text (Mac, Windows, Linux; www.sublimetext.com)** is a powerful text editor (\$70) beloved by many web coders. You'll find it frequently used in web design companies.

- **Dreamweaver (Mac and Windows; www.adobe.com/products/dreamweaver)** is a visual web page editor (from \$19.99 per month). It lets you see how your page looks in a web browser. The program also includes a powerful text editor and excellent CSS creation and management tools. Check out *Dreamweaver CC: The Missing Manual* for the full skinny on how to use this powerful program.

NOTE

The various types of software discussed in this section are general-purpose programs that let you edit both HTML and CSS. With them, you need to learn only one program for your web development needs.

About This Book

The World Wide Web is really easy to use. After all, grandfathers in Boise and first graders in Tallahassee log onto the Web every day. Unfortunately, the rules that govern how the Web *works* aren't so easy to understand. The computer scientists and other techie types who write the official documentation aren't interested in explaining their concepts to the average Joe (or Joanne). Just check out www.w3.org/TR/css3-transforms to get a taste of the technical mumbo-jumbo these geeks speak.

People just learning CSS often don't know where to begin. And CSS's finer points can trip up even seasoned web pros. The purpose of this book is to serve as the manual that should have come with CSS. In its pages, you'll find step-by-step instructions for using CSS to create beautiful web pages.

CSS: The Missing Manual is designed to help readers at every technical level. To get the most out of this book, you should know the basics of HTML and maybe even a sampling of CSS. If you've never built a web page before, then check out the tutorial that starts on **Tutorial: Creating Your First Styles**. The primary discussions in these chapters are written for advanced beginners or intermediates. But if you're new to building web pages, special boxes labeled "Up to Speed" provide the introductory information you need to understand the topic at hand. If you're an advanced web jockey, on the other hand, then keep your eye out for similar boxes called "Power Users' Clinic." They offer more technical tips, tricks, and shortcuts for the experienced computer fan.

About the Outline

CSS: The Missing Manual is divided into five parts; the first four each contain several chapters while the last part contains appendixes.

- **Part 1**, shows you how to create style sheets and provides an overview of key CSS concepts like *inheritance*, *selectors*, and the *cascade*. Along the way, you'll learn the best HTML writing practices when working with CSS. Tutorials reinforce the part's main concepts and give you a good taste of the power of CSS.
- **Part 2**, takes you into the real world of web design. You'll learn the most important CSS properties and how to use them to format text, create useful navigation tools, and enhance your page with graphics. You'll learn how to create simple animations

with CSS. This section also provides advice on how to make attractive tables and forms.

- **Part 3**, helps you with one of the most confusing, but most rewarding, aspects of CSS—controlling the layout of your web pages. You’ll learn how to create common designs (like two- and three-column layouts) and how to add sidebars, and you’ll learn about *floats* and *positioning*—two common CSS techniques for controlling page layout. You’ll also learn how to craft websites that adapt to look good on desktop, tablet, and mobile browsers, as well as how to use flexbox, a powerful new way of laying out web pages.
- **Part 4**, delves into professional tips for improving your CSS. It also provides an introduction to Sass—a powerful and efficient way of authoring your style sheets.
- **Part 5**, includes two sets of resources. The CSS Property Reference summarizes each CSS property in small, easy-to-digest chunks so you can quickly learn about useful CSS properties you may not have seen before or brush up on what you already know. The second appendix covers tools and resources for creating and using CSS.

The Very Basics

To use this book, and indeed to use a computer, you need to know a few basics. You should be familiar with these terms and concepts:

- **Clicking.** This book gives you three kinds of instructions that require you to use your computer’s mouse or trackpad. To *click* means to point the arrow cursor at something on the screen and then—without moving the cursor at all—to press and release the clicker button on the mouse (or laptop trackpad). A *right-click* is the same thing using the right mouse button. (On a Mac, press Control as you click if you don’t have a right mouse button.)

To *double-click* means to click twice in rapid succession, again without moving the cursor at all. And to *drag* means to move the cursor *while* pressing the button.

When you’re told to *Ctrl-click* something on a PC or *⌘-click* something on the Mac, you click while pressing the Ctrl or ⌘ key.

- **Menus.** The *menus* are the words at the top of your screen or window: File, Edit, and so on. Click one to make a list of commands appear, as though they’re written on a window shade you’ve just pulled down. This book assumes that you know how to open a program, surf the Web, and download files. You should know how to use the Start menu (Windows) or the Dock or the Apple menu (Mac), as well as the Control Panel (Windows) or System Preferences (Mac OS X).
- **Keyboard shortcuts.** Every time you take your hand off the keyboard to move the mouse, you lose time and potentially disrupt your creative flow. That’s why many experienced computer users use keystroke combinations instead of menu commands wherever possible. When you see a shortcut like Ctrl+S (⌘-S) (which saves changes to the current document), it’s telling you to hold down the Ctrl or ⌘ key, and, while it’s down, type the letter S, and then release both keys.

About→These→Arrows

Throughout this book, and throughout the Missing Manual series, you'll find sentences like this one: "Open the System → Library → Fonts folder." That's shorthand for a much longer instruction that directs you to open three nested folders in sequence, like this: "On your hard drive, you'll find a folder called System. Open that. Inside the System folder window is a folder called Library; double-click it to open it. Inside *that* folder is yet another one called Fonts. Double-click to open it, too."

Similarly, this kind of arrow shorthand helps to simplify the business of choosing commands in menus, as shown in [Figure 1](#).

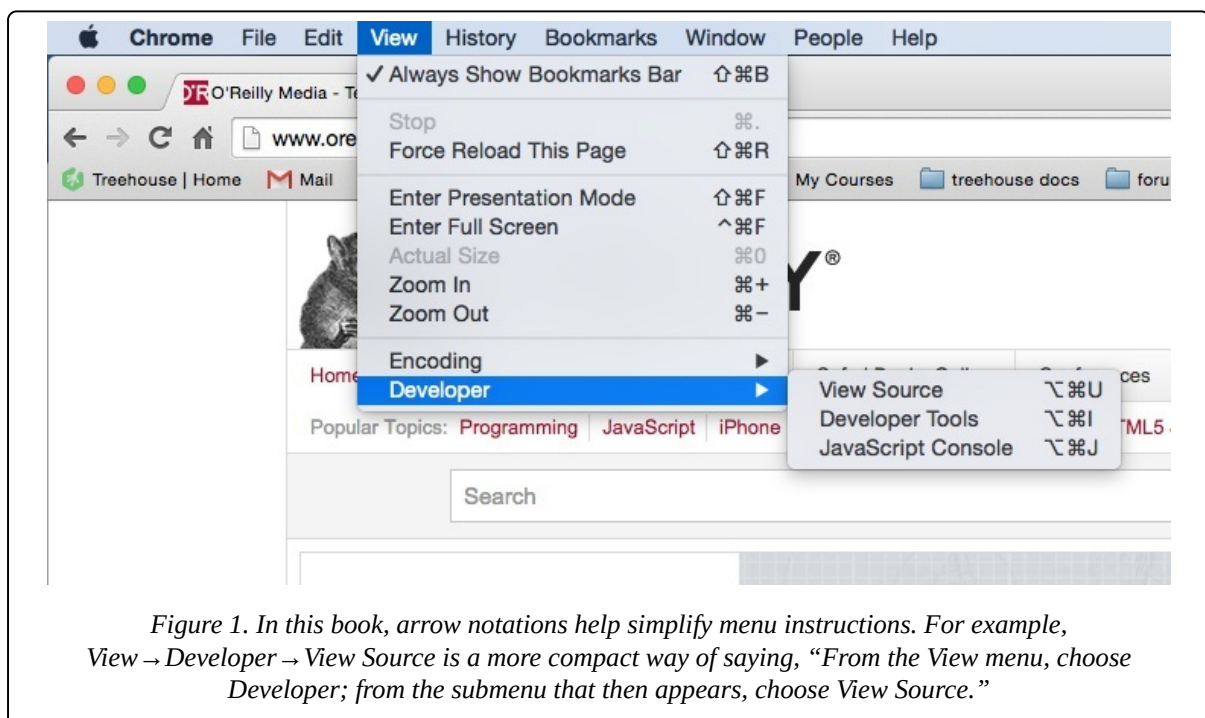


Figure 1. In this book, arrow notations help simplify menu instructions. For example, View → Developer → View Source is a more compact way of saying, "From the View menu, choose Developer; from the submenu that then appears, choose View Source."

About the Online Resources

As the owner of a Missing Manual, you've got more than just a book to read. Online, you'll find example files so you can get some hands-on experience, as well as tips, articles, and maybe even a video or two. You can also communicate with the Missing Manual team and tell us what you love (or hate) about the book. Head over to www.missingmanuals.com, or go directly to one of the following sections.

Living Examples

This book is designed to get your work onto the Web faster and more professionally. It's only natural, then, that half the value of this book lies on the Web.

As you read the book's chapters, you'll encounter a number of *living examples*—step-by-step tutorials that you can build yourself, using raw materials (like graphics and half-completed web pages) that you can download from https://github.com/sawmac/css_mm_4e. You may not gain very much by simply

reading these step-by-step lessons while relaxing in your porch hammock. But if you work through them at the computer, you'll discover that these tutorials give you insight into the way professional designers build web pages.

You'll also find, in this book's lessons, the URLs of the finished pages, so that you can compare your work with the final result. In other words, you won't just see pictures of how the web pages *should* look; you'll find the actual, working web pages on the Internet.

About MissingManuals.com

At www.missingmanuals.com, you'll find articles, tips, and updates to *CSS: The Missing Manual*. In fact, we invite and encourage you to submit such corrections and updates yourself. In an effort to keep the book as up-to-date and accurate as possible, each time we print more copies of this book, we'll make any confirmed corrections you've suggested. We'll also note such changes on the website, so that you can mark important corrections into your own copy of the book, if you like. (Go to www.missingmanuals.com/feedback, choose the book's name from the pop-up menu, and then click Go to see the changes.)

Also on our Feedback page, you can get expert answers to questions that come to you while reading this book, write a book review, and find groups for folks who share your interest in CSS.

We'd love to hear your suggestions for new books in the Missing Manual line. There's a place for that on missingmanuals.com, too. And while you're online, you can also register this book at www.oreilly.com (you can jump directly to the registration page by going here: www.oreilly.com/register). Registering means we can send you updates about this book, and you'll be eligible for special offers like discounts on future editions of *CSS: The Missing Manual*.

Errata

In an effort to keep this book as up-to-date and accurate as possible, each time we print more copies, we'll make any confirmed corrections you've suggested. We also note such changes on the book's website, so you can mark important corrections into your own copy of the book, if you like. Go to www.tinyurl.com/css4e-mm to report an error and view existing corrections.

Safari® Books Online

Safari Books Online is an on-demand digital library that delivers expert content in both book and video form from the world's leading authors in technology and business.

Technology professionals, software developers, web designers, and business and creative professionals use Safari Books Online as their primary resource for research, problem solving, learning, and certification training.

Safari Books Online offers a range of plans and pricing for enterprise, government,

education, and individuals.

Members have access to thousands of books, training videos, and prepublication manuscripts in one fully searchable database from publishers like O'Reilly Media, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit Press, Focal Press, Cisco Press, John Wiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FT Press, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, Course Technology, and hundreds more. For more information about Safari Books Online, please visit us online.

Part 1. CSS Basics

Chapter 1, HTML and CSS

Chapter 2, Creating Styles and Style Sheets

Chapter 3, Selectors: Identifying What to Style

Chapter 4, Saving Time with Style Inheritance

Chapter 5, Managing Multiple Styles: The Cascade

Chapter 1. HTML and CSS

CSS isn't anything without HTML. HTML provides web pages with content and meaningful structure, and while it may not be pretty by itself, the web wouldn't exist without it. So to get the most out of your CSS training, you need to know how to write HTML to create a solid, well-built foundation. This chapter introduces you to the basics of CSS and shows you how to write better, more CSS-friendly HTML.

The good news is that when you use CSS throughout your site, HTML actually becomes *easier* to write. You don't need to try to turn HTML into the design maven it was never intended to be. Instead, CSS offers most of the graphic design touches you'll likely ever want, and HTML pages written to work with CSS are easier to create, since they require less code and less typing. They'll also download faster—a welcome bonus your site's visitors will appreciate (see [Figure 1-1](#)).

HTML: Past and Present

HTML provides the foundation for every page you encounter on the Web. When you add CSS into the mix, HTML becomes simpler, because you don't need to use HTML tags (like the old `` tag) to control how a web page looks. That job is for CSS. But before jumping into CSS, here's a quick walk through HTML's past (and present).

HTML Past: Whatever Looked Good

When a bunch of scientists created the Web to share technical documentation, nobody called in the graphic designers. All the scientists needed HTML to do was structure information for easy comprehension. For example, the `<h1>` tag indicates an important headline, while the `<h2>` tag represents a lesser heading, usually a subheading of the `<h1>` tag. Another favorite, the `` (ordered list) tag, creates a numbered list for things like “Top 10 reasons not to play with jellyfish.”



But as soon as people other than scientists started using HTML, they wanted their web pages to look good. So web designers started to use tags to control appearance rather

than structure information. For example, you can use the `<blockquote>` tag (intended for material that's quoted from another source) on any text that you want to indent a little bit. You can use heading tags to make any text bigger and bolder—regardless of whether it functions as a heading.

In an even more elaborate workaround, designers learned how to use the `<table>` tag to create columns of text and accurately place pictures and text on a page. Unfortunately, since that tag was intended to display spreadsheet-like data—research results, train schedules, and so on—designers had to get creative by using the `<table>` tag in unusual ways, sometimes nesting a table within a table within a table to make their pages look good.

Meanwhile, browser makers introduced new tags and attributes for the specific purpose of making a page look better. The `` tag, for example, let you specify a font color, typeface, and one of seven different sizes. (If you're keeping score at home, that's about 100 fewer sizes than you can get with, say, Microsoft Word.)

Finally, when designers couldn't get exactly what they wanted, they often resorted to using graphics. For example, they'd create a large graphic to capture the exact font and layout for web page elements and then *slice* the Photoshop files into smaller files and piece them back together inside tables to recreate the original design.

While all of the preceding techniques—using tags in creative ways, taking advantage of design-specific tag attributes, and making extensive use of graphics—provide design control over your pages, they also add a lot of additional HTML code. More code makes your site more difficult to build and much slower for your visitors to download.

HTML Present: Scaffolding for CSS

No matter what content your web page holds—the fishing season calendar, driving directions to the nearest IKEA, or pictures from your kid's birthday party—it's the page's design that makes it look like either a professional enterprise or a part-timer's hobby. Good design enhances the message of your site, helps visitors find what they're looking for, and determines how the rest of the world sees your website. That's why web designers went through the contortions described in the previous section to force HTML to look good. By taking on those design duties, CSS lets HTML go back to doing what it does best—structuring content.

Using HTML to control the look of text and other web page elements is obsolete. Don't worry if HTML's `<h1>` tag is too big for your taste or bulleted lists aren't spaced just right. You can take care of that later using CSS. Instead, think of HTML as a method of adding structure to the content you want up on the Web. Use HTML to organize your content and CSS to make that content look great.

Writing HTML for CSS

If you're new to web design, you may need some helpful hints to guide your forays into HTML (and to steer clear of well-intentioned, but out-of-date HTML techniques).