THE EXPERT'S VOICE® IN WEB DEVELOPMENT

Beginning CSS3

MASTERING THE LANGUAGE OF WEB DESIGN

David Powers

Apress®

Beginning CSS3

David Powers

Apress®

Beginning CSS3

Copyright © 2012 by David Powers

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

ISBN 978-1-4302-4473-8

ISBN 978-1-4302-4474-5 (eBook)

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

President and Publisher: Paul Manning
Lead Editor: Ben Renow-Clarke
Technical Reviewer: Andrew Zack
Editorial Board: Steve Anglin, Ewan Buckingham, Gary Cornell, Louise Corrigan, Morgan Ertel, Jonathan Gennick, Jonathan Hassell, Robert Hutchinson, Michelle Lowman, James Markham, Matthew Moodie, Jeff Olson, Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Gwenan Spearing, Matt Wade, Tom Welsh
Coordinating Editor: Anamika Panchoo
Copy Editor: Linda Seifert
Compositor: SPi Global
Indexer: SPi Global
Artist: SPi Global
Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science + Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348–4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at www.apress.com/bulk-sales.

Any source code or other supplementary materials referenced by the author in this text is available to readers at www.apress.com. For detailed information about how to locate your book's source code, go to www.apress.com/source-code.

Contents at a Glance

Contents

About the Author

David Powers is the author of a more than a dozen highly successful books and video courses on web design and development, including *PHP Solutions: Dynamic Web Design Made Easy, Second Edition* (friends of ED, ISBN: 978-1-4302-3249-0) and *Adobe Dreamweaver CS6: Learn by Video: Core Training in Web Communication* (Peachpit Press, ISBN: 978-0-321-84037-0). He also served as the technical reviewer on *Cascading Style Sheets: Separating Content from Presentation, Second Edition* by Owen Briggs, Steven Champeon, et al. (friends of ED), and *Head First HTML and CSS* by Elisabeth Robson and Eric Freeman (O'Reilly). He is an Adobe Community Professional, and is constantly urging the Dreamweaver development team to improve support for web standards and the latest developments in CSS3.

David first began developing websites in 1994 when, as Editor, BBC Japanese TV, he needed a way to promote his fledgling TV channel with next to no budget. He persuaded the IT department to let him have some space on a web server, and created and maintained a bilingual Japanese-English website, much of it coded by hand. After a career in radio and TV journalism spanning nearly 30 years, he left the BBC at the turn of the century, and continued developing bilingual websites for leading clients, including the Japanese Foreign Ministry and an international consultancy. Since 2006, he's devoted himself full time to writing and teaching.

David has also translated several plays from Japanese. To relax, he enjoys nothing better than some cold sake and sushi.

About the Technical Reviewer

Andrew Zack is the CEO of ZTMC, Inc. (www.ztmc.com), which specializes in search engine optimization (SEO) and Internet marketing strategies. His project background includes almost 20 years of site development and project management experience and over 15 years as an SEO and Internet marketing expert.

Andrew has been very active in the publishing industry, having coauthored *Flash 5 Studio* (Apress, 2001) and served as a technical reviewer on more than ten books and industry publications.

Having started working on the Internet close to its inception, Andrew continually focuses on the cutting edge and beyond, concentrating on new platforms and technology to stay at the forefront of the industry.

Acknowledgments

My thanks, as always, go to my editor, Ben Renow-Clarke, who suggested I write this book a long time ago, and gently pestered me until I said yes. His perspicacious comments have greatly improved the manuscript. My technical reviewer, Andrew Zack, has also made many helpful suggestions and picked up my errors. Any that remain are my responsibility alone. I'd also like to thank my coordinating editor, Ana Panchoo, for smoothing the whole process over the many months it takes to write a book and see it through the editorial and production process.

Most of all, though, my thanks go to you for taking the trouble to read this book.

Introduction

CSS3 is the latest version of Cascading Style Sheets, the language that in the hands of a skilled designer turns the ugly duckling of unadorned HTML into the gracious swan of a sophisticated web page. Some web designers will tell you dismissively that CSS3 won't be ready for years. In one sense, they're right. Instead of one massive specification, CSS3 has been broken up into more than 40 modules, some of which are unlikely to gain formal approval for a long time. But the first modules are already complete, while others have reached a high level of stability and are widely implemented by browsers. What's more, CSS3 embraces the whole of the previous version, CSS2.1, a formal standard that's supported by all browsers in widespread use.

This book covers not only the new parts of CSS3, but also those inherited from CSS2.1. Even ancient browsers, such as Internet Explorer (IE) 6 and 7, support most of CSS2.1. So, you'll find large parts of this book relevant even if you still need to support older browsers.

Because CSS is constantly evolving, this book inevitably represents a snapshot in time (August 2012). But I've tried to structure the content in such a way that it will remain relevant even when new modules begin to mature. New modules will add extra features rather than supersede what's covered in this book.

In addition to all the visual properties of CSS2.1, it covers stable features from the following CSS3 modules:

- CSS3 Selectors
- CSS3 Color
- CSS3 Values and Units
- Media Queries (for responsive web design)
- CSS3 Backgrounds and Borders
- CSS3 Text
- CSS3 Fonts
- CSS3 Image Values and Replaced Content (gradients)
- CSS Multi-column Layout
- CSS Transforms
- CSS Transitions
- CSS Animations
- CSS Flexible Box Layout

Who This Book Is For

The title *Beginning CSS3* indicates that no prior knowledge of CSS is required. However, you should be familiar with HTML and the basic principles of building websites. Each chapter contains a mixture of reference material and hands-on exercises. The reference sections have been designed to make it easy to find properties and values, so the book should continue to be useful even after you have worked through a chapter.

The broad scope of the material covered in this book means that it should also appeal to readers who already have experience of working with CSS and are looking to consolidate their knowledge and learn new features in CSS3.

How This Book Is Structured

I've split the book into four parts.

Part I deals with the basics of CSS: how style rules are structured and attached to web pages; basic selectors; and specifying sizes, colors, and other values.

Part II explains how to format text and embed web fonts.

Part III covers the main aspects of page layout. It discusses the CSS box model, backgrounds and borders, floats, styling lists, fixed, absolute and relative positioning, and strategies for cross-browser layout.

Part IV dives into more advanced techniques, such as responsive web design with media queries, CSS animation, and flex layout.

Downloading the Code

The code for the examples is available on the Apress website, `www.apress.com/9781430244738`. A link can be found under the Source Code/Downloads tab. This tab is located underneath the Related Titles section of the page.

You are free to use or adapt the code in your own websites. However, the images remain the copyright of the author and are provided only for use with the exercises.

Browser Versions Covered

The examples have been tested on a wide range of browsers and operating systems, including Windows XP, Windows 7, and Mac OS X 10.7, as well as Apple and Android touch-screen devices. Where appropriate, the text notes whether a minimum version of a browser is required to support a feature. IE 8 and earlier support only features inherited from CSS2.1.

Windows 8 and IE 10 had not been released when this book went to press, but all the examples in Part IV were tested on the Developer Preview of IE 10. The most recent versions of other browsers used for testing were Firefox 14, Safari 6, Chrome 21, Opera 12.01, Safari in iOS 5.1.1, and the native browser in Android 2.2 and 3.2.

References in the text to Firefox 16 and Opera 12.5 are based on announcements made by the browser manufacturers about changes planned for those versions. For up-to-date information on which browsers support new CSS3 features, check `http://caniuse.com/#cats=CSS`.

Errors and Corrections

Every effort has been made to ensure the accuracy of the information in this book, but if something doesn't quite work as expected or you think you have spotted an error, I'll keep a list of known errors and significant updates on my website at `http://foundationphp.com/begincss3/`. Also check the Errata tab on the book's information page on the Apress website, `www.apress.com/9781430244738`. If the error isn't listed in either place, submit an error report through the Apress site. In normal circumstances, I try to respond to error reports within 24 hours of them being passed onto me. Submitting errors through the Apress site is the best way to ensure corrections are made to future printings of the book.

Getting Help

If you need help solving a problem with your own CSS, upload the web page to a temporary location on your website and post a request for help in an online forum. Give the URL of the problem page, and describe the issue briefly. Also mention if the problem is limited to a specific browser. It's only by seeing the HTML, CSS, and other assets in context that others can help troubleshoot problems with the way a page displays.

Spread the Word

If you find this book useful, consider spending a few minutes spreading the word by writing a brief review on your blog or an online bookstore. Constructive criticism of how the book might be improved is also welcome.

Note Shortly before this book was sent to the printers, members of the CSS Working Group informally announced that CSS3 won't be followed by CSS4. This is because new modules, such as Flexible Box Layout, don't have equivalents in earlier versions of CSS, creating difficulties for the formal numbering system. In a couple of chapters, I refer to the draft Selectors Level 4 module as "CSS4 Selectors." I decided to leave those references unchanged because no consensus has emerged on what to call them.

PART I

CSS Basics

If you come from a graphic design background, the idea of using code to control the look and feel of a website probably sounds bizarre. But it's not as bad as it sounds. Styling a site with Cascading Style Sheets (CSS) involves creating a set of instructions that tell the browser what color, size, and font you want to use for your text; how much whitespace you want around different parts of the page; and so on.

The instructions consist of simple pairs of properties and values. Most properties have intuitive names such as `font-size`, `margin-top`, or `border-color`. Values are specified using keywords or concepts that you're likely to be familiar with, for example pixels and RGB or HSL color formats. If you're not sure what they are, don't worry because everything is explained along the way. The three chapters in Part I provide all the basic knowledge you need to start styling your web pages with CSS.

Chapter 1 explains why CSS separates the style information from the HTML markup of web pages. You'll learn what a style rule looks like, and how to make your styles available to all pages in a website. There are also some hints on how to use browsers' developer tools to understand how styles interact with each other.

Chapter 2 introduces the important concept of CSS selectors. Selectors tell the browser which elements you want to apply the styles to. One of the most important selectors is the type (or tag) selector, which redefines the default look of an HTML element. CSS3 defines more than 40 selectors. This chapter introduces you to a subset of the most widely used, and shows you how to give links unique styles in different parts of a page.

Chapter 3 covers the nuts and bolts of specifying sizes, colors, and other values used in style rules. Don't try to memorize all the details. Come back to it as a reference when necessary.

CHAPTER 1

Introducing CSS—the Language of Web Design

Building a website with HTML (Hypertext Markup Language) is only half the story. HTML controls the structure of a web page, using tags such as `<p>` for paragraphs, `<img>` for images, and `<input>` for form elements. This structural role has become even more important with the introduction of semantic elements, such as `<article>` and `<nav>`, in HTML5. But HTML says nothing about how the page should look. That's the role of Cascading Style Sheets (CSS). Even default styles are controlled internally by the browser's own CSS.

Instead of applying styles directly to individual elements, the most common—and efficient—way of using CSS is to create the styles in a separate file. At first, this separation of style information from the content of a web page seems counterintuitive. But it has the following significant advantages:

- Less-cluttered HTML code, making it easier to read and maintain
- Changing the look of multiple pages across a site by editing a single file
- Greater control over the way page elements look

Tip CSS stands for Cascading Style Sheets, but most web designers say "CSS is. . ." rather than "CSS are. . ." This is because they're referring to CSS as a technology and not to individual style sheets.

In the early days of web design, the only way to style elements was with HTML tags, such as `<font>` and `<center>`, and attributes, such as `align` and `size`, which needed to be applied to each element individually. Styling web pages like this not only limited the range of design options, but it also meant you needed to make changes to each element individually. With an external style sheet, you make the change in just one place, and it's automatically applied to all matching elements on every page. It's a blueprint for the website's design. Update the blueprint, and the site updates immediately.

In this chapter, you'll learn about the following:

- A brief history of CSS
- The current state of CSS
- Choosing the tools to help you work with CSS
- How to write style rules and apply them to your web pages
- Using browser-specific prefixes for CSS3
- How to avoid common beginner mistakes

A Short History of CSS

In the beginning, the Web was simple. The first-ever public web page (see Figure 1-1) consisted of plain, unadorned text. Headings were in large, bold type; links were blue and underlined—and that was it.

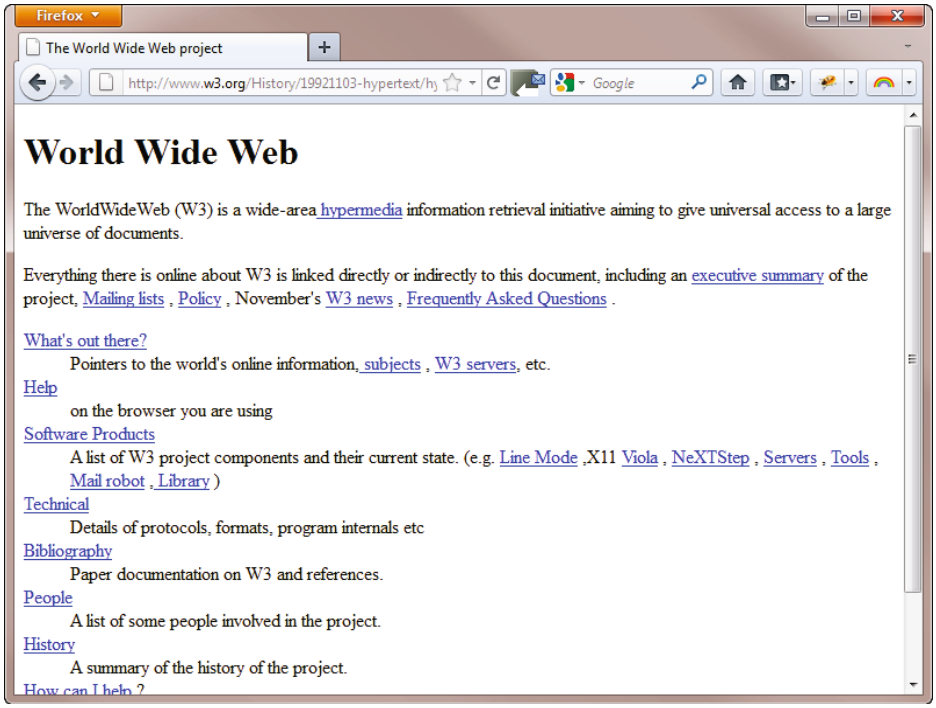

Figure 1-1. *The first-ever web page contained just text and links*

■ **Note** The original, which was created toward the end of 1990, no longer exists, but you can see a copy at `www.w3.org/History/19921103-hypertext/hypertext/WWW/TheProject.html`.

The lack of images and any attempt at styling the page seem odd to us now, but the Web's origins lie in the scientific community, not with artists or graphic designers. It didn't take long before people other than scientists began to demand the ability to include images. Once images began to brighten up web pages, designers began to use their imagination to invent new uses for existing tags. Most notably, the `<table>` tag, which was intended to display scientific data in tabular form, was adapted to provide a grid structure for page layout. HTML was being stretched beyond its limits. Tags such as `<h4>` were no longer used for low-level subheadings, but to display small, bold text. The `<blockquote>` tag, often nested several levels deep, became a way to indent objects, rather than to highlight a quotation from another source. Document structure was thrown to the wind, making it difficult and expensive to maintain web pages or adapt them for other uses, such as printed materials.

The answer was to restore HTML to its original purpose—marking up the structure of the document—and create a new markup language, CSS, devoted to styling the look of a web page. The body responsible for drawing up agreed standards for the Web, the World Wide Web Consortium (W3C), published the first version of this language (CSS1) at the end of 1996, but it wasn't an instant success.

Why Designers Took So Long to Embrace CSS

According to the W3C, the CSS specification was drawn up in response to "pressure from authors for richer visual control." The demand for better control of web pages was certainly there, but browsers in the late 1990s implemented CSS very poorly or not at all. Only the very brave or foolhardy adopted CSS in the early stages. Nevertheless, the W3C published CSS2 in 1998. CSS2 retained all the features of CSS1 and added some new ones.

Those brave enough to embrace CSS ended up banging their heads on their keyboards in frustration. The specification was a statement of what the W3C thought browsers ought to do. The reality was completely different. Neither of the main browsers, Netscape Navigator or Microsoft Internet Explorer (IE), had good support for CSS. However, Microsoft put a huge effort into improving its browser, sparking off what became known as the browser wars. By the time IE 6 was released in 2001, it supported most of CSS, and won the battle for market share.

With Netscape in terminal decline, adventurous designers began to use CSS in earnest, but IE 6 was far from perfect. Worse, Microsoft sat on its laurels and made no effort to improve CSS support in IE 6 until it began to see its market share eroded by new browsers, such as Firefox, Safari, and Opera. Microsoft's response eventually emerged in the form of IE 7 in 2006. IE 8, which followed in 2009, finally offered support for the whole of CSS2.1 (an updated version of CSS2). In the meantime, the rest of the browser market had already started supporting the next generation of standards, CSS3.

Understanding the Different Versions of CSS

Because no two browsers ever managed to implement the full CSS2 specification, the W3C issued a revised specification called CSS2.1 in 2002. However, it took a further nine years before CSS2.1 finally received the formal seal of approval in June 2011. Coincidentally, the first part of CSS3—the Color module—became a formal recommendation on the same day as CSS2.1. To speed up the development and adoption of CSS3, the W3C has divided the specification into some 50 modules. Work on some of them is already at an advanced stage. Others are only ideas, and don't yet have any formal proposals.

■ **Tip** For an up-to-date list of CSS3 modules and their status, visit `www.w3.org/Style/CSS/current-work`.

The CSS3 Color module expands the type of color formats that you can use in websites to include HSL (hue, saturation, lightness) and alpha transparency. The following year, the Media Queries module also became a formal recommendation. Media queries serve different styles to devices depending on screen width and orientation. Other modules at an advanced stage of development—and widely supported by browsers—at the time of writing include Backgrounds and Borders, and Multi-column Layout. The Background and Borders module makes it easy to add multiple backgrounds to page elements, and to create rounded corners and drop shadows without the need for images. There's also strong support for embedding web fonts.

All the main browser makers now seem committed to implementing CSS3 features as quickly as possible, and the pace at which new versions of browsers are released has rapidly accelerated. This means there's no need to wait for a particular module to gain formal approval. You can begin using many parts of CSS3 in your designs right now. One of the best places to get up-to-date information on which features are supported is the website at `http://caniuse.com` (see Figure 1-2).

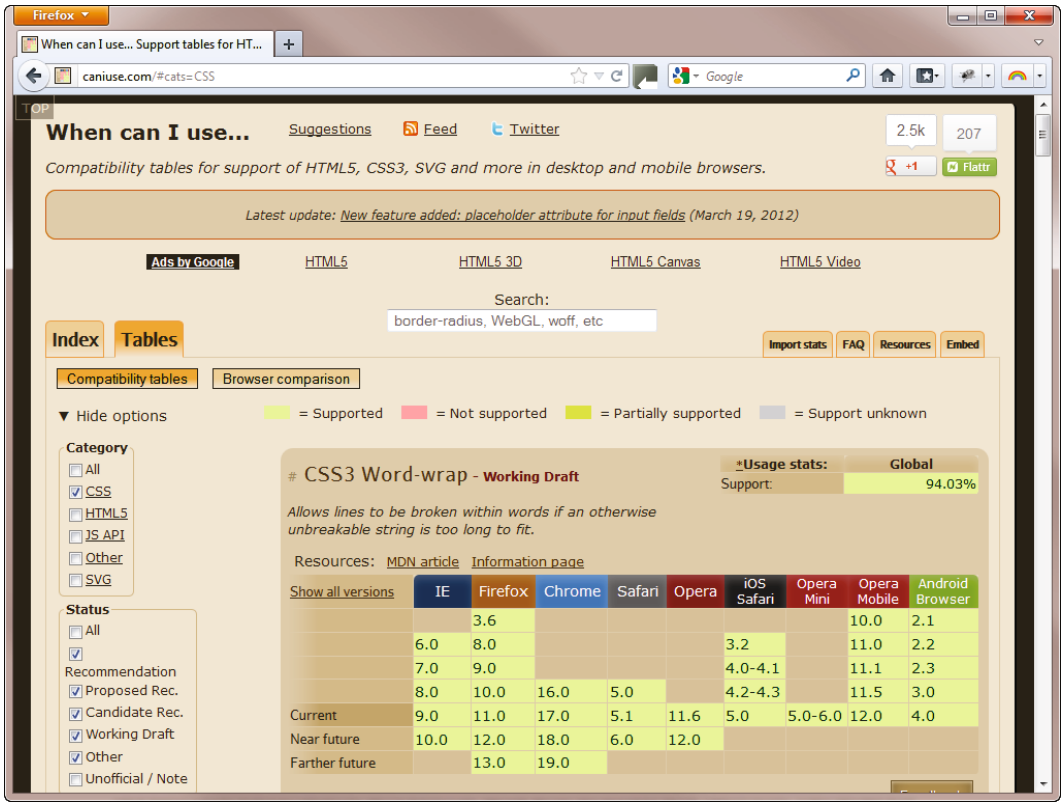

Figure 1-2. *The support tables at caniuse.com provide up-to-date information about CSS3*

Web designers and developers tend to upgrade their browsers much more frequently than other users, who might not even have the freedom to update because of policies imposed by the system administrators at their place of work or study. Choosing whether to use a feature is a decision that only you can make as the designer. If it's important to you that your design looks identical in all browsers, you'll be limited to whatever is supported by the lowest common denominator among your target browsers. However, most people use only one browser, so they won't see the difference—as long as you make sure that the design looks acceptable even in older browsers. Also, many people are now used to seeing websites look different on mobile phones and tablets, so the overall design is more important than pixel-perfect uniformity.

■ **Note** CSS3 builds on the previous versions, preserving existing features and adding new ones. CSS doesn't trigger an error if you use a feature the browser doesn't recognize. Browsers silently ignore CSS properties and values that they don't support.

So, How Do I Use CSS?

You normally style web pages by creating an external file called a *style sheet*, which contains a series of rules identifying which parts of the page you want to affect and how they should look. Like HTML, CSS is written as plain text. You don't need anything more sophisticated than a text editor, such as Notepad or TextEdit, to start writing

CSS. However, if you're using an HTML editor to build your web pages, you'll almost certainly find that it provides you with code hints or other features, such as code coloring, to help create your style rules. Adobe Dreamweaver (www.adobe.com/products/dreamweaver) and Microsoft Expression Web (www.microsoft.com/expression/products/overview.aspx?key=web), both offer pop-up code hints similar to those shown in Figure 1-3.

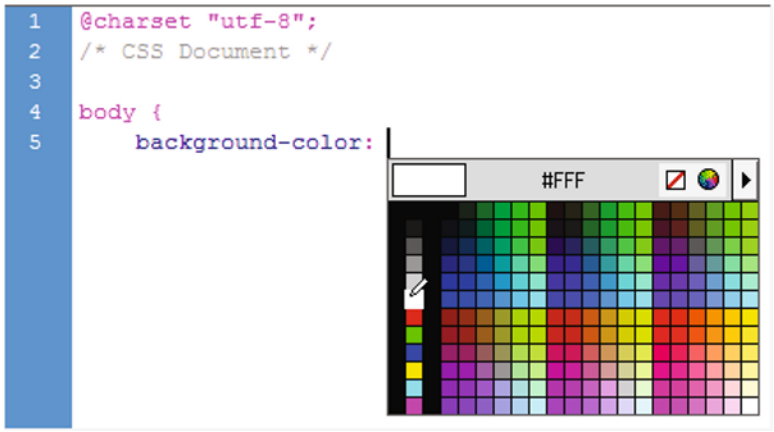

Figure 1-3. *Using CSS code hints in Dreamweaver*

There are also dedicated CSS editors. Among the most popular are Style Master for Windows and Mac (www.westciv.com/style_master/), Espresso for Mac (http://macrabbit.com/espresso/), and Top Style for Windows only (http://svanas.dynip.com/topstyle/index.html).

■ **Note** Your choice of editor is unimportant. This book is strictly software and operating system neutral.

How Do I Write a Style Rule?

Creating a style rule is simple. Figure 1-4 shows the different parts that make up a style rule.

Let's take a look at each part in turn:

- *Selector:* This tells the browser where you want to apply the rule. Figure 1-4 uses a *type selector* (sometimes called a *tag selector*), which redefines the default style of an HTML tag. This example redefines the style of all <p> tags—in other words, paragraphs. You'll learn about other selectors in the next chapter.

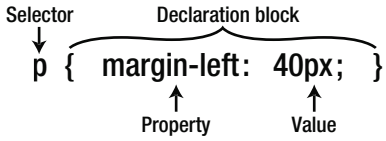

Figure 1-4. *The anatomy of a style rule*

- *Declaration block:* This begins with a left curly brace and ends with a right curly brace. You put your style declarations between these braces. Each declaration consists of a property followed by a colon (`:`) and value, and ends with a semicolon (`;`).
- *Property:* This is one of the properties defined in the CSS specification. Most have intuitive names. The property in Figure 1-4 affects the left margin of the element being styled. Property names are not case-sensitive, but they are normally written entirely in lowercase.
- *Value:* This is the value you want to apply to the property. Some properties have a fixed list of values that you can choose from. Others let you specify the value yourself. The example in Figure 1-4 sets the value of the left margin to 40 pixels.

The declaration block in Figure 1-4 contains only one property/value pair, but you can define any number of properties in the same declaration block.

■ **Tip** Strictly speaking, you can leave out the semicolon after the last declaration in a block or if the block contains only one property/value pair. But you should get into the habit of always using a semicolon because you might forget to insert it when later adding extra declarations to the same block. A missing semicolon in the middle of a block is a common cause of CSS failing to work as expected.

Using Browser-specific Prefixes for CSS3 Properties

At the time it was introduced, IE 6 was considered an excellent browser, but it had some terrible bugs that continued to frustrate designers more than 10 years after its release. To avoid a similar situation with CSS3, browser makers have decided to prefix the names of new or experimental properties with the browser-specific identifiers listed in Table 1-1.

Table 1-1. *The Main Browser-specific Prefixes for CSS3 Properties*

Prefix	Browsers
`-moz-`	Firefox
`-ms-`	Internet Explorer
`-o-`	Opera
`-webkit-`	Google Chrome, Safari

For example, when creating a CSS gradient, you need to use three browser-specific prefixes in addition to the standard property like this:

```
div {
   background-image: -moz-linear-gradient(#C24704, #FFEB79);
   background-image: -o-linear-gradient(#C24704, #FFEB79);
   background-image: -webkit-linear-gradient(#C24704, #FFEB79);
   background-image: linear-gradient(#C24704, #FFEB79);
}
```