Getting StartED with CSS

David Powers

friendsof ED

an Apress® company

GETTING STARTED WITH CSS

ISBN-13 (pbk): 978-1-4302-2543-0

ISBN-13 (electronic): 978-1-4302-2544-7

Printed and bound in the United States of America 9 8 7 6 5 4 3 2 1

Distributed to the book trade worldwide by Springer-Verlag New York, Inc., 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 info@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 http://www.apress.com/info/bulksales.

The information in this book is distributed on an “as is” basis, without warranty. Although every precaution has been taken in the preparation of this work, neither the author(s) nor Apress shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the information contained in this work.

The source code for this book is freely available to readers at www.friendsofed.com in the Downloads section.

Credits

President and Publisher:
Paul Manning

Lead Editor:
Ben Renow-Clarke

Technical Reviewer:
Peter Elst

Editorial Board:
Clay Andres, Steve Anglin, Mark Beckner, Ewan Buckingham, Tony Campbell, Gary Cornell, Jonathan Gennick, Michelle Lowman, Matthew Moodie, Jeffrey Pepper, Frank Pohlmann, Ben Renow-Clarke, Dominic Shakeshaft, Matt Wade, Tom Welsh

Coordinating Editor:
Kelly Moritz

Copy Editor:
Heather Lang

Compositors:
MacPS, LLC

Indexers:
BIM Indexing and e-Services

Artist:
April Milne

Interior Designer:
Anna Ishchenko

Contents at a Glance

Contents

About the Author

David Powers is an Adobe Community Expert for Dreamweaver and author of a series of highly successful books on web design and development, including *The Definitive Guide to Dreamweaver CS4 with CSS, Ajax, and PHP* (friends of ED, ISBN: 978-1-4302-1610-0) and *PHP Solutions: Dynamic Web Design Made Easy* (friends of ED, ISBN: 978-1-5905-9731-1). He also served as the technical reviewer on *Cascading Style Sheets: SeparatingContent from Presentation, Second Edition* by Owen Briggs, Steven Champeon et al (friends of ED), and *Head First HTML with CSS & XHTML* by Elisabeth Freeman and Eric Freeman (O'Reilly).

As a professional writer, he has been involved in electronic media for more than 30 years, first with BBC radio and television and more recently with the Internet. What started as a mild interest in computing was transformed almost overnight into a passion, when David was posted to Japan in 1987 as BBC correspondent in Tokyo. With no corporate IT department just down the hallway, he was forced to learn how to fix everything himself. When not tinkering with the innards of his computer, he was reporting for BBC TV and radio on the rise and collapse of the Japanese bubble economy.

David has also translated several plays from Japanese. To relax, he enjoys nothing better than visiting his favorite sushi restaurant.

About the Technical Reviewer

Peter Elst is a freelance web 2.0 consultant and Founding Partner of Project Cocoon — a collaborative project of web designers and developers based in South India. As a respected member of the online Flash platform community, Peter has spoken at various international industry events and published his work in leading journals.

www.peterelst.com

Acknowledgments

Many people have helped and inspired me in my quest to learn CSS. The person who first opened my eyes to its possibilities was Al Sparber of Project Seven (www.projectseven.com). He still amazes me with his in-depth knowledge of the subject and the thoroughness of his cross-browser testing. Others who have helped me directly or indirectly include Eric Meyer, Owen Briggs, Stephanie Sullivan, Thierry Koblentz, and a secretive group that goes by the name of “No Nephews” (you know who you are). My thanks go to them all.

I would also like to thank everyone at Apress/friends of ED who contributed to bringing this book to fruition. Particular thanks go to my editor, Ben Renow-Clarke, and technical reviewer, Peter Elst. Both made valuable suggestions that greatly improved this book.

Introduction

Most people who build websites fall into one of two categories: geeks (like me) who take great pleasure in working with code, and artistic types who think in terms of overall design. Of course, that's a sweeping generalization. To be successful in modern web design, you need an element of both. But most people will be stronger in one field than the other.

If your main strength lies in visualizing an overall design and you find code a bit of a turn-off, learning how to style websites with Cascading Style Sheets (CSS) might seem a daunting task. But it needn't be. . . . Think of CSS as the *language of design*. The code in a style sheet is simply telling the browser how you want your pages to look. It defines the fonts, colors, and layout of the various elements on the page. You might find yourself despairing at how long it takes to put together a style sheet to reflect your visual design. Take heart; even if you're an expert, it's not something you can dash off in a few minutes. Attention to detail is important—and, as always, it shows in the results. Moreover, once you have created the style sheet, the visual design is automatically applied to every new page that you build. That's the magic of CSS.

Code warriors face a different challenge. The amount of code involved in CSS is quite small. There are fewer than 100 properties to learn, and most of them are intuitively named. It doesn't take a genius to work out that the `border-top` property defines the top border of an element. The syntax is also very simple. As a result, an experienced developer might expect to have the whole thing licked in a couple of days. You won't.

Regardless of your background, learning CSS takes time. It's not because CSS is hard—far from it. What takes time is understanding the infinite number of ways CSS properties can be combined with each other, providing a stunning degree of control over the look and layout of web pages. The other challenge—although it's becoming less significant by the day—comes from the way different browsers interpret CSS. Older browsers, particularly Internet Explorer 6 and 7, don't understand all CSS properties or have bugs. However, all other browsers in widespread use (including Internet Explorer 8) have excellent support for CSS. As long as you build your style sheets to work in a modern browser, you can usually fix any problems in older browsers at the end of the design process.

Although you won't become a CSS master overnight, you can achieve impressive results quite quickly. I have organized the chapters in this book in a logical sequence to lead you from simple beginnings to progressively complex concepts. Even if you already have some experience of CSS, I recommend that you read the chapters in the order they are presented. Each chapter contains a mixture of reference material and hands-on exercises that build on what you have learned previously. The appendix at the end of this book also serves as a quick reference to all the properties in the current version of CSS (CSS2.1), as well as CSS selectors used in all mainstream browsers.

Take things gradually. You'll get there in the end, and you'll discover the time invested eventually repaid in websites that not only look good, but are much easier to maintain.

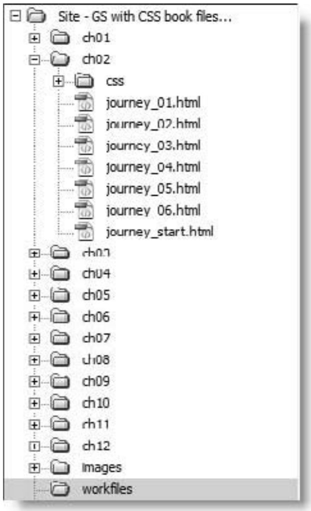
Who this book is for

This book is aimed at anyone involved in building websites using HTML (or XHTML). It assumes no prior knowledge of CSS, but I do expect you to understand the basics of HTML and web page construction. After the first couple of chapters, the book moves at a fairly rapid pace, so this book should also appeal to readers who have dabbled with CSS, but still haven't quite "got it." If you already know some CSS, I strongly urge you *not* to skip the early chapters, because I try to steer you away from overreliance on CSS classes and other bad habits.

Although I show you a lot of cool tricks on the way, throughout this book I concentrate on teaching you how CSS works. Cool tricks are fine, but if you don't understand why something works a particular way, you'll find it difficult to adapt the CSS to achieve the particular effect that *you* want. As far as possible, I steer away from hacks. The primary emphasis is always on how CSS should work in a standards-compliant browser. But until Internet Explorer 6 and 7 finally disappear, you need to know how to deal with the problems they cause. So, I include workarounds for all the major problems with those browsers.

Using the files for this book

The files for all the exercises and examples in this book can be downloaded from the friends of ED website at `http://friendsofed.com/download.html?isbn=9781430225430`. The files are organized into a separate folder for each chapter, and all the internal links are document-relative. To ensure that the internal links continue to work, I suggest that you create a new folder called `workfiles` at the same level as the individual chapter folders, as shown in the following screenshot:

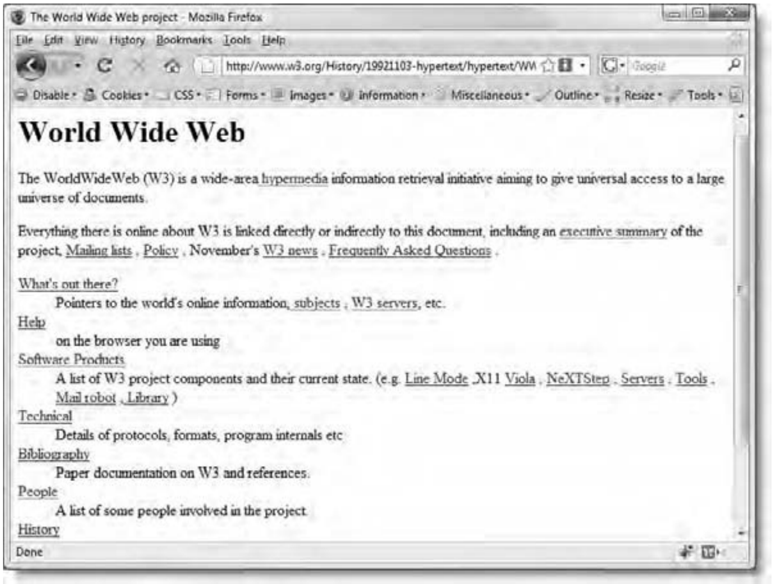


Copy each file as you need it into the `workfiles` folder. You should also create a `css` subfolder inside `workfiles`, and build or copy the style sheets there. This will maintain the correct relationship between the files and the images in all the pages.

Chapter 1

What Is CSS, and Why Should I Learn It?

In the beginning, the Web was simple. Figure 1-1 shows what the first-ever public web page looked like. As you can see, it consisted of plain, unadorned text. Headings were in large, bold type; links were blue and underlined—and that was it.


The World Wide Web project - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.w3.org/History/19921103-hypertext/hypertext/WW
Google
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools
Done

World Wide Web

The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary of the project, Mailing lists , Policy , November's W3 news , Frequently Asked Questions .

What's out there?
    Pointers to the world's online information, subjects , W3 servers, etc.
Help
    on the browser you are using
Software Products
    A list of W3 project components and their current state. (e.g. Line Mode ,X11 Viola , NeXTStep , Servers , Tools , Mail robot , Library)
Technical
    Details of protocols, formats, program internals etc
Bibliography
    Paper documentation on W3 and references.
People
    A list of some people involved in the project.
History

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

LinkED

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. The inventor of the World Wide Web (WWW), Tim Berners-Lee, was working at the European Organization for Nuclear Research (CERN) in Switzerland and was frustrated by the need to log onto different computers to get information. So he devised a way of sharing information among computers. Putting aside the technical details of how information is transferred from one computer to another, the principle behind the Web is very simple. Documents are marked up to indicate what's a heading, paragraph, list, and so on; and links (or hyperlinks, to give them their correct name) tell the web browser where to find related documents. The tags used to mark up the text evolved into what we now know as HyperText Markup Language (HTML).

It didn't take long before people other than scientists realized the potential of the Web and began to demand the ability to include images. Once images began to brighten up web pages, designers wanted not only a way to make text look more interesting but also to lay out the contents of a page in more attractive ways than just headings and paragraphs. In addition to new tags being added to HTML, 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.

The rapid growth of the Web was exciting, but it was also chaotic. HTML was being stretched beyond its limits. Tags such as `<h4>` were no longer being 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 devoted to styling the look of a web page. That new markup language was called Cascading Style Sheets (CSS), and that's what this book is about.

ExplainED

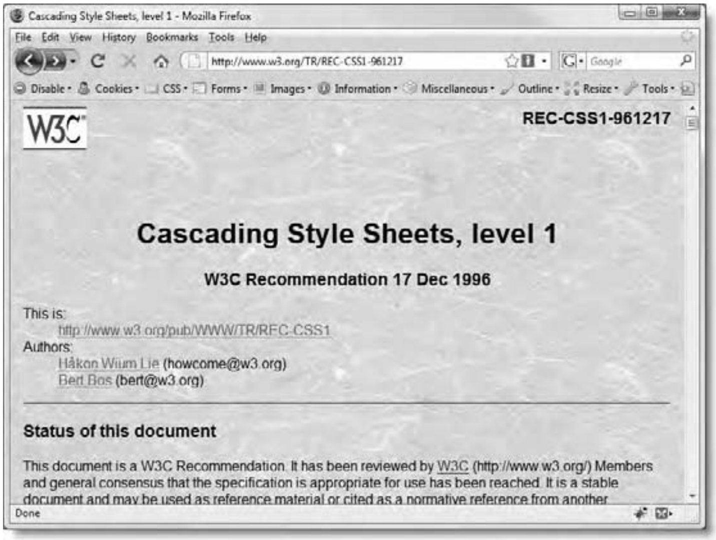
CSS stands for Cascading Style Sheets, but most web designers say "CSS is . . ." rather than "CSS are. . ." This is because they're referring to CSS as a technology and not to individual style sheets.

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

- The advantages of CSS and why now is a good time to start learning
- How to write style rules and apply them to your web pages
- How to avoid common beginner mistakes
- Choosing the tools to help you work with CSS

A short history of CSS

Many designers think of CSS as the "new" way to style web pages, so it comes as quite a surprise to discover that CSS has been with us for years. The original specification (see Figure 1-2) was published by the World Wide Web Consortium (W3C) at the end of 1996.


Cascading Style Sheets, level 1 - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.w3.org/TR/REC-CSS1-961217
Google
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools
W3C
Done

REC-CSS1-961217

Cascading Style Sheets, level 1

W3C Recommendation 17 Dec 1996

This is:
http://www.w3.org/pub/WWW/TR/REC-CSS1
Authors:
Håkon Wium Lie (howcome@w3.org)
Bert Bos (bert@w3.org)

Status of this document

This document is a W3C Recommendation. It has been reviewed by W3C (http://www.w3.org/) Members and general consensus that the specification is appropriate for use has been reached. It is a stable document and may be used as reference material or cited as a normative reference from another

Figure 1-2. The original CSS specification used a very basic set of rules to style the page.

ExplainED

The W3C (`www.w3.org`*) is the body responsible for drawing up agreed standards for the Web. Its members include all the big software and technology companies, as well as government and research institutions from over 40 countries. This often leads to slow decisions. Strictly speaking, W3C standards are only recommendations. That's why they're not always fully supported by all browsers. Equally important, the recommendations sometimes lag behind the pace of innovation on the Web. For example, browsers began supporting CSS opacity long before its incorporation into a W3C standard.*

Why CSS has taken so long to be embraced by designers

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 just weren't up to the job. They implemented CSS very poorly or not at all. As a result, only the very brave or foolhardy adopted CSS in the early stages. Nevertheless, the W3C continued work on the specification and brought out a new version, CSS2, in 1998. This retained all the features of CSS1 and added some new ones.

Instead of designers leaping with joy, 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. The now-defunct Netscape Navigator 4 (usually referred to simply as Netscape 4) was the most popular browser at the time, with an estimated 80 percent market share in 1997. It supported a lot of CSS, but not very well. Its rival, Microsoft Internet Explorer (IE), also in version 4, was even worse. However, Microsoft put a huge effort into improving its browser, and by the time IE6 was released in 2001, it supported most of CSS—although it was far from perfect.

Microsoft's battle with Netscape completely reversed the browser scene, with IE taking about 90 percent of the market share by 2001–2002. With Netscape in terminal decline, adventurous designers began to use CSS in earnest, but they faced several problems:

- Netscape's original dominance of the browser market left a significant user base resistant to change, particularly in schools and public libraries in the United States. This meant finding ways to style web pages that could still be rendered in Netscape 4 without causing it to crash.
- CSS in IE6 was usable, but it had many bugs and didn't support every feature.
- Although IE was the dominant browser, new ones, such as Firefox, Safari, and Opera, came on the scene aiming at full standards compliance and with better support for CSS.

Even when designers felt they could reasonably stop supporting Netscape 4, they faced a new dilemma with the emergence of the standards-compliant browsers. They could either ignore the standards and design for IE6 or create CSS that worked well in the new browsers and find ways to compensate for the flaws in IE6.

For a long time, instead of fixing the CSS bugs in IE6, Microsoft issued only security updates. However, a combination of security scares and demands for better CSS support eventually resulted in Firefox making considerable inroads into its market share. Firefox is managed by the Mozilla Corporation, an organization that traces its origins to Netscape but is now controlled by the nonprofit Mozilla Foundation. For legal reasons, Firefox underwent several name changes, but within two years of its original release in February 2004, it had taken an estimated 10 percent of the browser market share. Microsoft's response emerged in the form of IE7 in 2006. It still wasn't perfect, but it was followed in March 2009 by IE8, which finally supports the whole of CSS2.1. Firefox, in the meantime, has continued its rise and currently represents nearly one in four of all browsers in use.

ExplainED

The W3C doesn't formally adopt a CSS specification until all parts of it are implemented by two browsers or user agents, such as screen readers for the visually impaired. Because no two browsers ever managed to implement the full CSS2 specification, the W3C dropped some features, added some new ones, and issued a revised specification called CSS2.1 in 2002. More than a decade after the original publication of the CSS2 specification, CSS2.1 still hadn't received formal approval. No wonder web designers have been frustrated with the glacial progress of CSS!

The time for CSS has finally come

After such a dismal history, you might be wondering whether it's worth the effort of learning CSS. The answer is a resounding yes. Within seven months of its release, IE8 represented 18 percent of browsers in use, reducing the combined market share of IE6 and IE7 to roughly 40 percent. The bad news for web designers is that, in the early months at least, IE8's rise was due to people switching from IE7; little dent was made in the market share for IE6, which has annoying CSS bugs. It's fair to say it will take several years before IE6 and IE7 disappear completely from the scene, but the arrival of IE8 and other modern browsers, such as Firefox, Safari, and Opera, means you can now use CSS with far greater confidence than ever before. As long as the website remains usable in older browsers, you don't need to worry if there are some minor differences in display.

AdvancED

Before deploying a website on the Internet, check what it looks like in all the main browsers and on different operating systems. If you don't have access to some browsers or operating systems, ask in an online forum for others to check for you, or subscribe to a browser testing service, such as Browsercam (`www.browsercam.com`*). If you have Dreamweaver CS4 or later, you can use Adobe BrowserLab (*`http://browserlab.adobe.com/`*). On Windows, you can also use Expression Web SuperPreview (*`http://expression.microsoft.com/en-us/dd565874.aspx`*) to compare how your pages look in IE6 and IE7 or IE8. The version of Expression Web SuperPreview released just as this book was about to go to press also included support for Firefox 3.5, so more browsers might be added in future.*

Some designers fret if their site doesn't look exactly the same in each browser. Don't worry about the difference of an odd pixel or two. Most visitors only see your site in one browser. What really matters is that it works and looks acceptable to each visitor.

What are the advantages of CSS?

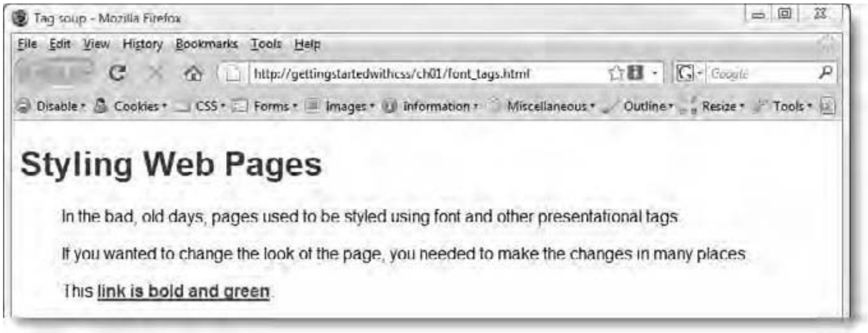
CSS has three huge advantages, namely:

- Less-cluttered HTML code, making it easier to read and maintain
- The ability to change the look of a whole site by changing a single file
- Greater control over the way page elements look

Let's take a look at each of these in detail.

Write simpler markup

Figure 1-3 shows a simple web page with a heading, three paragraphs, and a link. The download files for this chapter contain two versions of the same page: one styled the old way with presentational HTML (`font_tags.html`), and the other styled using CSS (`css.html`). Both look exactly the same in a browser.


Tag soup - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://gettingstartedwithcss/ch01/font_tags.html
Google
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools

Styling Web Pages

In the bad, old days, pages used to be styled using font and other presentational tags

If you wanted to change the look of the page, you needed to make the changes in many places

This link is bold and green

Figure 1-3. Both CSS and old-style presentational tags remain hidden from view in the browser.

The difference between the two versions becomes obvious only when you examine the underlying HTML. This is what the page that uses presentational tags (highlighted in bold) looks like:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<title>Tag soup</title>
</head>
```

```
<body bgcolor="#FFFFFF" text="#000000" link="#006600"
vlink="#009966" alink="#006600">
<h1><font color="#990000" size="6" face="Arial, Helvetica,
sans-serif">Styling Web Pages</font></h1>
<blockquote>
  <p><font size="3" face="Arial, Helvetica, sans-serif">In the
bad, old days, pages used to be styled using font and other
presentational tags.</font></p>
  <p><font size="3" face="Arial, Helvetica, sans-serif">If you
wanted to change the look of the page, you needed to make the
changes in many places.</font></p>
  <p><font size="3" face="Arial, Helvetica, sans-serif">This
<strong><a href="#">link is bold and green</a></strong>.</font>
</p>
</blockquote>
</body>
</html>
```

The HTML for the version of the page that uses CSS looks like this:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=utf-8" />
<title>Styling with CSS</title>
<link href="css/simple.css" rel="stylesheet" type="text/css" />
</head>

<body>
<h1>Styling Web Pages</h1>
<p>In the bad, old days, pages used to be styled using font and
other presentational tags.</p>
<p>If you wanted to change the look of the page, you needed to
make the changes in many places.</p>
<p>This <a href="#">link is bold and green</a>.</p>
</body>
</html>
```

ExplainED

The examples in this book use Extensible HyperText Markup Language (XHTML) 1.0, which is identical to HTML 4.01, except that it follows slightly stricter rules. XHTML was originally intended to replace HTML, but the W3C began work on HTML5 in 2007. Then in July 2009, it announced that work would stop on XHTML 2. This threw the web development community into confusion over which standard they should use, with some people mistakenly believing that XHTML 1.0 was also being abandoned. XHTML 1.0 remains an approved standard, and is still preferred by many developers because of its stricter rules.

At the time of this writing, HTML5 is still only a draft, but the W3C has stated that it will be compatible with both HTML 4.01 and XHTML 1.0. So, it doesn't matter which version you choose. I use "HTML" to refer to all three flavors. The CSS taught in this book will work with all of them.

The markup is much simpler and easier to read. Even if you use an HTML editor that generates the markup for you, there are times when it's necessary to examine the underlying code. The lightweight code used by a well-designed CSS site makes this a pleasure, rather than a perilous journey hacking through masses of tangled code. There's also less of it, so the page loads more quickly and uses less bandwidth.

AdvancED

Just because most people use broadband these days, it doesn't mean that page size is no longer important. Masses of code not only take longer to display in a browser, but the bigger your pages, the more bandwidth you consume. On a popular site, this can cost a lot of money in extra bandwidth charges.

"So, where's the CSS?" you might be asking. It's not in the web page, but in a separate file (called a style sheet), `simple.css`. The `<link>` tag highlighted in bold in the `<head>` of the page tells the browser where to find the styles. The code inside `simple.css` looks like this:

```
body {
  font-family: Arial, Helvetica, sans-serif;
  color: #000;
  background-color: #FFF;
}
h1 { color: #900; }
p { margin-left: 40px; }
a:link {
  color: #060;
  font-weight: bold;
}
a:visited { color: #096; }
a:hover, a:active { color: #060; }
```

Don’t worry about the meaning of the CSS code yet. You’ll learn about the structure of style rules in “How do I write a style rule?” later in this chapter, and you’ll be writing your own rules to format text in Chapter 2.

Although it might seem strange to put instructions on how your page should look in a separate file, there’s a very good reason for doing so. You can attach the same set of instructions to every page in your website. Unlike presentational HTML markup, which needs to be applied individually to each element, CSS gives you central control over the look of the whole site. What’s more, changes to your style sheet are automatically applied to all pages that are linked to it.

AdvancED

When styles are defined in an external style sheet, the browser stores them in the cache on the visitor’s computer, so they need to be downloaded only once regardless of how many pages are viewed in your site. This speeds up the display of subsequent pages and reduces bandwidth usage

Turn into a quick change artist

The best way to see the power of CSS in action is to visit www.csszengarden.com. Every page in the site looks completely different (see Figure 1-4), but if you look at the underlying HTML, you’ll see that it’s actually exactly the same page. What changes the look of the page is the style sheet attached to it.

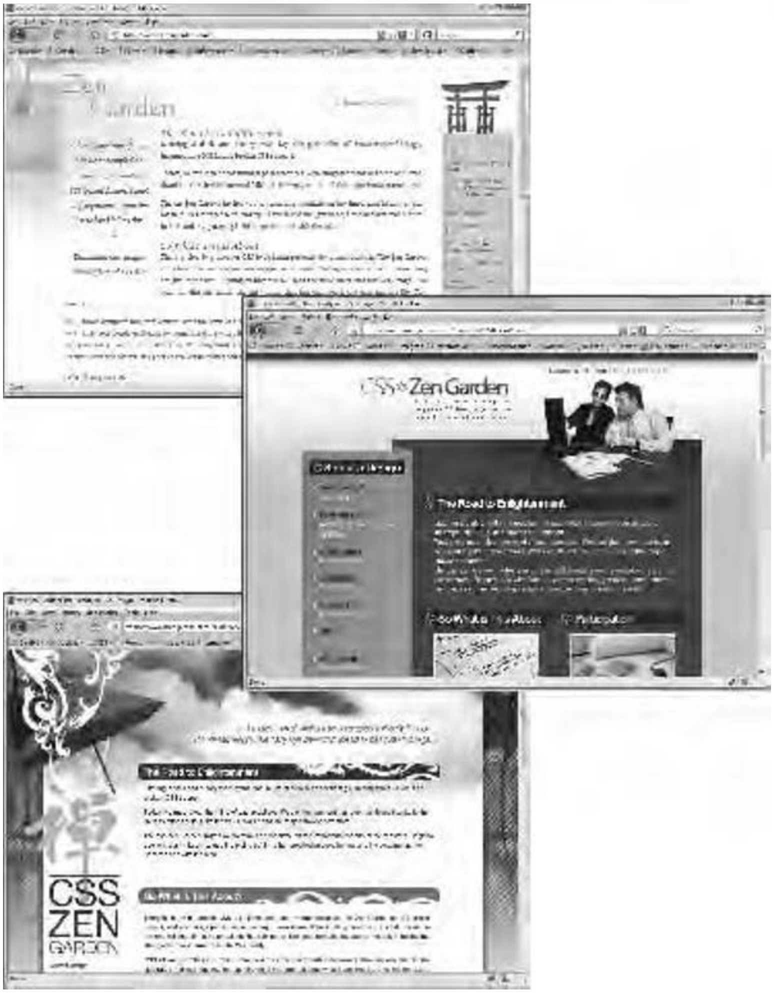
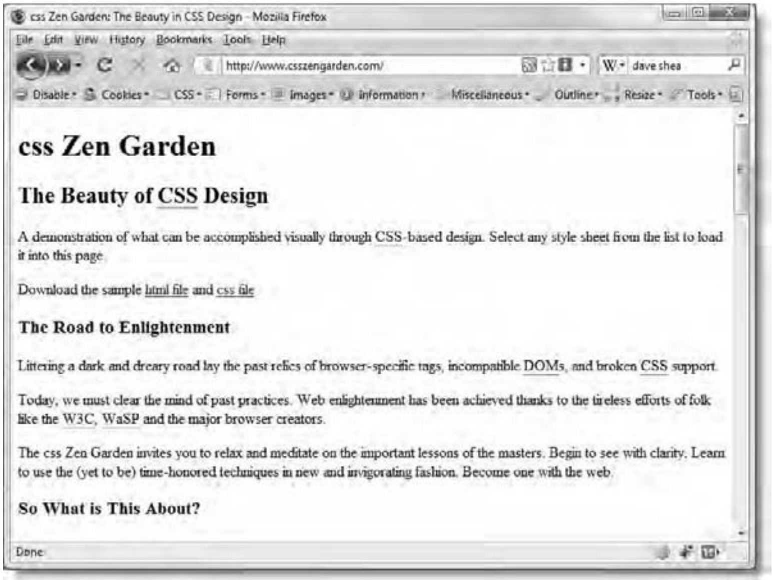


Figure 1-4. The CSS Zen Garden demonstrates the power of CSS to change the look of a website.

The CSS Zen Garden was launched in 2003 by a Canadian web designer named Dave Shea. It was a call to arms to web designers to show what could be done with CSS. He asked designers to submit original visions in the form of style sheets and images. There was one basic rule: no changes could be made to the underlying HTML. In fact, if you look at the site with CSS turned off, every single page looks like Figure 1-5—plain, unadorned text. There are no images in the page itself; they're all added as background images through CSS.


css Zen Garden: The Beauty in CSS Design - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.csszengarden.com/
dave shea
Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools
Done

css Zen Garden

The Beauty of CSS Design

A demonstration of what can be accomplished visually through CSS-based design. Select any style sheet from the list to load it into this page.

Download the sample html file and css file

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, and broken CSS support.

Today, we must clear the mind of past practices. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP and the major browser creators.

The css Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the (yet to be) time-honored techniques in new and invigorating fashion. Become one with the web.

So What is This About?

Figure 1-5. Without a style sheet, the CSS Zen Garden is just plain text.

AdvancED

All modern browsers let you view web pages without CSS. In Firefox, select View ➤ Page Style ➤ No Style. *In IE8, select* Page ➤ Style ➤ No Style. *In Opera, select* View ➤ Style ➤ User Mode. *Safari also lets you disable styles, but first you need to enable the* Develop *menu by opening the* Preferences *panel (from the* Safari *menu on a Mac, or the* Edit *menu on Windows). Select the* Advanced *tab, and enable the* Develop *menu. Thereafter, you can turn off CSS in Safari by selecting* Develop ➤ Disable Styles.

The CSS Zen Garden shows how one page can be restyled in many different ways. This is powerful stuff. It means that you can make a site look completely different just by changing the style rules. But just imagine if, instead of one page, you have a site with dozens or hundreds of pages. If the same style sheet is attached to each page, you can change the look of the whole site just by making changes to the style sheet. To take a very simple example, the `simple.css` style sheet in the previous section contains this style rule: