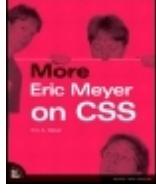


[< Day Day Up >](#)

[NEXT](#)



[Table of Contents](#)

[Index](#)

More Eric Meyer on CSS

By Eric A. Meyer

Publisher: New Riders Publishing

Pub Date: April 19, 2004

ISBN: 0-7357-1425-8

Pages: 304

Slots: 1.0

More Valuable CSS Projects from the Master-Eric Meyer!

Just like its predecessor, this book takes a hands-on approach to teaching CSS. Through ten all new projects, Eric Meyer continues to instruct CSS devotees on how to make CSS work to solve their design challenges. Projects include converting and existing page to use CSS-P instead of tables and styling a variety of pages, such as: a photo gallery, financial report, list-based menu, tab interface, weblog, and irregular edged text.

Eric A. Meyer has been working with the Web since late 1993. He is the principal consultant for Complex Spiral Consulting. A graduate of and former Webmaster for Case Western Reserve University, Eric is also an Invited Expert with the W3C CSS+FP Working Group and coordinated the authoring and creation of the W3C's CSS1 Test Suite. He often speaks at conferences on the subjects of CSS, Web design, Web standards, Web browsers, and how they all go together.

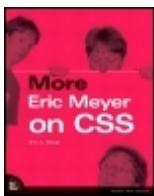
[< Day Day Up >](#)

[NEXT](#)

[PREV](#)

[< Day](#) [Day Up >](#)

[NEXT](#)



[Table of Contents](#)

[Index](#)

More Eric Meyer on CSS

By Eric A. Meyer

Publisher: New Riders Publishing

Pub Date: April 19, 2004

ISBN: 0-7357-1425-8

Pages: 304

Slots: 1.0

[Copyright](#)

[About the Author](#)

[About the Technical Reviewers](#)

[Acknowledgments](#)

[Tell Us What You Think](#)

[Foreword](#)

[Introduction](#)

[Should You Buy This book?](#)

[What You Can Expect from This Book](#)

[Project Overview](#)

[Companion Web Site](#)

[Conventions](#)

[Chapter 1. Converting an Existing Page](#)

[Project Goals](#)

[Preparation](#)

[Laying the Groundwork](#)

[Converting the Document](#)

[Rebuilding the Design](#)

[Assessing the Benefits](#)

[Branching Out](#)

[Chapter 2. Styling a Photo Collection](#)

[Project Goals](#)

[Preparation](#)

[Laying the Groundwork](#)

[Creating the Contact Sheet View](#)

[Creating the Gallery View](#)

[Creating the Catalog View](#)

[Branching Out](#)

[Chapter 3. Styling a Financial Report](#)

[Project Goals](#)

[Preparation](#)

[Styling for the Screen](#)

[Styling for Print](#)

[Branching Out](#)

[Chapter 4. Positioning in the Background](#)

[Project Goals](#)

[Preparation](#)

[Style at Dawn](#)

[Beached Styles](#)

[Branching Out](#)

[Chapter 5. List-Based Menus](#)

[Project Goals](#)

[Preparation](#)

[Laying the Groundwork](#)

[Open and Airy](#)

[Enclosing the Links](#)

[Branching Out](#)

[Chapter 6. CSS-Driven Drop-Down Menus](#)

[Project Goals](#)

[Preparation](#)

[Laying the Groundwork](#)

[Laying Out the Menus](#)

[Reorienting the Menus](#)

[For Your Consideration](#)

[Branching Out](#)

[Chapter 7. Opening the Doors to Attractive Tabs](#)

[Project Goals](#)

[Preparation](#)

[Laying the Groundwork](#)

[Styling the Links](#)

[Creating Actual Tabs](#)

[Branching Out](#)

[Chapter 8. Styling a Weblog](#)

[Project Goals](#)

[Preparation](#)

[Laying the Groundwork](#)

[Styling the Weblog](#)

[Branching Out](#)

[Chapter 9. Designing a Home Page](#)

[Project Goals](#)

[Preparation](#)

[Laying the Groundwork](#)

[Creating the Design](#)

[Branching Out](#)

[Chapter 10. Designing in the Garden](#)

[Project Goals](#)

[Preparation](#)

[Laying the Groundwork](#)

[Creating the Design](#)

[Adding a PNG](#)

[Reflections](#)
[Branching Out](#)

Index



[< Day](#) [Day](#) [Up >](#)



 PREV

[< Day](#) [Day Up >](#)

NEXT 

Copyright

Copyright 2004 by Eric A. Meyer

FIRST EDITION: April 2004

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from the publisher, except for the inclusion of brief quotations in a review.

Library of Congress Catalog Card Number: 2003112955

06 05 04 7 6 5 4 3 2 1

Interpretation of the printing code: The rightmost double-digit number is the year of the book's printing; the rightmost single-digit number is the number of the book's printing. For example, the printing code 04-1 shows that the first printing of the book occurred in 2004.

Printed in the United States of America

Trademarks

All terms mentioned in this book that are known to be trademarks or service marks have been appropriately capitalized. New Riders Publishing/Peachpit Press cannot attest to the accuracy of this information. Use of a term in this book should not be regarded as affecting the validity of any trademark or service mark.

Warning and Disclaimer

This book is designed to provide information about cascading style sheets (CSS). Every effort has been made to make this book as complete and as accurate as possible, but no warranty of fitness is implied.

The information is provided on an as-is basis. The authors and New Riders Publishing/Peachpit Press shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book.

Credits

ASSOCIATE PUBLISHER

Stephanie Wall

 PREV

[< Day](#) [Day Up >](#)

NEXT 

[PREV](#)

[< Day Day Up >](#)

[NEXT](#)

About the Author



Eric A. Meyer has been working with the Web since late 1993 and is an internationally recognized expert on the subjects of HTML, CSS, and Web standards. A widely read author, he is also the founder of Complex Spiral Consulting (www.complexspiral.com), which focuses on helping clients save money and increase efficiency through the use of standards-oriented Web design techniques and counts Macromedia and Wells Fargo Bank among its clients.

Beginning in early 1994, Eric was the visual designer and campus Web coordinator for Case Western Reserve University Web site, where he also authored a widely acclaimed series of three HTML tutorials and was project coordinator for the online version of the Encyclopedia of Cleveland History combined with the Dictionary of Cleveland Biography (ech.cwru.edu), the first example of an encyclopedia of urban history being fully and freely published on the Web.

Author of Eric Meyer on CSS: Mastering the Language of Web Design (New Riders), Cascading Style Sheets: The Definitive Guide (O'Reilly & Associates), and CSS2.0 Programmer's Reference (Osborne/McGraw-Hill), as well as numerous articles for the O'Reilly Network, Web Techniques, and Web Review, Eric also created the CSS Browser Compatibility Charts and coordinated the authoring and creation of the W3C's official CSS Test Suite. He has lectured to a wide variety of organizations, including Los Alamos National Laboratory, the New York Public Library, Cornell University, and the University of Northern Iowa. Eric has also delivered addresses and technical presentations at numerous conferences, among them the IW3C2 WWW series, Web Design World, CMP, SXSW, the User Interface conference series, and The Other Dreamweaver Conference.

In his personal time, Eric acts as List Chaperone of the highly active css-discuss mailing list (www.css-discuss.org), which he co-founded with John Allsopp of Western Civilisation and is now supported by evolt.org. Eric lives in Cleveland, Ohio, which is a much nicer city than you've been led to believe, and is the host of "Your Father's Oldsmobile," a Big Band-era radio show heard weekly on WRUW 91.1-FM in Cleveland (www.wruw.org). When not otherwise busy, he is usually bothering his wife Kat in some fashion.

[PREV](#)

[< Day Day Up >](#)

[NEXT](#)

[PREV](#)

[< Day Day Up >](#)

[NEXT](#)

About the Technical Reviewers

These reviewers contributed their considerable hands-on expertise to the entire development process for More Eric Meyer on CSS. As the book was being written, these dedicated professionals reviewed all the material for technical content, organization, and flow. Their feedback was critical to ensuring that More Eric Meyer on CSS fits our reader's need for the highest-quality technical information.



Porter Glendinning is the owner and Principal Consultant of Cerebellion Design. He lives outside Washington, D.C. with his wife, Laura, who puts up with his obsession with the Internet; and their very large yellow Lab, Arrow, who eats his socks. Porter can be found online at www.g9g.org and www.cerebellion.com. He co-administers the Babble mailing list, a forum for discussions on advanced Web design topics of all sorts (www.babblelist.com), and is a member of the Web Standards Project Steering Committee (www.webstandards.org).



Dave Shea is the creator and cultivator of the highly influential CSS Zen Garden (<http://www.csszengarden.com/>). As well as being a member of the Web Standards Project (<http://www.webstandards.org/>), Dave is the owner and director of Bright Creative (<http://www.brightcreative.com/>), and he writes about all things Web for his daily weblog, [mezzoblue.com](http://www.mezzoblue.com). Living in Vancouver, B.C., Canada, you can usually find him at the local farmer's markets or independent coffee roasters when he's not in front of a screen.

[PREV](#)

[< Day Day Up >](#)

[NEXT](#)

 PREV

[< Day Day Up >](#)

 NEXT

Acknowledgments

Linda Bump Harrison and Lori Lyons, both of New Riders Publishing/Peachpit Press, provided a great deal of needed support and encouragement throughout this entire project, and deserve special mention for the patience and tolerance they displayed whenever life ran roughshod over my deadlines. Which was often.

Major thanks are due to my technical reviewers, Dave Shea and Porter Glendinning, for their input regarding points to highlight, passages to clarify, and mistakes to fix. Extra thanks to Dave for the stunning visual design he contributed to [Project 10](#).

I'd particularly like to thank Douglas Bowman for agreeing to write the Foreword. I've admired Doug's work ever since he redesigned Wired News in 2002; he has time and again combined technical savvy with great visual design to produce truly wonderful results. It's an honor to have him introduce the book.

My thanks to the thousands of members of css-discuss, the mailing list I chaperone, for making it one of the best resources I know, and for keeping the signal level so high. Thanks also to evolt.org and John Handelaar, for giving the list a home and keeping it running smoothly, and to John Alsopp of Western Civilisation for helping me launch the list in the first place.

As always, I'd like to express my deep gratitude to everyone who has contacted me over the years with praise, complaints, comments, suggestions, questions, and ideas regarding CSS, browsers, and my writing. I'm sorry that I couldn't respond to everyone, but I did read what you had to say. Thank you, one and all.

Finally, to my wife, Kathryn—you are the most amazing companion a man could ever hope to have. Without your support, strength, and abiding faith, I would likely never have done so much nor come so far, and I'll never be able to thank you enough for all you've done and meant to me.

Eric A. Meyer
February 2004

 PREV

[< Day Day Up >](#)

 NEXT

 PREV

[< Day Day Up >](#)

 NEXT

Tell Us What You Think

As the reader of this book, you are the most important critic and commentator. We value your opinion and want to know what we're doing right, what we could do better, what areas you'd like to see us publish in, and any other words of wisdom you're willing to pass our way.

As the Associate Publisher for New Riders Publishing/Peachpit Press, I welcome your comments. You can fax, email, or write me directly to let me know what you did or didn't like about this book—as well as what we can do to make our books stronger.

Please note that I cannot help you with technical problems related to the topic of this book, and that due to the high volume of mail I receive, I might not be able to reply to every message.

When you write, please be sure to include this book's title and author as well as your name and phone or fax number. I will carefully review your comments and share them with the author and editors who worked on the book.

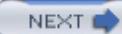
Fax: 317-428-3280

Email: stephanie.wall@peachpit.com

Mail:
Stephanie Wall
Associate Publisher
New Riders Publishing/Peachpit Press
800 E. 96th Street, 3rd Floor
Indianapolis, IN 46240 USA

 PREV

[< Day Day Up >](#)

 NEXT

 PREV

[< Day](#) [Day Up >](#)

NEXT 

Foreword

"Eric Meyer." Say that name, and you'll immediately grab my attention, and possibly engage me in a conversation, even if you're a complete stranger. I was browsing through the tech section of a bookstore last year when I heard a stranger announce to her companion the title of a book she had been thumbing through, "It's called Eric Meyer on CSS. I think I've heard of this guy."

I stepped a little closer, pardoned myself, and offered my unsolicited advice.

"Don't even hesitate if you're thinking about buying that book."

I qualified my statement, making sure she had at least been introduced to the basics of CSS. "It's good, and Eric Meyer is good. You'll be more enlightened after working through just one chapter of that book."

We talked a little more about the book, and my knowledge of Eric Meyer. She thanked me, and then tucked it confidently under her arm to head for the cashier.

If you knew just how pivotal Eric Meyer has been in turning around my understanding of CSS, and in how I use it to push the limits of Web design, you'd also know why I have no qualms recommending his books to complete strangers.

You see, I ignored CSS for years.

While I was working at HotWired, my colleagues thought I would love CSS, and took every opportunity they could to encourage me to dive into the world of style-sheet-driven Web design. Although I am first and foremost a designer, my colleagues knew I have a fairly strong technical mind that can wrap itself around confined logical concepts. However, I'm no good at tolerating inconsistency and unpredictability when it comes to code and its behavior.

When I finally gave in to pressure and started dabbling with CSS, I immediately hit a brick wall. With 4.0 versions of Netscape Navigator and Internet Explorer, I faced nothing but frustration every time I tried using CSS beyond color and basic type treatment. I wanted to see consistent margins, type size, and positioning across common browsers and platforms. In 1998, support for even these basic features was horrendous, causing big headaches for any designer who tried to produce the same look in multiple browsers.

Thus, I wrote off CSS as a failed pipedream that certainly wasn't for me. I wanted to continue reproducing beautiful, usable design, and I wasn't about to trust CSS and its buggy browser support as the means to implement and control my design.

During those trials with CSS, one of my good fortunes was coming across one of the only books at the time dedicated entirely to Cascading Style Sheets. Luckily for me, it was written by Eric Meyer.

Eric's book sat unused on my shelf for a few years while I avoided CSS. Eventually, circumstances began to change. I started seeing news of much-improved browser support for CSS. Small sites were using CSS more abundantly, and it looked like they were producing fairly consistent results. The changes piqued my interest enough to turn my head toward CSS and make me crave more information.

Almost everywhere I looked, I saw Eric Meyer's name—the author of the book I owned—attached to helpful resources. Articles on CSS, CSS test suites, a CSS mailing list, and his CSS master grid that I started using religiously to check possible properties and value combinations.

His book I had purchased several years prior no longer resided on the bookshelf, but on a corner of my desk where I could easily reach it, and make use of it as a constant reference. "How does positioning work again?" "What's that

 PREV

[< Day](#) [Day Up >](#)

NEXT 

 PREV

[< Day Day Up >](#)

NEXT 

Introduction

What you're holding in your hands right now, assuming you aren't viewing a preview online, is more or less a sequel to Eric Meyer on CSS, which was published in 2002 to fairly resounding acclaim. The project-based approach drew high marks, and it seems that a lot of people liked the feeling of being able to watch over my shoulder as I worked through the projects. That was exactly the feeling I aimed to provide, and I've endeavored to create the same feeling with this book.

So, if you do buy this book and you like it, you can get more of the same in Eric Meyer on CSS. On the other hand, it's important to note that you don't have to own Eric Meyer on CSS to use and enjoy this book. Each stands on its own as a self-contained, independent work. So don't be afraid that you won't be able to understand what the characters in this book are doing because you never read the first one. I don't have any characters.

There is a plot, though (actually, two of them). The first plot describes a journey of learning and experimentation, wherein our hero (that's you) follows the path of an experienced guide and learns the ways of a new and wondrous land. The second plot (kind of a subplot) is an underhanded attempt to lure you into using more CSS by tempting you with design flexibility, improved accessibility, reduced page weight, and cool visual effects.

 PREV

[< Day Day Up >](#)

NEXT 

 PREV

[< Day](#) [Day Up](#) [>](#)

NEXT 

Should You Buy This book?

This isn't a facetious question. As proud as I am of the work contained in these pages, I'm also keenly aware that this book is not for every reader. So let me take a moment to describe two kinds of readers: those for whom this book was written and those for whom it was not.

Those for Whom This Book Is Meant

You ought to find this book useful if you match one or more of the following criteria:

- You want a hands-on, practical guide to using CSS in real-world projects. That's exactly what this book is all about.
- You're a hands-on learner, someone who gets a lot more out of interactive experimenting than from just reading a book. Despite the fact that this is indeed a book, it's been intentionally designed to let the reader "play along at home," as it were.
- You've been meaning to increase your CSS skills for some time now, but you keep putting it off because CSS is a large, complex subject, and you don't have a roadmap for how to get to the next level.
- You've always wanted someone to show you how to convert a typical, old-school, pure-HTML design into a pure-CSS design and to explain why it's to your advantage to do so. If that's the case, go to [Project 1](#), "Converting an Existing Page," without another moment's delay.
- If asked, you would describe your HTML skill level as "intermediate" or "expert" and your CSS skill level as "basic" or "intermediate." In other words, you understand HTML fairly well and have used enough CSS to have a basic grasp of how it's written.

Those for Whom This Book Is Not Meant

You might not find this book to be useful if one or more of the following describes you:

- You've never used or even seen CSS before. Although some basic terms are defined in the text, the assumption here is that the reader knows the basics of writing CSS and is fairly proficient with HTML authoring. Some readers of Eric Meyer on CSS said they were able to use it even though they'd hardly ever touched CSS before, but this book was not written with the beginner in mind.
- You want to understand all of the subtleties of the theory underlying CSS and grasp the nuances of the specification. There are now many books on the market that occupy that niche. The focus here is on demonstrating effects that work.
- You've only done Web design in a point-and-click editing environment. This book assumes that you can edit

 PREV

[< Day](#) [Day Up >](#)

NEXT 

 PREV

[< Day Day Up >](#)

 NEXT

What You Can Expect from This Book

From the outset, my intent has been to write an engaging, interactive book that focuses on practical and interesting uses of CSS that can be deployed in today's browsers. To do this, each project evolves from having no styles to being fully styled and ready for deployment on the Web. If I've done my job well, you should get the feeling of watching over my shoulder as I work on a project, with me commenting on what I'm doing as I do it.

Although you can simply read the text and look at the figures to get a sense of how a project is evolving, I think the best way to work along with the book is to have a Web browser and a text editor open as you read. That way, you can follow along with the changes I make in the text by physically making the same changes in your project file and seeing the changes in your own Web browser.

There is one point on which I want to be very clear: The techniques shown in this book are generally meant for browsers whose version number is greater than or equal to 5 (well, and Safari 1.0+). If you have to design a site that looks the same in Explorer 4.x and Netscape 4.x as it does in IE6.x and NS7.x, this book is probably not for you.

 PREV

[< Day Day Up >](#)

 NEXT

[PREV](#)[< Day Day Up >](#)[NEXT](#)

Project Overview

In keeping with the practical, hands-on nature of the book, I've divided it up into a series of 10 projects—each one effectively a chapter. It is possible to skip around from project to project as the spirit moves you, as each project was written to stand on its own as much as possible. However, the book was still written with the "linear reader" in mind; if you read from front to back, you should find that the projects build on one another.

With a few exceptions, the projects are titled in as self-obvious a way as possible. For example, [Project 1](#), "Converting an Existing Page," takes a page designed using only HTML markup and spacer GIFs, and converts it to a pure CSS design that uses positioning for layout instead of tables.

[Projects 2](#) and [3](#) cover some fairly basic projects, from styling a photo gallery to making a financial report look better than it would by default. [Project 4](#) increases the sophistication somewhat by showing how to use backgrounds in creative ways, and how to mix relative and absolute measures in order to set up translucency effects.

Then, in [Projects 5](#) through [7](#), the topic of discussion is using lists in various ways. The first of these three projects uses a list of links to create a sidebar menu, complete with two different layouts for the same list. The second project in the trilogy takes a series of nested lists and turns them into a dynamic set of "drop-down" menus that work in most browsers (and that includes IE/Win). The last of the three projects explores using the Sliding Doors technique to turn a list of links into a set of "tabs."

[Projects 8](#) and [9](#) consider the styling of a weblog and the styling of a home page around that weblog, respectively. [Project 10](#) is the most ambitious and complex of the book: It takes a design for the CSS Zen Garden and works through the application from design to markup. This was done by soliciting a design from the Garden's creator and then working to translate the design into a styled document. For those of you who work in the print world, we take a comp and turn it into a finished product.

[PREV](#)[< Day Day Up >](#)[NEXT](#)

 PREV

[< Day Day Up >](#)

NEXT 

Companion Web Site

Each project in this book is based on the editing of a real project file. You can download the project files either for the entire book all at once or for each chapter individually. The project files are available on the book's companion Web site: <http://more.erictayeroncss.com/>. There you will find the files that were used to produce the figures throughout the book, any errata to the book, and supplemental materials like bonus text, commentary from the author, and links to useful online resources.

For each project, there will be an archive of all the files you need to work along with the text; this includes any graphic files needed as well as a version of the project file at its outset. These files follow a consistent naming scheme; for example, the figure that corresponds to Figure 7.3 will be named ch0703.html, the Project 1 working file will be ch01proj.html. This is the file you should open up with a text editor and make changes to as the project moves forward. You can also load it into a Web browser and hit Reload at each step to see what effect the new styles have.

You'll also find the files for the first book, Eric Meyer on CSS. You can download them whether or not you've read the book, but they probably won't make as much sense if you haven't.

 PREV

[< Day Day Up >](#)

NEXT 

 PREV

[< Day Day Up >](#)

NEXT 

Conventions

This book follows a few typographical conventions that you should be familiar with before proceeding.

A new term is set in *italics* the first time it is introduced. There will often be a short definition of the term nearby. Program text, functions, variables, and other "computer language" are set in a fixed-pitch font. In regular text, it will also be a dark blue color—for example, when mentioning the property margin or a value like 10px.

Code blocks are set entirely in a fixed-pitch font. Any blue text within a code block indicates a change to the code from its previous state. Most code blocks show only a fragment of the overall document or style sheet, with the lines to be changed (or inserted) surrounded by unchanged text. This extra text provides a sense of context, making it easier to find the part you need to change if you're following along with the text. Here is an example:

 PREV

[< Day Day Up >](#)

NEXT 

 PREV

[< Day Day Up >](#)

 NEXT

Project 1. Converting an Existing Page

Look inside a typical CSS [designer's] house. What do you see? Chairs, only chairs. No tables.

—JEAN-YVES STERVINOU

For years upon years—about eight of them, as this is being written—we've been using tables and spacer images to lay out Web pages. For the first part of all those years, it was the only way to create compelling visual design. Tools grew up to support this desire, design firms embraced it wholeheartedly, and pages got more and more bloated as a result.

When CSS came along, there began to be some hope that the trend might reverse and that pages could get smaller and more meaningful. When CSS2 introduced positioning, the door was opened. It became theoretically possible to do almost everything that tables did and in a fraction of the page weight.

That was theory: The practice for at least a few years was very different, thanks to incomplete and incompatible browser implementations. That improved slowly until, by the dawn of the 21st century, positioned layouts were really held hostage only by the persistence of Netscape 4.x, and even there some simple positioning could be achieved.

A good way to get familiar with the basics of positioning layout is to take a table-driven layout and convert it to CSS positioning (CSS-P). This allows for comparisons between the document structures and serves as a primer in how basic positioning can make life a lot easier.

 PREV

[< Day Day Up >](#)

 NEXT

 PREV

[< Day Day Up >](#)

 NEXT

Project Goals

Our goal for this project is as straightforward as can be: to take an HTML-heavy design and convert it to use CSS-driven layout. In so doing, we'll explore how commonly used HTML structures and tricks can be replaced with vastly simpler markup and CSS, and how doing this makes the document markup a great deal easier to read. When we're done, we'll take some measurements to determine just how much of a savings our effort has yielded.

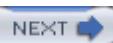
We'll assess each portion of the document as we reach it, so what approaches we'll take aren't known ahead of time. We can still articulate some general goals:

- The number of images on the page should be reduced to a minimum. This will have the dual benefit of making the document structure much cleaner and also reducing the potential number of server hits required to display the page.
- All tables intended solely for layout should be removed. When we're done, only tables that contain data appropriate for a table should remain.
- The markup that results from our conversion should have a strong structure; that is, headings should be enclosed in heading tags such as h2. Furthermore, the content should be in an order that makes sense if the page is presented with no style at all.
- The final product should look as much like the all-HTML design as possible. While there may not be a perfect pixel-for-pixel fidelity between the two, we should do our utmost to minimize any differences.

If we can fulfill all of these goals, we'll have done something fairly remarkable.

 PREV

[< Day Day Up >](#)

 NEXT

 PREV

[< Day](#) [Day Up >](#)

 NEXT

Preparation

Download the files for [Project 1](#) from this book's Web site. If you're planning to play along at home, load the file ch01proj.html into the editing program of your choice. This is the file you'll be editing, saving, and reloading as the project progresses.



See the [Introduction](#) for instructions on how to download files from the Web site.

 PREV

[< Day](#) [Day Up >](#)

 NEXT

< PREV

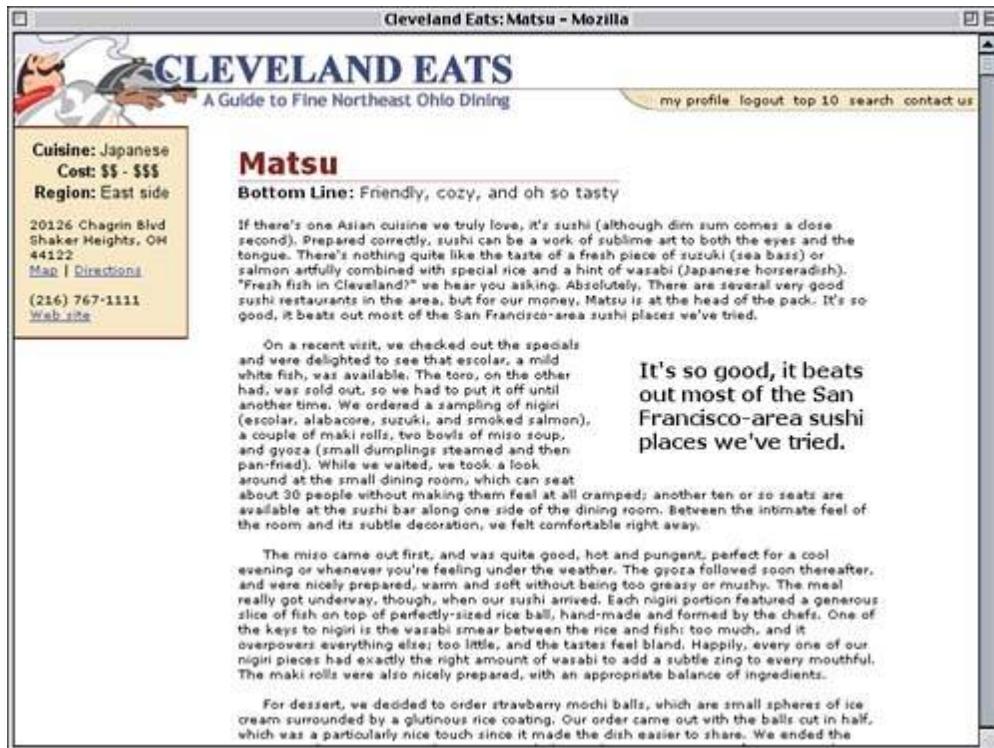
< Day Day Up >

NEXT >

Laying the Groundwork

The first thing we need to do is take a look at the existing all-HTML page in a Web browser and then look at its markup. [Figure 1.1](#) shows what the page looks like.

Figure 1.1. The page as it exists in an all-HTML format.



Now it's time to look at the HTML itself. Unfortunately, we can't provide it here because a listing of the page's source code would be about seven pages long! So we'll have to consider another approach.



You can see the source code for [Figure 1.1](#) by loading the file ch0101.html into your favorite text editor.

Instead of going through the HTML line by line, let's get a quick look at how the page has been put together. To do this, we're going to add some temporary styles to the simple style sheet that already appears at top of the document.

< PREV

< Day Day Up >

NEXT >

 PREV

[< Day](#) [Day Up >](#)

NEXT 

Converting the Document

Going from an all-HTML design to a CSS-driven layout requires two steps: shedding the HTML-based presentation and adding in CSS to replace it. In many cases, it's easier to do this in a gradual fashion by stripping down a small portion of the document and styling it before moving on to the next section. For this project, though, we're going to go the hardcore route and strip the file all the way down to its minimum state before we start styling. (It makes the figures look a lot better, too.)



You can see the fully stripped-down version of the page by loading the file ch0104.html into your favorite HTML or text editor.

Stripping Down to the Minimum

It's time to make a copy of the original all-HTML file and strip out nearly all of its HTML-based presentation. This is undeniably the toughest part of converting an all-HTML design to a CSS layout. A good deal of the work can be done using find-and-replace utilities, but there is still the need to go through and delete any leftover HTML-based presentation manually.

Tidying Up Your Markup



It's possible to use one of a number of utilities to clean up markup and in the process strip out most or all of the presentational aspects of a document. HTML Tidy is one of the most popular such tools and is available for free download from <http://tidy.sourceforge.net/>.

It should be mentioned that sometimes it's easier to just copy the text from a page and create a new structure around it, rather than trying to convert the old markup to newer, more efficient markup. We'll go through a conversion process in this project—that will help illustrate how certain kinds of table-oriented markup can be drastically simplified. Just keep in mind that it's sometimes easier to just bring across the content and build up new markup around that content.

No matter how you slim down the markup, things to eliminate include:

- font elements
-
 elements
- entities
- Any attributes on the body element (for example, text and link)

 PREV

[< Day](#) [Day Up](#) [>](#)

NEXT 

 PREV

[< Day Day Up >](#)

 NEXT

Rebuilding the Design

Because our goal is to re-create the original design using CSS for both content styling and layout, we'll want to refer to the table-driven design throughout the rest of the project. Although we won't worry about exact to-the-pixel reproduction, we'll do our best to get as close as possible to the original, starting with some basic styles and working our way through the document, styling each piece as we come to it.

Basic Styles

Before we start reconstructing the overall layout, let's set some "global" styles (that is, styles that apply to the document as a whole). The first thing to do is bring the font size back in line with how it looked in the original HTML-driven design. The HTML approach used `` to set most of the font sizes, and the closest CSS equivalent is the keyword `x-small`. Thus, we'll change the property `font-family` to `font` and drop in the size value.

 PREV

[< Day Day Up >](#)

 NEXT

 PREV

[< Day](#) [Day Up >](#)

NEXT 

Assessing the Benefits

In describing a conversion process like the one we've undertaken in this project, you might meet with some bewilderment. "Why bother, when tables already work?" is the usual question. There are two good reasons.

First, the document structure is a lot cleaner and thus a lot easier to edit and generally maintain. Let's say you had an unclosed element somewhere in your markup, and it was completely messing up the layout. Would you rather sift through the HTML design, choked as it is with font tags and tables nested inside tables, or the relatively clean structure we had at the end of the conversion process? Probably the latter.

Second, there is a noticeable savings in terms of file size. [Table 1.1](#) compares the file sizes, element counts, and server hits that are (or may be) necessary for each of three approaches: the all-HTML design, the converted document with embedded style sheet, and the converted document with the style sheet made external and linked in.

A Range of Hits



The reason that some of the "Server Hits" values are given as a range is due to repeated images. In the all-HTML design, for example, there were 18 spacer.gif images. Most browsers will only download once and use a cached version of the image from then on, but a browser without a cache or with caching turned off has to load the image every time it's referenced in the HTML.

Table 1.1. A Qualitative Comparison of the Three Approaches

Method	Size ^[*]	Characters	Server Hits	Images	Tables	Font Tags
All HTML	100%	8,994	6–23	22 (18 repeat)	7	20
HTML + CSS	75.4%	6,785	5	4	1	0
HTML + linked CSS	59.8%	5,375 + 1,435	5–6	4	1	0

[*] Compared to all-HTML method; refers to size of HTML document only

Even with the style sheet left inside the document, there's a nearly 25% reduction in the size of the file. If the style sheet is moved into an external file, the benefits are even more striking, down to just under 60% of its original size. Either way, the page will download faster, making users happier and reducing the amount of bandwidth for which we have to pay.

The 59.8% figure deserves a short explanation. It only counts the HTML, not the external style sheet. This is because browsers usually cache external style sheets after they're loaded the first time. Thus, the server won't be asked for the

 PREV

[< Day](#) [Day Up >](#)

NEXT 

 PREV

[< Day](#) [Day Up >](#)

NEXT 

Branching Out

Starting with the styles we created in this project, try to accomplish the following modifications without changing the document's structure. Notice how easy it is to do it this way instead of trying to rewrite the tables.

1.

Move the navbar into the upper-right corner of the page and put the subtitle below it so that the baselines of the subtitle and main title line up. Since that frees up some space under the masthead, move the review text upward so that you don't waste too much screen real estate.

The screenshot shows a web browser window for "Cleveland Eats: Matsu - Mozilla". The page has a header with a cartoon character logo, the title "CLEVELAND EATS", and a subtitle "A Guide to Fine Northeast Ohio Dining". A navigation bar with links like "my profile", "logout", "top 10", "search", and "contact us" is at the top right. On the left, there's a sidebar with information about the restaurant: Cuisine: Japanese, Cost: \$\$ - \$\$\$, Region: East side, address: 20126 Chagrin Blvd, Shaker Heights, OH 44122, and links to "Map | Directions", "(216) 767-1111", and "Web site". The main content area starts with a section titled "Matsu" and a "Bottom Line" summary: "Friendly, cozy, and oh so tasty.". Below this is a detailed review of the dining experience, mentioning nigiri, maki rolls, miso soup, and gyoza. Another summary sentence "It's so good, it beats out most of the San Francisco-area sushi places we've tried." is placed within the review text. The review concludes with a paragraph about the miso soup.

2.

Flip the layout around so that the sidebar is on the right and the review on the left. Remember that the chef gets cut off right where the sidebar sits now, so a visual finish of some kind will be needed. For extra credit, extend the summary's top border to touch the info box without allowing any of the text in the review to get within 40 pixels of the sidebar.

The screenshot shows the same website layout as the first one, but with the sidebar moved to the right side of the page. The main content area on the left contains the "Matsu" section, the "Bottom Line" summary, and the detailed review of the dining experience. The sidebar on the right contains the restaurant information: Cuisine: Japanese, Cost: \$\$ - \$\$\$, Region: East side, address: 20126 Chagrin Blvd, Shaker Heights, OH 44122, and links to "Map | Directions", "(216) 767-1111", and "Web site". The "It's so good, it beats out most of the San Francisco-area sushi places we've tried." summary sentence is now positioned directly next to the sidebar, aligned with the "Bottom Line" summary. The review text continues below this.

 PREV

[< Day](#) [Day Up](#) [>](#)

NEXT 

 PREV

[< Day](#) [Day Up >](#)

NEXT 

Project 2. Styling a Photo Collection

All photographs are there to remind us of what we forget. In this—as in other ways—they are the opposite of paintings. Paintings record what the painter remembers. Because each one of us forgets different things, a photo more than a painting may change its meaning according to who is looking at it.

—JOHN BERGER

Although not everyone puts his or her photographs online, such collections are an interesting layout challenge. Each photo and its associated information forms a small, self-contained unit that nevertheless has to be laid out with respect to the other photographs on the page. In a way, they're like portals, except with each "box" in this portal leading to more information about a photo instead of to the latest headlines or sports scores.

Photo collections are also reminiscent of another, far more common layout challenge: that of a catalog of products for sale via an e-commerce site. In fact, sometimes the photos themselves can be products for sale, which is the assumption we'll be making in this project.

 PREV

[< Day](#) [Day Up >](#)

NEXT 

Project Goals

In this project, we're looking for ways to present a collection of photographs for sale. Our client has given us the following requirements:

- We need to have three different possible presentations: a Contact Sheet view for the artist to check what's available and to show off to his peers, a Gallery view for users to be able to see all the offerings, and a more detailed Catalog view to allow for ordering.
- In the Gallery and Contact Sheet views, as many photographs as possible should appear "above the fold" and without requiring a horizontal scroll, no matter the browser window size. It is acceptable to show only the photo and its title in this view. However, the pictures should arrange themselves into a regular grid.
- In the Catalog view, every photograph should be presented along with its title, catalog number, and price. Scrolling is not a problem in this view.
- The same markup should drive all three views because our client doesn't want to pay for a dynamic site and therefore wants the page markup to be produced only once.

For this project, we're only working on the photo collection piece of the layout, so we don't have to worry about anything but that piece. We will assume that the layout will go into a main central column in a larger layout, but that doesn't really change anything for this project.

Due to the constraints of the project, particularly those of the Gallery and Contact Sheet views, we won't be able to use tables to lay out these photos. Why not? Because of the request to get as many pictures as possible "above the fold" (that is, into the browser window at page load).

So, instead of tables, we'll need to float the pictures and their information for those two "compact" views. Floating them will allow us to get as many pictures in each "row" as will fit in the browser window. In other words, a user with an 800x600 browser window might get four images per row, while a 1280x1024 user will get six or seven. Using floats allows for this kind of "flow" behavior, whereas using tables does not. As an added bonus, we can set up the floats so that each one is the same width. This will ensure that they lay themselves out in a grid-like fashion.

 PREV

[< Day Day Up >](#)

NEXT 

Preparation

Download the files for [Project 2](#) from this book's Web site. If you're planning to play along at home, load the file ch02proj.html into the editing program of your choice. This is the file you'll be editing, saving, and reloading as the chapter progresses.



See the [Introduction](#) for instructions on how to download files from the Web site.

 PREV

[< Day Day Up >](#)

NEXT 

 PREV

[< Day](#) [Day Up >](#)

NEXT 

Laying the Groundwork

The first thing we ought to do is take a look at the markup with which we'll be working. Here are the first two sets of images and information in the document:

 PREV

[< Day](#) [Day Up >](#)

NEXT 

 PREV

[< Day Day Up >](#)

 NEXT

Creating the Contact Sheet View

At this stage, we can see the document structure itself: images followed by unordered lists. What we're after at this point is the creation of a "contact sheet" layout, in which the images are all laid out in a grid. This will let us see as many images as possible at once.

Floating Away

Since we aren't using a table, the obvious solution is to float the images. We know the images are no more than 128 pixels wide or tall, so we'll make our divs 128x128 and give them a white background and black border. Speaking of borders, we want to get rid of the blue link-border around the images as well.

Border Incidents



Not all browsers will put a border around a linked image, but some will, so it's a good idea to explicitly get rid of the border. It won't hurt in browsers that didn't have a border in the first place.

 PREV

[< Day Day Up >](#)

 NEXT

 PREV

[< Day Day Up >](#)

NEXT 

Creating the Gallery View

As interesting as the preceding style sheet may be, the drawback is that it doesn't show us what photos are called. This might be a perfect presentation for the artist himself, who probably knows what they're all called and just wants to see everything in a compact form. For visitors, though, it won't be nearly as useful. Therefore, let's turn our slides into a gallery of photos, each one captioned with its title.

Removing the Slide Styles

To clear the field, so to speak, we'll want to drop the styles that place the background images into place (see [Figure 2.7](#)). This leaves us with the style sheet in [Listing 2.2](#).

Listing 2.2. The Reduced Style Sheet

 PREV

[< Day Day Up >](#)

NEXT 

 PREV

[< Day Day Up >](#)

NEXT 

Creating the Catalog View

Now that we've seen how to create flowing grids of pictures, let's consider a different approach. This time, we'll set up a vertical listing of images, next to each of which will be placed the title, catalog number, and price of the image. To do this, we'll toss out nearly all of the styles we had before and start over. The only things we'll keep are the body and footer styles, as shown in [Listing 2.4](#). We're basically back to where we were at the beginning of the project (see [Figure 2.1](#) for an illustration).

Listing 2.4. Starting Nearly from Scratch

 PREV

[< Day Day Up >](#)

NEXT 

 PREV

[< Day](#) [Day Up](#) [>](#)

NEXT 

Branching Out

Try re-creating the following changes to the work in this project.

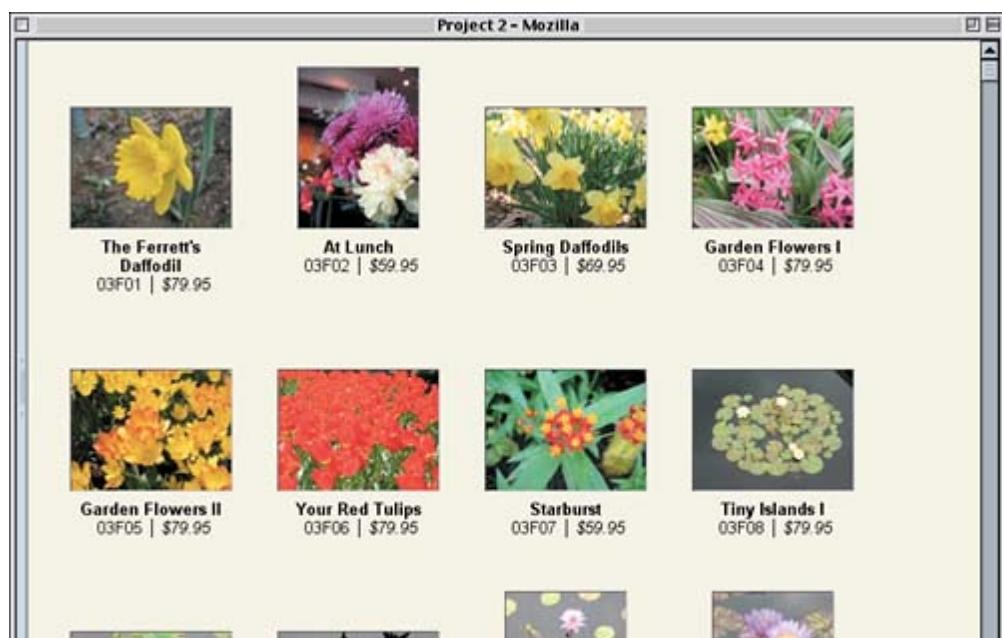
1.

In the Contact Sheet view, try placing the title right below the picture, looking as if it were written onto the lower part of the slide frame. To do this, you'll need to remove the bottom margin from the images without throwing off the overall layout. Note that this will not be possible for portrait images since CSS is not able to rotate text, so you'll need to constrain your styles accordingly. Note also that long titles might flow off the slide, so a property like overflow might be useful.



2.

In the Gallery view, add the catalog number and price back into the layout, but put them next to each other instead of one on top of the other. This will enhance the Gallery view without significantly changing the layout. Remember that a little extra height may be needed.



 PREV

[< Day](#) [Day Up >](#)

NEXT 

 PREV

[< Day Day Up >](#)

 NEXT

Project 3. Styling a Financial Report

The genius of capitalism consists precisely in its lack of morality. Unless he is rich enough to hire his own choir, a capitalist is a fellow who, by definition, can ill afford to believe in anything other than the doctrine of the bottom line.

—LEWIS H. LAPHAM

Although we're used to thinking of the Web as a collection of personal sites, e-commerce powerhouses, fringe groups, and nuggets of information, there's a lot more going on—albeit not all in the public eye. Corporate intranets, for example, are often awash in HTML-based information ranging from employee portals to Webmail interfaces and beyond.

Although an entire book could be devoted to exploring how just a sampling of this information could be styled, it will be sufficient here to examine one type of information that tends to be particular to corporate sites: a financial report showing profit and loss in a number of markets in a given quarter. We won't be seeking to replace the table markup with divs and CSS—that would be silly. Since we have a table of data, it will be contained in table markup. What we'll do is add useful hooks to the structure and use those to style the table and its contents.

 PREV

[< Day Day Up >](#)

 NEXT

Project Goals

Our goal for this project is to take a table of financial information and style it for both screen and print output while adding as little as possible to the markup. We'll accomplish this mainly by adding class and ID hooks to the elements already in place.

For this project, upper management has given us a detailed set of design goals:

- The Totals row at the bottom of the report should stand out visually, preferably by boldfacing the figures.
- The labels (both across the top of the report and down the side) should be right-justified and should be separated from the figures by gray lines. The top labels should be separated by gray vertical lines.
- In a browser, each line of the report should be separated from the others by a light gray line. In print, every other row should have a light gray background, and columns should be separated by light gray vertical lines.
- Any negative number should appear as red with a yellow background in a Web browser and as italicized in print.
- The profits (in the rightmost column) need to be visually highlighted in some fashion.
- All dollar amounts should line up at the decimal point, and all others should have their thousands-separator commas line up.

In addition, we've been given the vague (but useful) directive to "make things look good." So, with those goals in mind, let's get things underway.

 PREV

< Day Day Up >

NEXT 

Preparation

Download the files for [Project 3](#) from this book's Web site. If you're planning to play along at home, load the file ch03proj.html into the editing program of your choice. This is the file you'll be editing, saving, and reloading as the project progresses.



See the [Introduction](#) for instructions on how to download files from the Web site.

 PREV

< Day Day Up >

NEXT 

 PREV

[< Day Day Up >](#)

 NEXT

Styling for the Screen

Since all of the users will be accessing the reports via a Web browser, we'll start by creating screen styles. During this phase, we'll leave aside the print-style design goals and just focus on what will be put onto the monitor.

Laying the Foundations

As the basis for our work, we'll be using a report table that's been created for us by the database guys. Its markup is, as we requested earlier, as simple and unadorned as possible. The first few rows of the table are shown in [Listing 3.1](#), and the whole table in its default rendering is shown in [Figure 3.1](#).

Listing 3.1. The Table's Beginning

 PREV

[< Day Day Up >](#)

 NEXT

 PREV

[< Day Day Up >](#)

 NEXT

Styling for Print

Now that we've created a style sheet for the screen, it's time to turn our attention to the print styles defined in the project goals. The goals specifically related to print are as follows:

- In print, every other row should have a light gray background, and light gray vertical lines should separate columns.
- Any negative number should be italicized in print.

These are in addition to the other project goals, but those have already been satisfied by the style sheet we've written. All we really need to do is override and adjust styles to optimize them for print.

Starting Out

The first thing to do is set up a second style sheet. Although it's eventually intended for print styling, this is what we'll add after the previous style sheet:

 PREV

[< Day Day Up >](#)

 NEXT

 PREV

[< Day](#) [Day Up >](#)

NEXT 

Branching Out

Try creating styles to accomplish each of the variations described here. If the markup needs to be changed to make the variation easier or even possible, it will be noted in the text.

1.

Come up with a visual highlight for the market names that is distinct from the profit figures. If you use the markup from the file shown in [Figure 3.9](#), this will require a change to the markup in the form of adding some information. If you use the markup from the file in [Figure 3.13](#), no markup has to be changed.

Project 3 - Mozilla

Northeast U.S.							
Fall 2003							
Market	Units Sold	Revenue	Units Returned	Units Damaged	Loss	Expenses	Profit
Boston	21,223	\$317,283.85	2,761	2,968	\$85,657.41	\$295,830.50	-\$64,204.06
Cleveland	23,544	\$351,982.90	1,832	1,128	\$44,247.19	\$216,411.76	\$91,323.85
Chicago	19,437	\$290,583.15	2,841	2,592	\$81,223.68	\$305,189.00	-\$95,829.53
Cincinnati	22,983	\$343,595.85	1,641	1,934	\$53,457.49	\$286,294.98	\$3,843.38
Detroit	17,238	\$257,708.10	2,716	672	\$50,649.04	\$244,984.65	-\$37,925.59
Milwaukee	28,649	\$428,302.55	2,458	862	\$49,631.41	\$198,592.65	\$180,078.49
New York City	47,128	\$704,563.60	2,802	2,617	\$81,013.00	\$501,983.00	\$121,567.60
Philadelphia	20,057	\$299,852.15	767	2,560	\$49,750.52	\$379,058.49	-\$128,956.86
Pittsburgh	20,992	\$313,830.40	1,492	2,556	\$60,520.33	\$237,459.20	\$15,850.87
Washington DC	31,569	\$471,956.55	1,361	1,765	\$46,736.05	\$458,230.54	-\$33,010.04
Totals	252,820	\$3,779,659.00	20,673	19,654	\$602,886.12	\$3,124,034.77	\$52,738.11

Date

2.

"Fill in" the top row and left column and create a box around the whole table of headings and figures. You should be able to do this in a number of ways, but for a challenge, try to find one that doesn't simply set a border on the table element itself.

Project 3 - Mozilla

Northeast U.S.							
Fall 2003							
Market	Units Sold	Revenue	Units Returned	Units Damaged	Loss	Expenses	Profit
Boston	21,223	\$317,283.85	2,761	2,968	\$85,657.41	\$295,830.50	-\$64,204.06
Cleveland	23,544	\$351,982.90	1,832	1,128	\$44,247.19	\$216,411.76	\$91,323.85
Chicago	19,437	\$290,583.15	2,841	2,592	\$81,223.68	\$305,189.00	-\$95,829.53
Cincinnati	22,983	\$343,595.85	1,641	1,934	\$53,457.49	\$286,294.98	\$3,843.38
Detroit	17,238	\$257,708.10	2,716	672	\$50,649.04	\$244,984.65	-\$37,925.59
Milwaukee	28,649	\$428,302.55	2,458	862	\$49,631.41	\$198,592.65	\$180,078.49
New York City	47,128	\$704,563.60	2,802	2,617	\$81,013.00	\$501,983.00	\$121,567.60
Philadelphia	20,057	\$299,852.15	767	2,560	\$49,750.52	\$379,058.49	-\$128,956.86
Pittsburgh	20,992	\$313,830.40	1,492	2,556	\$60,520.33	\$237,459.20	\$15,850.87
Washington DC	31,569	\$471,956.55	1,361	1,765	\$46,736.05	\$458,230.54	-\$33,010.04
Totals	252,820	\$3,779,659.00	20,673	19,654	\$602,886.12	\$3,124,034.77	\$52,738.11

 PREV

[< Day](#) [Day Up](#) [>](#)

NEXT 

 PREV

[< Day Day Up >](#)

 NEXT

Project 4. Positioning in the Background

But I was told there would be no math! Too bad.

—MACINTOSH TECHNICAL NOTE #31

It's a common thing, at least in print design, to use shaded variations of a background to make portions of the design stand out. A good example is an ad in which there's a big picture of a mountain or beach or beautiful woman filling the entire ad, and in the middle is some compelling yet meaningless text, and surrounding that text is a region where the picture in the background has been washed out, as if the text were written on a half-opaque block of plastic.

Since opacity styles aren't part of CSS as of this writing, it's been generally thought that such effects are effectively impossible. There are fixed-attachment backgrounds (see Project 11 in Eric Meyer on CSS for more details), but they aren't supported by Explorer for Windows. One can use semi-opaque PNG graphics, but they aren't supported by Explorer for Windows. In fact, short of hacking the browser with proprietary behavior scripts, there's only one way to get a smooth translucency effect in Explorer for Windows, and that's by manipulating the position of background images.

 PREV

[< Day Day Up >](#)

 NEXT

 PREV

[< Day Day Up >](#)

NEXT 

Project Goals

We've taken on a project in which a local author is publishing some of his short essays and wants them to look artistic. He's a big fan of translucency effects, so he wants to see them used in his designs. Specifically:

- We are to use a sunrise picture for the first essay, "Mourning in Mansfield." This will include a dark shade over the background behind the title and a lightening effect over the background behind the essay's main text.
- For the essay "Gathering Stormclouds," we'll be using a picture of clouds at sunset. For this one, the title will have a lightening effect for the background, while the main text will have a darkened background and light text.

The author is supplying the images, so fortunately we don't have to worry about acquiring them. All we really have to do is pull a little sleight of style to get the translucency effects our client has requested.

 PREV

[< Day Day Up >](#)

NEXT 

 PREV

< Day Day Up >

NEXT 

Preparation

Download the files for [Project 4](#) from this book's Web site. If you're planning to play along at home, load the file ch04proj.html into the editing program of your choice. This is the file you'll be editing, saving, and reloading as the project progresses.



See the [Introduction](#) for instructions on how to download files from the Web site.

 PREV

< Day Day Up >

NEXT 

 PREV

[< Day Day Up >](#)

NEXT 

Style at Dawn

For the first half of this project, we'll take the first of the two documents and add the styles necessary to create translucent effects using ordinary JPEG images. As we'll soon see, the images in question make it fairly easy to create an attractive design.

Getting Started

As usual, our first step should be to look at the structure of the document. As we can see from [Listing 4.1](#), there isn't a whole lot to it really, just a masthead div with a heading and a "main" div containing the text of the entry. Without any styles, we get the very plain rendering shown in [Figure 4.1](#).

Listing 4.1. The Basic Document Structure

 PREV

[< Day Day Up >](#)

NEXT 

 PREV

[< Day](#) [Day Up >](#)

NEXT 

Beached Styles

In this half of the project, we'll take a new document with a slightly different structure and give it a translucent-effect makeover. We'll use the principles explored in the first half of the project, but we'll use them in a more sophisticated way because in this document none of the elements will line up along the top edge.

Assessing Structure and Style

As always, our first order of business is to scout out the lay of the land, so to speak, by examining the document structure and any styles that may already exist. The basic structure looks like this:

 PREV

[< Day](#) [Day Up >](#)

NEXT 

 PREV

[< Day](#) [Day Up >](#)

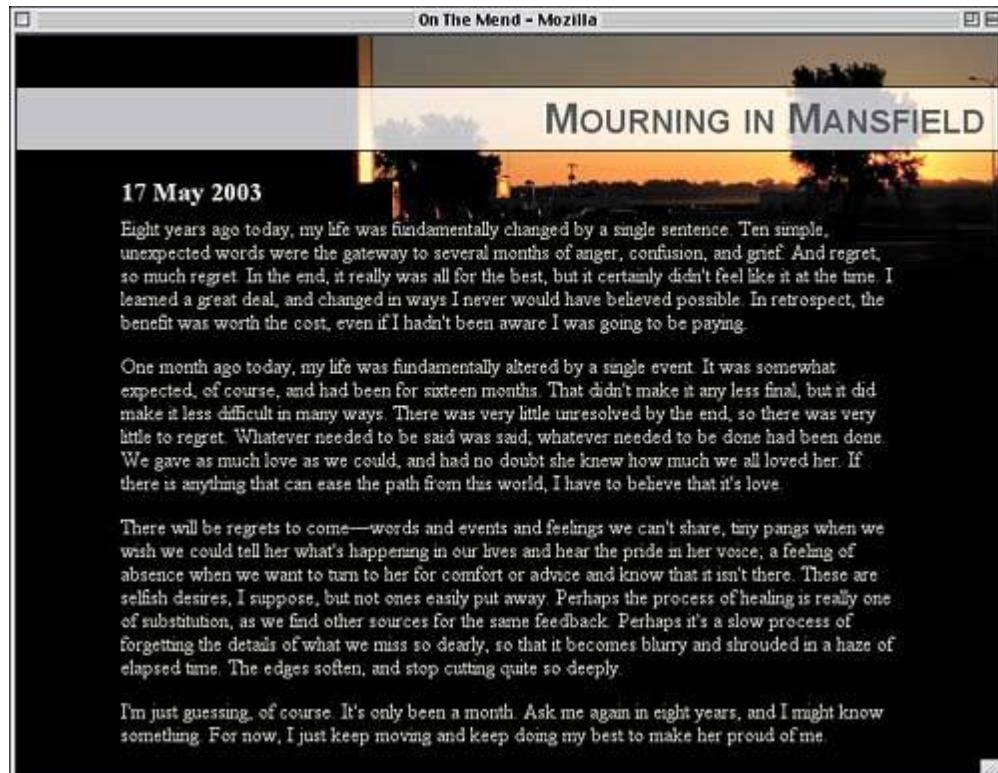
NEXT 

Branching Out

Try creating styles to accomplish each of the variations described here. If the markup needs to be changed to make the variation easier or even possible, it will be noted in the text.

1.

In the "Mansfield" design, try changing the backgrounds (and background colors) around. For example, you could use the washed-out background for the title, the faded image for the masthead, and the basic image for the body. Don't forget to change the colors, too!



2.

In the "stormclouds" design, change the placement of the background images so that they're all centered horizontally within their elements. Make sure they all continue to line up with each other.



 PREV

[< Day](#) [Day Up >](#)

NEXT 

 PREV

[< Day Day Up >](#)

 NEXT

Project 5. List-Based Menus

In America, even your menus have the gift of language .

Oh, those menus. In America, they are poetry.

—LAURIE LEE

There has been a movement in the CSS design community toward an increased use of unordered lists to contain, well, just about everything. Although this has on occasion been taken a little too far, a very common technique nowadays is to take collections of links (sometimes called menus) and place them inside lists, with one link per list item.

Why is this such a popular approach? There are a few reasons. The most important is that, when you have a list of links, it makes a great deal of sense to enclose them in a list. From the semantic-markup point of view, it's a pretty close match.

From a styling point of view, there are some major benefits as well. Because each list item contains a link, two different elements (li and a) can be styled independently. Since the basis of styling is elements, the more elements you have to work with the better.

 PREV

[< Day Day Up >](#)

 NEXT

 PREV

[< Day Day Up >](#)

 NEXT

Project Goals

After doing a great deal of work on various client projects, it seems like a good time to take a breather and work on a personal project. To that end, we'll work on enhancing the presentation of the sidebar links in a personal journal. Let's define some basic design directions and see where they take us.

- We'll be starting with a page that already has some styles, so the menu's styles need to fit in with the presentation that already exists.
- The links in the menu should be visually separated from one another; that is, we don't want a list of links with no separators or other visual effects.
- We should come up with a design that makes the menu feel open and airy so that the links seem to be a part of the other content in the design.
- We should come up with another design that encloses the links in a box or some other visual device that obviously separates them from the main content.

With these goals in mind, it's time to get set up and start styling!

 PREV

[< Day Day Up >](#)

 NEXT

 PREV

< Day Day Up >

NEXT 

Preparation

Download the files for [Project 5](#) from this book's Web site. If you're planning to play along at home, load the file ch05proj.html into the editing program of your choice. This is the file you'll be editing, saving, and reloading as the project progresses.



See the [Introduction](#) for instructions on how to download files from the Web site.

 PREV

< Day Day Up >

NEXT 

< PREV

< Day Day Up >

NEXT >

Laying the Groundwork

Since we're going to focus on ways to style an unordered list of links within an existing design, we'll start with some CSS already applied. The starting style sheet is provided in [Listing 5.1](#) and illustrated by [Figure 5.1](#).

Figure 5.1. The design as it now stands.



Second Time Around



Readers of Eric Meyer on CSS may recognize the design in this project because it's taken from that book's [Project 9](#), "Multicolumn Layout." The difference here is that the links are now contained in an unordered list, and the entry text is from another date.

Listing 5.1. The Starting Styles

< PREV

< Day Day Up >

NEXT >

 PREV

[< Day Day Up >](#)

 NEXT

Open and Airy

For our first approach, we'll create a set of styles that place visible separators between some spread-out links. That is, we'll push the links apart vertically and then drop separators in between each link.

Separation

The first thing we'll do is spread apart the links by padding the list items that contain them. A little top and bottom padding should do the trick nicely.

Uneven Padding?



We've made the padding uneven on purpose. Because of the way English text is formed, there generally tends to be more apparent space underneath text than above it. Thus, we've added more padding above than below. This is one design decision. Others are possible, and in cases where the text is all uppercase, top and bottom padding should probably be symmetric.

 PREV

[< Day Day Up >](#)

 NEXT

 PREV

[< Day](#) [Day Up >](#)

NEXT 

Enclosing the Links

Since we've experimented with styles that give the sidebar links an open, airy appearance, let's try a different approach: enclosing the links in a box and making them into button-like objects. To do this, we'll return to an earlier version of the project and build from there.

Changes

We're going to replicate the file we had at the time [Figure 5.3](#) was created and make five changes. First, we'll take the border off the h4 and change its text color so that the rule looks like this:

 PREV

[< Day](#) [Day Up >](#)

NEXT 

 PREV

[< Day](#) [Day Up >](#)

NEXT 

Branching Out

With the elements we have available, there are almost too many choices for ways to style our menu. Here are a few ideas to get you thinking:

1.

Change the hover effect from the first half of the project so that it encloses the whole link in a border but without causing any offset to the left when hovering. The background image should also move to the right side of the link. While you're at it, tie the sidebar heading and the main entry heading together visually by using negative margins and add in some padding to get the sidebar title's text to right-align with the double border on the right side of the sidebar.



2.

Rewrite the menu styles from the second half of the project so that the links appear to be written on the background but are surrounded by a dark shade similar to that used in the masthead. Thus, the links should have a background that matches the body and should be "connected" to it with lines of color that match both links and body.



 PREV

[< Day](#) [Day Up >](#)

NEXT 

 PREV

[< Day Day Up >](#)

 NEXT

Project 6. CSS-Driven Drop-Down Menus

Parents who want a fresh point of view on their furniture are advised to drop down on all fours and accompany the nine or ten month old on his rounds.

—SELMA H. FRAIBERG

As we saw in the preceding project, it's possible to take a simple, unordered list of links and make it look good. The missing component there was the capability to have submenus and even sub-submenus. Is there a way to use CSS and simple HTML to create drop-down menus, and menus within menus?

As you might have expected, the answer is "yes." It does require using a bit of proprietary technology to get one browser on board, but since that browser is Internet Explorer for Windows, the nonstandard bit is very likely more than worth it. As we'll see in this project, we can take simple nested lists of links and turn them into multilevel menu systems.

 PREV

[< Day Day Up >](#)

 NEXT

 PREV

[< Day Day Up >](#)

 NEXT

Project Goals

Our basic goal is simple: to create a multilevel menu system using nothing more than some unordered lists and CSS. Embedded within that overall goal are a number of more specific goals:

- The menu system we create needs to be able to handle multiple levels of drop-downs so that we can have menus spawn from menus.
- It should be easy to set up our menus as either sidebars or horizontal toolbars. This should also be possible without changing the markup at all.
- If we can avoid using JavaScript, we will. In other words, we want to have the menus dynamically shown and hidden via CSS alone. Doing so will reduce page load since the browser will only have to load the page and the style sheet.
- The menus need to work in as many browsers as possible, and that includes IE/Win.

It might sound like we have our work cut out for us, but the whole process will be simpler than you might believe right now. Most of what we have to do is lay out the menus how we want. Once that's done, we'll just add in a few extra bits to make the menus appear and disappear dynamically, and we'll be good to go!

 PREV

[< Day Day Up >](#)

 NEXT

 PREV

< Day Day Up >

NEXT 

Preparation

Download the files for [Project 6](#) from this book's Web site. If you're planning to play along at home, load the file ch06proj.html into the editing program of your choice. This is the file you'll be editing, saving, and reloading as the project progresses.



See the [Introduction](#) for instructions on how to download files from the Web site.

 PREV

< Day Day Up >

NEXT 

 PREV

[< Day Day Up >](#)

 NEXT

Laying the Groundwork

The first thing to consider is the way the document is structured. In this case, there are three major components to the document:

- The page's title
- The navigation links
- The main content of the page

As the ordering of the bullet points suggests, the navigation links are contained in a div that comes after the h1 that contains the page title but before the main div that holds the text content of the page. The file already has some basic styles, which are shown in [Listing 6.1](#) and illustrated by [Figure 6.1](#).

Listing 6.1. The Basic Styles

 PREV

[< Day Day Up >](#)

 NEXT

 PREV

[< Day Day Up >](#)

 NEXT

Laying Out the Menus

Before we get to the point of having the submenus appear and disappear, we're going to style them to look decent. In other words, we'll get the menus and submenus all styled and then take the steps necessary to make them appear and disappear in response to the user's actions.

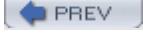
Planning Ahead

Our first step will be to call a behavior file using the Windows Internet Explorer-only property behavior. This lets us bring in a separate file that will add capabilities to IE/Win that it doesn't ordinarily have.

 PREV

[< Day Day Up >](#)

 NEXT

 PREV

[< Day](#) [Day Up](#) [>](#)

 NEXT

Reorienting the Menus

So far, we've managed to create a good multilevel menu system in which each submenu opens up to the right of its parent. That's great if the navigation is going to be in the sidebar, but what if it isn't? What if we want the main-level navigation to be a horizontal toolbar across the top of the page? Good news: With some small style adjustments and a few extra rules, we can do exactly that with the same markup we're already using.

Reorientation

Before anything else, let's rework the styles so that the main-level links are shown as a horizontal toolbar. This will require very few changes, as it happens. The first thing to do is strip out the width declarations from the div#nav and div#nav ul rules.

 PREV

[< Day](#) [Day Up](#) [>](#)

 NEXT

[PREV](#)[< Day Day Up >](#)[NEXT](#)

For Your Consideration

There are some things to bear in mind if you're going to implement CSS-driven menus like those explored in this project. Leaving aside wider questions like whether or not drop-down menus are a good user interface (some say yes, others no), the actual behavior of these menus should be considered.

Because the appearance of submenus depends on list items being in a hover state, it's fairly easy for users to accidentally let the menus snap closed. The instant the mouse pointer moves outside any of the menus, all of the currently visible submenus will disappear. The same is often true of JavaScript-driven menus, but in JavaScript you can build in delay timers that give the user a small interval of time to move the mouse back inside the menus before they go away. CSS offers no capability to define such delay timers and is not likely to do so any time soon (if ever).

For this reason, it is imperative that submenus at least sit adjacent to their parent list items, if not actually overlap them by a small amount. If a gap even a pixel wide appears between a menu entry and its submenu, there is a high probability that the mouse pointer will exit the hover state while moving through that gap and cause any open submenus to disappear. Such a situation would basically make it impossible to use some (or all) of the submenus.

Of course, it is possible to use the techniques presented here to define the appearance of menus and then use JavaScript to drive the dynamic behaviors—that is, the showing and hiding of menus, any delay timers to make the menus more usable, and the event handling that glues it all together. Such a technique is beyond the scope of this book, but in essence it would involve leaving out the display: block; rule and behavior declaration, relying on JavaScript to take the place of those styles.

As to which is better, again, there is controversy that we do not presume to resolve here. Some say that behavior should not be driven by CSS, which is a presentation language, and should instead be handled by a scripting language like JavaScript. Others feel that this sort of behavior falls within the realm of what CSS covers, and so there's nothing wrong with such techniques. It is expected that the debate will continue for a long time to come, so authors need to make up their own minds about which they prefer and act accordingly.

[PREV](#)[< Day Day Up >](#)[NEXT](#)

 PREV

[< Day](#) [Day Up >](#)

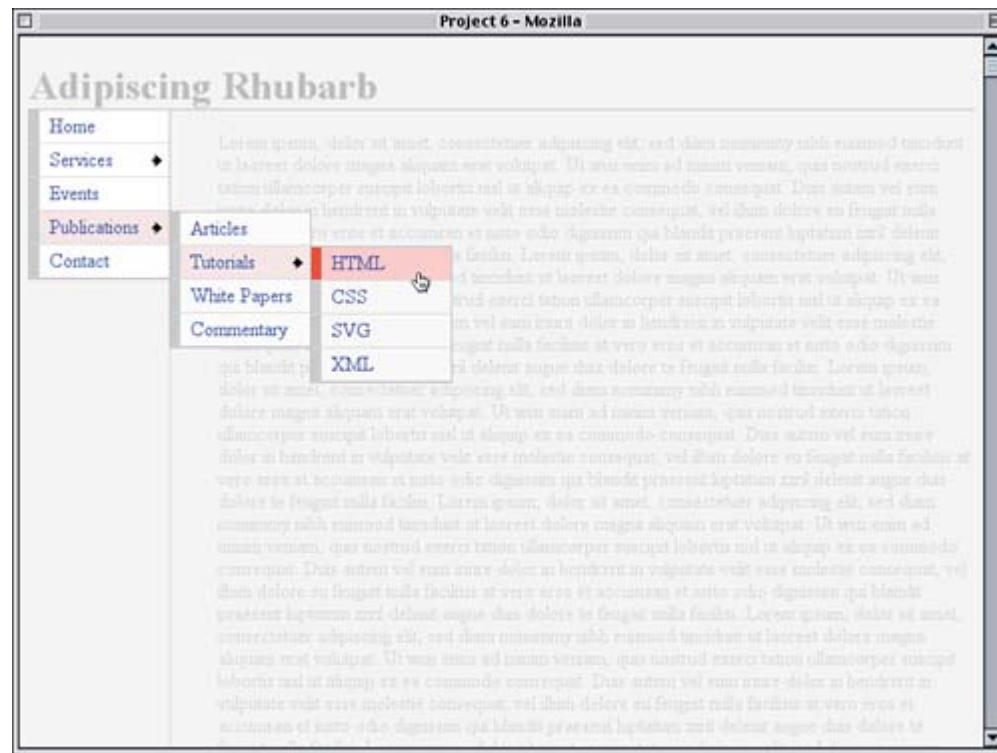
NEXT 

Branching Out

Now that you know how to create drop-down menus using CSS, try sprucing them up in the following ways:

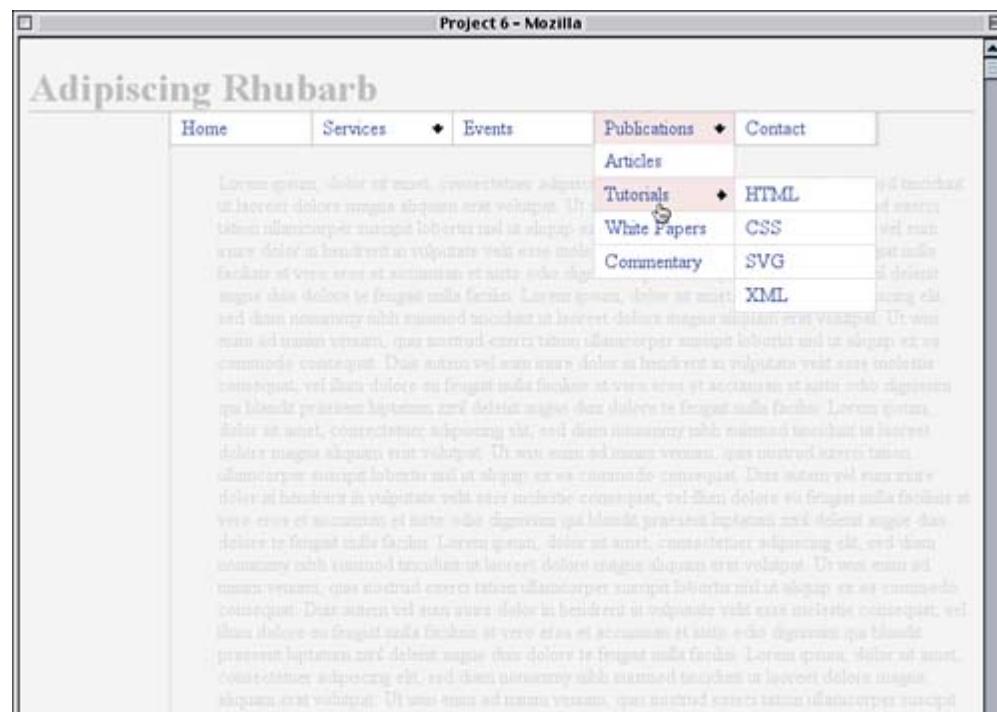
1.

Take the style sheet from the first half of the project and expand it so that the entries have a gray stripe along their left edges, and do so without using a background image. The stripe should change to red for the menu entry currently being hovered. While you're at it, make the background of each level of submenu slightly darker.



2.

Extend the styles from the second half of the project so that gray borders separate the top-level links. Here's the challenge: Do it without changing the visual appearance of the submenus in any way.



 PREV

[< Day](#) [Day Up](#) [>](#)

NEXT 

 PREV

[< Day Day Up >](#)

 NEXT

Project 7. Opening the Doors to Attractive Tabs

Welcome evermore to gods and men is the self-helping man. For him all doors are flung wide: him all tongues greet, all honors crown, all eyes follow with desire.

—RALPH WALDO EMERSON

With the trend toward using lists to represent "menus" (collections of links), there has been a good deal of interest in setting up not only link sidebars, but also horizontal rows of links. A good example of such links would be the ones across the top of Amazon.com or the Apple web site. In fact, such collections often are made to look like small tabs, although this isn't always the case.

For some time, translating an unordered list into a row of tabs meant that they were, well, kind of boxy. Not to mention a bit boring. This was the case because the tabs were usually created by setting borders on the list items or the links themselves. That's good for a basic design, but if you want a professional look, the result is rather lacking.

So, in late 2003, Douglas Bowman (<http://www.stopdesign.com/>), perhaps best known for his table-free redesign of Wired News, pioneered a new approach that allowed authors to create tabs just about as visually stunning as their imaginations would allow. He first described this technique in the article "Sliding Doors of CSS" (<http://alistapart.com/articles/slidingdoors/>) and followed it up with another article that dug more deeply into the technique. In this project, we'll be exploring some variants on Doug's original idea, but the core idea is all his.

 PREV

[< Day Day Up >](#)

 NEXT

 PREV

[< Day Day Up >](#)

NEXT 

Project Goals

Thanks to some outstanding efforts from the sales team, we've landed a contract to design a site for a new publishing house called "New Writers." Shaking off an inexplicable feeling of déjà vu, we consider the goals for this phase of the project.

- We need a template showing how the persistent navigation elements will look. This template will use some basic styles that have already been worked out, particularly for the page title.
- The navigation will be contained in an unordered list, but it needs to be presented as a horizontal row of buttons or tabs situated between the page title and the main content.
- The buttons/tabs need to be visually attractive. The client has seen plenty of CSS-driven buttons that end up as plain rectangles with a background color, and he wants something better for this site.
- Whatever look we devise for the buttons/tabs, they need to be easy to update or change in case the site's design changes.

So we need to concentrate on turning the navigation elements into something pretty. This isn't the entire contract, of course; a full site design takes a lot more work than that. For the purposes of this project, though, it's enough to concentrate on creating a basic template for styling the navigation elements.

 PREV

[< Day Day Up >](#)

NEXT 

 PREV

< Day Day Up >

NEXT 

Preparation

Download the files for [Project 7](#) from this book's Web site. If you're planning to play along at home, load the file ch07proj.html into the editing program of your choice. This is the file you'll be editing, saving, and reloading as the project progresses.



See the [Introduction](#) for instructions on how to download files from the Web site.

 PREV

< Day Day Up >

NEXT 

< PREV

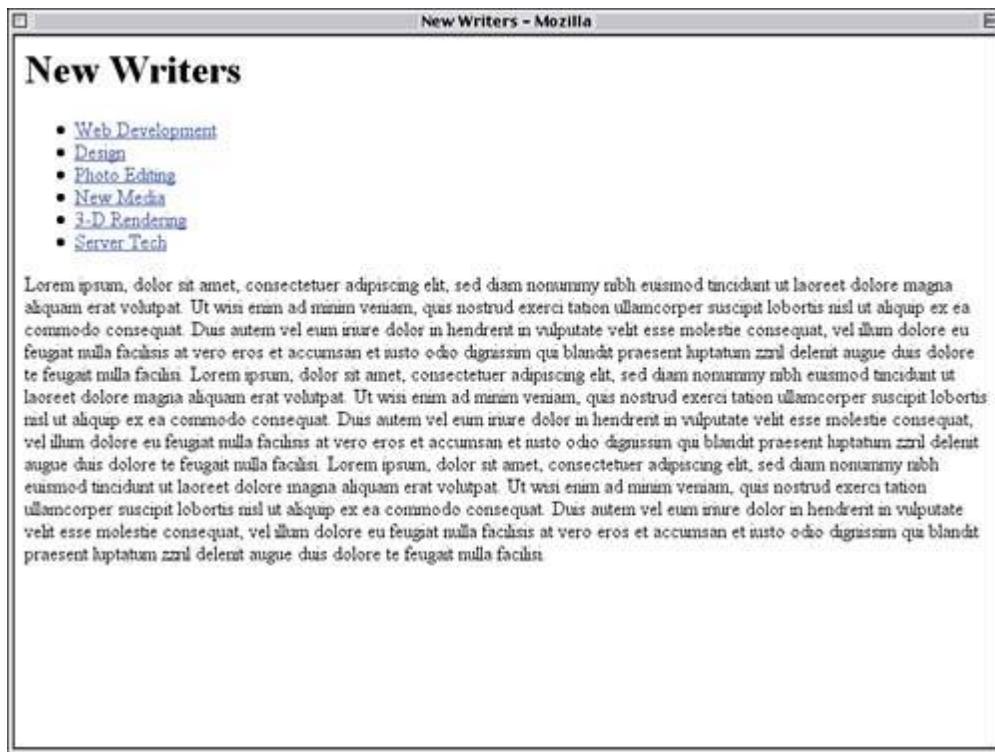
[< Day](#) [Day](#) [Up >](#)

NEXT >

Laying the Groundwork

The template page for our project is pretty bare, as [Figure 7.1](#) reveals.

Figure 7.1. The unstyled template document.



Before we get to work on the navigation elements, let's style the rest of the template. That will give us a framework for our button styles.

The first step is to style the overall document itself, in effect writing "global" styles that will apply to the document and all its constituent elements. We'll do this by removing any "gutter space" around the edges of the document, setting foreground and background colors, and defining a list of possible fonts for the html and body elements.

< PREV

[< Day](#) [Day](#) [Up >](#)

NEXT >

 PREV

[< Day Day Up >](#)

NEXT 

Styling the Links

Having set up the rest of the page, we're ready to tackle our navigation. We can see it in [Figure 7.4](#), sitting there as an unordered list wedged between the title and main content. Frankly, it's rather ugly, and it's up to us to fix it.

Starting the Styles

There are two ways to take a list and make it into a horizontal set of links. One is to make the list items inline, and the other is to float them. The two approaches both have benefits and drawbacks, but one of the major benefits of the floating approach is that it lets us create graphically beautiful buttons that work in modern browsers—as well as Internet Explorer!

Let's consider the markup for the navigation elements, shown in [Listing 7.1](#).

Listing 7.1. The Navigation Markup

 PREV

[< Day Day Up >](#)

NEXT 

 PREV

[< Day](#) [Day Up](#) [>](#)

NEXT 

Creating Actual Tabs

Although the link styles we've created are nice in their own way, they share something in common with most links these days. They're boxy and somewhat boring. We need some more attractive link styles—some nice rounded-corner tabs, for example, with smooth highlighting and shading.

As you already guessed, we can do just that, and we won't even need to change the HTML markup to do it. All we need is some modified CSS and a large (yet compact) image.

Making Some Changes

Before we get to the really pretty stuff, let's make a few changes to the style sheet in [Listing 7.2](#). The first change is to make the document background white instead of a pale gray-blue.

 PREV

[< Day](#) [Day Up](#) [>](#)

NEXT 

 PREV

[< Day](#) [Day Up](#) [>](#)

NEXT 

Branching Out

Now that we've seen how to create buttons and tabs with simple markup, here's your chance to take the concepts even further.

1.

Try adapting the Sliding Doors technique to a vertical list of links that can be as tall or short as you prefer. Note that, in this case, it isn't as big a deal if links wrap to two or more lines.



2.

Create your own tab effect and implement it using the same markup and techniques explored in this project. The more creative, the better!



 PREV

[< Day](#) [Day Up >](#)

NEXT 

 PREV

[< Day Day Up >](#)

NEXT 

Project 8. Styling a Weblog

Quiet few days well not really, but I'm not telling you lot. Which is obviously the point of a weblog.

—DAVE WHYTE

For whatever reason, personal, Web-based journals have come to be called weblogs (at least by most people). Weblogs are kind of an interesting layout microcosm, when you think about it. Each entry in the weblog usually contains a title, the date the entry was posted, some content, and then some extra information such as the entry's category, a link to any comments, and so on.

From a layout point of view, each entry needs to be considered as if it were a mini-document within the larger page. Every entry should be styled the same as other entries while still relating visually to each other in an appealing way. It wouldn't be good if the entries overlapped each other, for example. In this project, we'll take a look at a weblog based on clean, structural markup and explore ways to style the entries.

 PREV

[< Day Day Up >](#)

NEXT 

 PREV

[< Day Day Up >](#)

NEXT 

Project Goals

This project is all about taking the entries in a weblog and making them look good. While we're at it, we need to make sure the entries relate to each other visually. This leads us to the following general goals:

- Visually speaking, make each entry seem distinct and self-contained. That doesn't mean drawing a box around each one—we just have to be sure it's obvious where one entry ends and another begins.
- Try to make the entry title and date come together in some sense so that they appear to be closely related.
- Take any extra information and reduce its visual emphasis so that it doesn't compete with the main entry text, but don't make the information completely disappear either.

Furthermore, we're going to use a natural theme for the design, all green and woodsy. That doesn't come from the preceding points; it's just a design decision we're making from the outset.

 PREV

[< Day Day Up >](#)

NEXT 

 PREV

[< Day](#) [Day Up >](#)

 NEXT

Preparation

Download the files for [Project 8](#) from this book's Web site. If you're planning to play along at home, load the file ch08proj.html into the editing program of your choice. This is the file you'll be editing, saving, and reloading as the project progresses.



See the [Introduction](#) for instructions on how to download files from the Web site.

 PREV

[< Day](#) [Day Up >](#)

 NEXT

 PREV

[< Day Day Up >](#)

NEXT 

Laying the Groundwork

As usual, we'll need to dive into the markup of the project document—the better to understand what we have to work with—and get a peek at the document before we start adding CSS to it. [Listing 8.1](#) gives us a detailed look at the markup for the weblog, and [Figure 8.1](#) shows it in its raw, unstyled glory.

Listing 8.1. A Look at the Weblog's Markup

 PREV

[< Day Day Up >](#)

NEXT 

 PREV

[< Day](#) [Day Up >](#)

NEXT 

Styling the Weblog

A good first step would be to define baselines for the background color, text color, and font family and sizing for the overall document. We're going to be aiming for a natural kind of feel, so we'll go with some green shades for the colors. The font will be, as is so often the case for personal sites, a smaller-than-default sans-serif font.

 PREV

[< Day](#) [Day Up >](#)

NEXT 

 PREV

[< Day](#) [Day Up](#) [>](#)

NEXT 

Branching Out

Here are some stylistic variations to try:

1.

Change the extra information so that the category sits on its own line before the other two bits of information. Also try adding a top border to the comment and pingback items and making sure that the pingback border does not touch the border to its right.

Project 8 - Mozilla

Eric's Thoughts

Hot Steaming Internet Tuesday, 23 November 2003

If you're on the east side of Cleveland and want a nice warm caffeinated place to get online, the new Arabica on Lee Road, just a block or so south of Cedar-Lee, is the place to be. The network: SSID is 2wxz173; it is a closed network but they'll tell you the password at the counter. Note to Mac users: you'll need to enter the password as a 40-bit hex key, not as a plain password. Something about their security setup causes this, although neither I nor they knew exactly what that might be. I figure it's no big deal, since once you enter the information and add it to your Keychain, you'll never have to worry about it again (unless of course they change it).

Morimoto Wednesday, 26 November 2003

Wow.

Kat and I have just returned from [Morimoto](#), where we had one of the most amazing meals of our entire lives. Although we'd been seated at a table to start, Kat decided (and rightly so) that we should move to the sushi bar. So with a little help from the hostess, we moved to sit at the end of the bar, just a few feet from Morimoto himself, and after a bit of debate we decided to start out with the seared kobe beef and green tea soba noodles. These were by themselves

WiFi | 0 comments | 3 Pingbacks

2.

Get rid of the margins on the entry titles and instead use padding and background positioning to place the title text and background image as they were in the project. This will also necessitate the removal of another border.

Project 8 - Mozilla

Eric's Thoughts

Hot Steaming Internet Tuesday, 23 November 2003

If you're on the east side of Cleveland and want a nice warm caffeinated place to get online, the new Arabica on Lee Road, just a block or so south of Cedar-Lee, is the place to be. The network: SSID is 2wxz173; it is a closed network but they'll tell you the password at the counter. Note to Mac users: you'll need to enter the password as a 40-bit hex key, not as a plain password. Something about their security setup causes this, although neither I nor they knew exactly what that might be. I figure it's no big deal, since once you enter the information and add it to your Keychain, you'll never have to worry about it again (unless of course they change it).

Morimoto Wednesday, 26 November 2003

Wow.

Kat and I have just returned from [Morimoto](#), where we had one of the most amazing meals of our entire lives. Although we'd been seated at a table to start, Kat decided (and rightly so) that we should move to the sushi bar. So with a little help from the hostess, we moved to sit at the end of the bar, just a few feet from Morimoto himself, and after a bit of debate we decided to start out with the seared kobe beef and green tea soba noodles. These were by themselves

WiFi | 0 comments | 3 Pingbacks

 PREV

[< Day](#) [Day Up >](#)

NEXT 

 PREV

[< Day Day Up >](#)

 NEXT

Project 9. Designing a Home Page

Good design keeps the user happy, the manufacturer in the black, and the aesthete unoffended.

—RAYMOND LOEWY

Styling a weblog, as in the preceding project, can be a challenge, but weblog style in itself isn't enough. Any weblog entries are likely to appear in the context of a site design, and the two need to go together. After all, it would look odd to have a leafy, natural theme for the weblog and a bunch of neon, J-pop decorations for the rest of the site's design. It would be even stranger to have both on the same page.

Since we already have a weblog design, let's actually work outward and create a site design that is consistent with the weblog styles, instead of going the other way around. This will let us build on the work we've already done while providing an example of how the various components of a page can fit together.

 PREV

[< Day Day Up >](#)

 NEXT

 PREV

[< Day Day Up >](#)

 NEXT

Project Goals

The nature (no pun intended) of this project doesn't lend itself to a lot of specific goals. In fact, the sum total of what we need to do can be expressed in three points:

- Take the weblog we styled in [Project 8](#) and place it into a full page. This means styling the masthead, sidebar, and footer to be consistent with the visual theme we established when styling the weblog.
- Place the navigation and presentation links to the right of the main content and make this "sidebar" touch the masthead.
- Highlight the name of the current theme.

These are fairly open guidelines, it's true, but they'll be enough to guide us to a finished design.

 PREV

[< Day Day Up >](#)

 NEXT

 PREV

< Day Day Up >

NEXT 

Preparation

Download the files for [Project 9](#) from this book's Web site. If you're planning to play along at home, load the file ch09proj.html into the editing program of your choice. This is the file you'll be editing, saving, and reloading as the project progresses.



See the [Introduction](#) for instructions on how to download files from the Web site.

 PREV

< Day Day Up >

NEXT 

 PREV

[< Day Day Up >](#)

NEXT 

Laying the Groundwork

Since we're going to be using the weblog styles, we already have a simple embedded style sheet, which is shown in [Listing 9.1](#) as a part of the overall document structure.

Listing 9.1. The Basic Document Structure and Starting Style Sheet

 PREV

[< Day Day Up >](#)

NEXT 

 PREV

[< Day](#) [Day Up](#) [>](#)

NEXT 

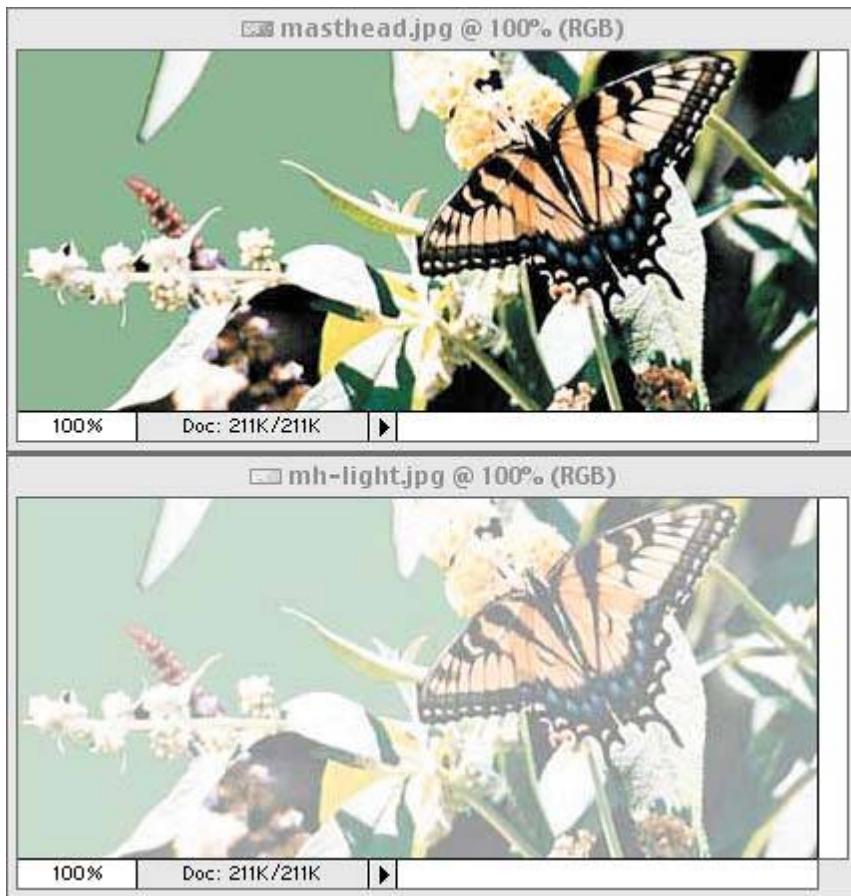
Creating the Design

As is usually the case, we'll start at the top of the document and work our way down. After getting the masthead put together, we'll move on to the main content and sidebar. This will let us effectively divide the layout into two zones: "masthead" and "the rest of the page." While it's true that the main content and sidebar will be placed in relation to each other, they'll both be placed in relation to the masthead, so in many ways it is the most important aspect of the design.

Two Images Behind the Masthead

Since the masthead is a key portion of the design, it stands to reason that it should look pretty. It's hard to create beautiful Web design without images, and in this case, we have two images to use in the masthead. They're shown in [Figure 9.3](#).

Figure 9.3. The images available for use in the masthead.



All right, so it's really two variants of the same image, masthead.jpg and mh-light.jpg, the latter of which is a washed-out version of the former. We'll be able to use them together to create a translucency effect, as we did in [Project 4](#), "Positioning in the Background."

First let's get the h1 element styled. It will be getting masthead.jpg for its background, so we need to match its background color to the background of the image. We're also going to be placing the image to the right edge of the element.

 PREV

[< Day](#) [Day Up >](#)

NEXT 

 PREV

[< Day](#) [Day Up](#) [>](#)

NEXT 

Branching Out

1.

Convert the design to use floats for the two columns, as discussed in the aside "Positioning Rather Than Floating." Remember that the styles given in the aside will not be enough—you'll still need to deal with things like margins, padding, and the placement of the footer.



2.

Reverse the layout so that the sidebar is on the left and the main content on the right. This will require changing the borders on the sidebar, and some text alignment should probably change as well. Also try altering the masthead to fit visually with the change in sidebar placement.



3.

 PREV

[< Day](#) [Day Up](#) [>](#)

NEXT 

 PREV

[< Day Day Up >](#)

 NEXT

Project 10. Designing in the Garden

I love you twisted

And I love you straight

I'd write it down but I can't concentrate

Words won't me obey they do as they please

And all I am left with is these

—ELVIS COSTELLO

In May 2003, the CSS world was introduced to an amazing new resource: the CSS Zen Garden. The goal of this site was to provide designers with an HTML document that they could not change in any way and to challenge them to write a style sheet that would present the file with flair, visual appeal, and originality.

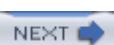
As of this writing, there were close to 100 different official Zen Garden designs, every one striking and every one different, some radically so. The site's goal had been to show that CSS design could be beautiful, that such beauty could be created on top of a relatively simple document structure, and that the same document could be presented in totally different ways. It has succeeded brilliantly on all counts.

Since Eric Meyer on CSS finished with a challenging project (in that case, re-creating the book's layout in CSS), it seemed to me that taking on a similar challenge would be fitting for this book. Creating a Zen Garden design seemed to fit the bill nicely.

There was only one problem. I freely admit that I'm not a strong visual artist (and some would label that a gross understatement), and Zen Garden designs need to look really good. So I decided to go to someone who is a strong visual artist: Dave Shea, founder of the CSS Zen Garden and a technical reviewer for this very book. Dave produced a beautiful design, and I turned it into a CSS-driven layout. This project is a recounting of the steps I took to make that happen.

 PREV

[< Day Day Up >](#)

 NEXT

 PREV

[< Day Day Up >](#)

NEXT 

Project Goals

We really only have one project goal this time around: to take a visual design and convert it to a CSS-driven layout. This simple goal contains a trio of more specific goals, however, especially since we're creating this design for the CSS Zen Garden.

- Create the layout without changing a single character of the content and markup in the HTML document provided. Only the CSS is under our control.
- Make sure the layout looks good in IE5.5+/Win, IE5/Mac, Safari, and the latest Gecko-based browsers (as of this writing, Mozilla 1.6 and Firefox 0.8).
- Have the design tolerate changes in font size up to 150% of the size of the user's default text size.

What we'll be doing is combining two very common situations in the life of a Web designer. The first is an attempt to reproduce a layout that a visual artist has created; this often happens in large companies (or any other organization) where the look of a site is under the direction of artists, not programmers. The second situation is one in which you have unchangeable markup that you need to style. In this case, the markup is specifically set up to make CSS layout easier, which is a good thing.

 PREV

[< Day Day Up >](#)

NEXT 

 PREV

[< Day Day Up >](#)

NEXT 

Preparation

Download the files for [Project 10](#) from this book's Web site. If you're planning to play along at home, load the file ch10proj.html into the editing program of your choice. This is the file you'll be editing, saving, and reloading as the project progresses.



See the [Introduction](#) for instructions on how to download files from the Web site.

 PREV

[< Day Day Up >](#)

NEXT 

 PREV

[< Day Day Up >](#)

NEXT 

Laying the Groundwork

As always, we need to understand the document skeleton before we try to clothe it in CSS. The basic document skeleton is given in [Listing 10.1](#).

Listing 10.1. The Document Skeleton

 PREV

[< Day Day Up >](#)

NEXT 

 PREV

[< Day](#) [Day Up >](#)

NEXT 

Creating the Design

To start out, we'll set some "global" styles: rules that will apply throughout the document. This includes stripping off the page margins (or padding), setting basic background and foreground colors, and removing the borders and underlines from acronym and link elements, respectively.

 PREV

[< Day](#) [Day Up >](#)

NEXT 

 PREV

[< Day](#) [Day Up >](#)

NEXT 

Adding a PNG

You may recall that fairly early in the project, we added the image of the big flower to the design. That image file is a GIF89a with a transparent area (skip back to [Figure 10.5](#) if you don't remember clearly how it looks). At the time, there was a brief mention that a PNG image would be a much prettier solution, and that we'd talk about it at the end of the project.

And PNG Means ?



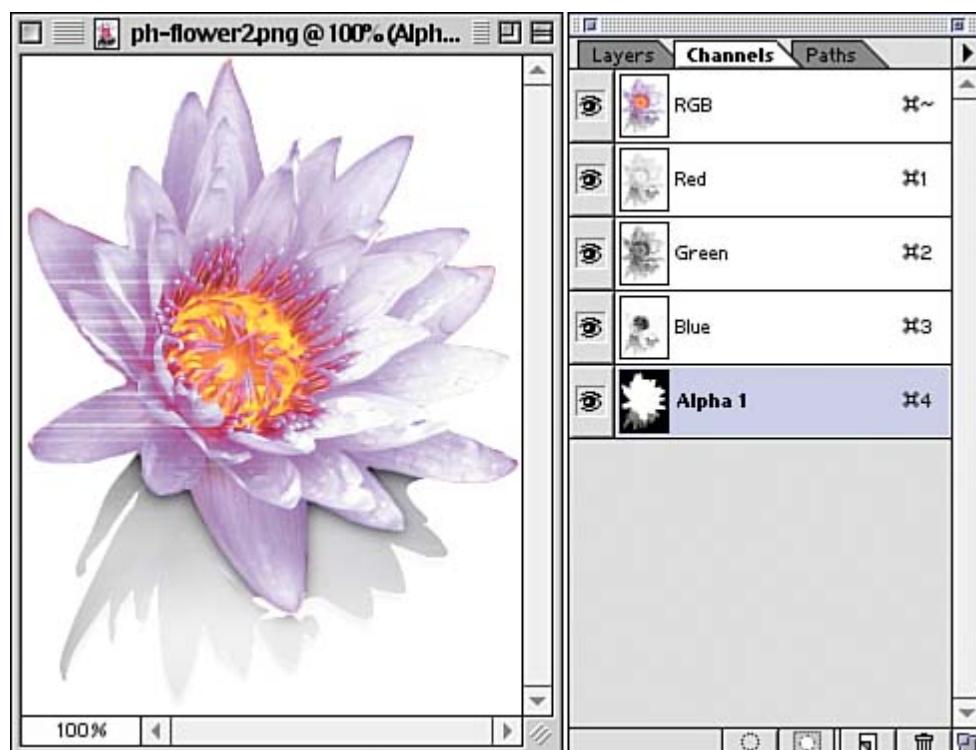
PNG stands for Portable Network Graphics, and is usually pronounced "ping." It's an image format that was devised in the early 1990s and became a W3C Recommendation in 1996 (<http://www.w3.org/TR/REC-png-multi.html>). Its intent was to be a patent-free and technically superior replacement for the GIF format.

Well, here we are at the end of the project, so let's talk about using a PNG instead of a GIF.

From a visual design perspective, the primary advantage of PNG is that a PNG file can contain gamma-correction information (so you don't have to worry about your images becoming lighter or darker in different operating systems) and it can include up to a full 16-bit alpha channel. This allows for translucency and transparency that's far more sophisticated than the simple on/off transparency of GIF files. In a GIF, a pixel is either opaque or transparent; there's no in-between. In a PNG image, every pixel can be semi-opaque to whatever degree is desired. So, creating a PNG that fades smoothly from black at the top to totally transparent at the bottom of the image is a snap.

Let's take a look at [Figure 10.23](#), which shows the PNG we'll be using as well as the channels in the file.

Figure 10.23. The PNG image and the RGBA channels it contains.



 PREV

[< Day](#) [Day Up](#) [>](#)

NEXT 

[PREV](#)[< Day Day Up >](#)[NEXT](#)

Reflections

In many ways, this was one of the most difficult projects I've created for the book. Why? Because I wasn't able to touch the markup and because I was working toward a very specific visual goal. The combination of these two constraints made the process very, very interesting.

The perfect example is the sidebar positioning. Sure, it's cool that we can use top and margin-top in combination to place the link list, but it turned out to be a fragile solution. Suppose the user bumps the text size up to 120% of normal. The quick summary text immediately flows to four lines, but there's no way for the top value to change as a result (short of using JavaScript, but that's outside the scope of the book). To keep everything aligned, it would need to change to 10.6em. Instead, the sidebar content ends up overlapping the quick summary area.

As I say, this is the case because the document markup was off limits. If it had been changeable, I would have reworked things so that all of the main column content was in one div so that I could just give it a single border. I also would have moved the link list into that div and then used it as the containing block for the link list. That way, the user could change text size all he wanted, and the sidebar would still line up with the content column. There would be no more danger of overlap.

This illuminates an issue that is very important but often overlooked: Presentation is dependent on structure. You may have heard the phrase "complete separation of structure and presentation." That's impossible with current technology and may always be impossible, although I'm no prophet and can't guess what might be possible in 5, 10, or 30 years. I can say that, as of now, a document with no structure—that is to say, no elements, just an undifferentiated sea of text—cannot be styled in any meaningful way. Without paragraphs and headings and divs and anchor elements to mark your hyperlinks, there's no hope of making things look good.

Similarly, if the structure of a document doesn't relate very well to the visual result you want (as was the case in this project), you end up either getting very creative or else having to change something. Usually, designers will just change the structure to better meet their layout needs. That's okay; in fact, it's often a good idea. The other possibility is to change the visual layout from what you wanted to what the document's structure can support. That's fine too, although it's generally not as satisfying.

So always remember that your design will depend on the document's structure. Sometimes that means grafting in an occasional div or span for presentational purposes. As long as you're doing that only when necessary, don't worry about it. If you find yourself frequently nesting spans inside (or around) links, though, you might want to rethink how you're doing whatever you're doing. It's important to keep things as simple and structurally appropriate as possible while still meeting your design needs.

One last note: My respect for the designers who have created Zen Garden layouts increased substantially by undertaking this project. Adapting a known design to the markup was a challenge; to create a completely new and original design on top of that markup bespeaks incredible skill and talent. To each and every one of the Zen Garden designers, I bow in reverence and humility. Thank you, one and all.

[PREV](#)[< Day Day Up >](#)[NEXT](#)

 PREV

[< Day](#) [Day Up >](#)

 NEXT

Branching Out

This time around, I really only have one suggestion for taking the concepts in this chapter further.

1.

Create your own Zen Garden design! It doesn't have to be the most beautiful design ever, nor does it have to be submitted, although you're certainly encouraged to do so if you like. All you need to do is create one or more style sheets that present the Zen Garden base file in a new and interesting way. Try different layouts with the markup, making them more and more complex as you learn. Remember: The markup is not to be touched. All you can do is write CSS. When you hit a limitation, see if you can find a creative way around it. Try things that don't seem like they have a chance in Hades of working. When you make a mistake, stop before you undo it. Was the result interesting in a way you didn't expect? If so, try following that path instead of the one you meant to follow. You never know what interesting technique or effect you might find just by working in the Zen Garden.

 PREV

[< Day](#) [Day Up >](#)

 NEXT

 PREV

[< Day](#) [Day Up >](#)

[[SYMBOL](#)] [[A](#)] [[B](#)] [[C](#)] [[D](#)] [[E](#)] [[F](#)] [[G](#)] [[H](#)] [[I](#)] [[J](#)] [[L](#)] [[M](#)] [[N](#)] [[O](#)] [[P](#)] [[R](#)] [[S](#)] [[T](#)] [[U](#)] [[V](#)] [[W](#)] [[Z](#)]

 PREV

[< Day](#) [Day Up >](#)

 PREV

[< Day](#) [Day Up >](#)

[[SYMBOL](#)] [[A](#)] [[B](#)] [[C](#)] [[D](#)] [[E](#)] [[F](#)] [[G](#)] [[H](#)] [[I](#)] [[J](#)] [[L](#)] [[M](#)] [[N](#)] [[O](#)] [[P](#)] [[R](#)] [[S](#)] [[T](#)] [[U](#)] [[V](#)] [[W](#)] [[Z](#)]

(octothorpe)

[permalinks](#)

[****changing branching out to suggestions for](#)

[***note to self all of book is about CSS****](#)

[\:nth-child\(\) pseudo-class](#)

 PREV

[< Day](#) [Day Up >](#)

 PREV

< Day Day Up >

[[SYMBOL](#)] [[A](#)] [[B](#)] [[C](#)] [[D](#)] [[E](#)] [[F](#)] [[G](#)] [[H](#)] [[I](#)] [[J](#)] [[L](#)] [[M](#)] [[N](#)] [[O](#)] [[P](#)] [[R](#)] [[S](#)] [[T](#)] [[U](#)] [[V](#)] [[W](#)] [[Z](#)]

[a elements](#)

[absolutely positioned elements](#)

accessing

structure

[for translucency effects](#) [2nd](#) [3rd](#)

adding

backgrounds

[translucency effects](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#)

graphics

[to home pages](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

images

[to page headers](#)

[to Zen Garden designs](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#)

PNG

[to Zen Garden designs](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#)

texture

[to links](#) [2nd](#) [3rd](#) [4th](#)

aligning

background images

blocks

text [2nd](#) [3rd](#)

aligning photos

catalog view [2nd](#) [3rd](#) [4th](#)

alignment

character alignment

financial reports baseline styles [2nd](#)

problems with wrapping text

rounding errors

alignments

alpha channels

alternate row markup

alternate rows

[highlighting](#) [2nd](#)

appearance of tabs

[refining](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#)

arbitrary hovering

arrow styles

[adding to sidebars](#) [2nd](#) [3rd](#) [4th](#)

attributes

[padding](#) [2nd](#) [3rd](#)

auto margins

[in catalog view](#) [2nd](#) [3rd](#) [4th](#)

auto-margins

avoiding

[horizontal scrollbars](#)

 PREV

< Day Day Up >

 PREV

[< Day](#) [Day Up >](#)

[[SYMBOL](#)] [[A](#)] [[B](#)] [[C](#)] [[D](#)] [[E](#)] [[F](#)] [[G](#)] [[H](#)] [[I](#)] [[J](#)] [[L](#)] [[M](#)] [[N](#)] [[O](#)] [[P](#)] [[R](#)] [[S](#)] [[T](#)] [[U](#)] [[V](#)] [[W](#)] [[Z](#)]

background colors

[printing](#)

[background images](#)

[aligning](#)

slide effects

[photo collections](#) [2nd](#) [3rd](#)

[summaries](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

[background warnings](#)

[background-position values](#)

backgrounds

adding

[translucency effects to](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#)

[adding images to page headers](#)

[making visible](#)

positioning

[for translucency effects.](#) [See [translucency effects](#)]

tabs

[solid background colors](#)

baseline styles

[financial reports](#) [2nd](#)

[weblogs](#) [2nd](#)

[behavior files](#) [2nd](#)

benefits

[of converting HTML to CSS](#) [2nd](#) [3rd](#) [4th](#)

blocking

links

[CSS-driven drop-down menus](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

blocks

[aligning](#)

[blogs.](#) [See [weblogs](#)]

body

[dropping images into body's background](#)

[borders](#) [2nd](#) [3rd](#)

and auto margins

[in catalog view](#) [2nd](#) [3rd](#) [4th](#)

around images

[deleting](#)

[changing](#) [2nd](#) [3rd](#)

colors

[deleting](#)

[div](#) [2nd](#)

divs

[drop-down menus](#) [2nd](#) [3rd](#) [4th](#)

home pages

[designing](#) [2nd](#)

[links](#)

[Opera](#)

shifting

[to links](#)

slide effects

[photo collections](#) [2nd](#) [3rd](#)

[Bowman, Douglas](#)

[box model](#) [2nd](#)

[boxed links](#) [2nd](#) [3rd](#)

branching out

 PREV

[< Day](#) [Day Up >](#)

 PREV

[< Day](#) [Day Up >](#)

[[SYMBOL](#)] [[A](#)] [[B](#)] [[C](#)] [[D](#)] [[E](#)] [[F](#)] [[G](#)] [[H](#)] [[I](#)] [[J](#)] [[L](#)] [[M](#)] [[N](#)] [[O](#)] [[P](#)] [[R](#)] [[S](#)] [[T](#)] [[U](#)] [[V](#)] [[W](#)] [[Z](#)]

[capitalization of text](#)

[catalog view \(photo collections\)](#) [2nd](#)

2nd [3rd](#) [4th](#)

borders and auto margins [2nd](#) [3rd](#) [4th](#)

floating photos [2nd](#) [3rd](#)

formatting text [2nd](#) [3rd](#)

[cellspacing attribute](#)

centering

[text](#)

centering photos

[contact sheet view](#) [2nd](#) [3rd](#) [4th](#)

changing

[borders](#) [2nd](#) [3rd](#)

channels

[alpha channels](#)

[character alignment](#)

[class attribute](#)

classes

[naming](#)

cleaning up

[translucency effects](#) [2nd](#) [3rd](#)

clear element

[margins](#)

[clipped fonts](#) [2nd](#)

collapsing

[margins](#)

[collections of photos.](#) [See [photo collections](#)]

color

background colors

[printing](#)

text

[changing](#)

[Color Blender](#)

colors

[borders](#)

[of links](#)

[sidebar links](#)

[text](#) [2nd](#)

columns

[HTML elements for](#)

[contact sheet view \(photo collections\)](#)

[floating photos](#) [2nd](#)

[slide effects](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

[spacing and centering photos](#) [2nd](#) [3rd](#) [4th](#)

content

[designing home pages](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#)

styling

[Zen Garden designs](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#) [8th](#) [9th](#) [10th](#) [11th](#)

[translucency effects](#) [2nd](#) [3rd](#) [4th](#)

converting

[HTML designs to CSS](#)

[goals for](#) [2nd](#)

[groundwork for](#) [2nd](#) [3rd](#)

[rebuilding the design.](#) [See CSS, rebuilding the desing]

[skeletal structure after removing HTML](#) [2nd](#) [3rd](#) [4th](#)

 PREV

[< Day](#) [Day Up >](#)

 PREV

< Day Day Up >

[[SYMBOL](#)] [[A](#)] [[B](#)] [[C](#)] [[D](#)] [[E](#)] [[F](#)] [[G](#)] [[H](#)] [[I](#)] [[J](#)] [[L](#)] [[M](#)] [[N](#)] [[O](#)] [[P](#)] [[R](#)] [[S](#)] [[T](#)] [[U](#)] [[V](#)] [[W](#)] [[Z](#)]

[default line-height values](#)

deficit styles

[financial reports](#) 2nd

deleting

[borders](#)

[borders around images](#)

[bullets](#) 2nd

[lists](#)

slide effects

[from gallery view](#) 2nd

designing

[home pages](#)

[adding graphics](#) 2nd 3rd 4th 5th

[borders](#) 2nd 3rd 4th 5th

[content and side bars](#) 2nd 3rd 4th 5th 6th 7th

[finishing touches](#) 2nd 3rd 4th 5th 6th

[natural highlighting](#) 2nd 3rd

[placing images behind the masthead](#) 2nd 3rd 4th 5th 6th

[project goals](#)

[shifting images](#)

[sidebar links](#) 2nd 3rd 4th

[styling sidebars](#) 2nd 3rd 4th 5th

designs

enhancing

[after converting to CSS](#) 2nd 3rd 4th

[fixed versus liquid](#)

[display](#)

div

[borders](#) 2nd

[padding](#)

[requirements div](#) 2nd 3rd 4th 5th 6th 7th 8th

[div#nav](#)

divs

[borders](#)

[drifting text](#)

dropping

images

[into body's background](#)

 PREV

< Day Day Up >

 PREV

< Day Day Up >

[[SYMBOL](#)] [[A](#)] [[B](#)] [[C](#)] [[D](#)] [[E](#)] [[F](#)] [[G](#)] [[H](#)] [[I](#)] [[J](#)] [[L](#)] [[M](#)] [[N](#)] [[O](#)] [[P](#)] [[R](#)] [[S](#)] [[T](#)] [[U](#)] [[V](#)] [[W](#)] [[Z](#)]

editing

[versus shifting images](#)

effects

[translucency effects.](#) [See [translucency effects](#)]

elements

[a](#)

[absolutely positioned elements](#)

[display](#)

[h4](#)

[ul](#)

[visibility](#)

[em-based margins](#)

ems

[sizing fonts](#)

enclosing

[links](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

[straddling the lines](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

enhancing

designs

[rebuilding after converting HTML to CSS](#) [2nd](#) [3rd](#) [4th](#)

entries

[hovering](#) [2nd](#) [3rd](#)

entry text styles

[weblogs](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

entry title styles

[weblogs](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

examining

markup

[for list-based menus](#) [2nd](#) [3rd](#) [4th](#)

Explorer for Windows

[auto margins](#)

[hiding styles from](#)

[lists in](#)

 PREV

< Day Day Up >

[SYMBOL] [A] [B] [C] [D] [E] [F] [G] [H] [I] [J] [L] [M] [N] [O] [P] [R] [S] [T] [U] [V] [W] [Z]

financial reports 2nd

preparation for project

print styles 2nd 3rd 4th

row highlighting 2nd

total styles 2nd

project goals 2nd

screen styles

baseline styles 2nd

deficit styles 2nd

layout tweaks 2nd 3rd

profit styles 2nd

table labels 2nd 3rd 4th

table markup 2nd

total styles

suggestions for adjustments 2nd 3rd

finishing touches

for home pages 2nd 3rd 4th 5th 6th

fixed designs

versus liquid designs

fixing

problems

after reorienting menus 2nd 3rd 4th

with submenus 2nd 3rd 4th 5th 6th

floating

links 2nd 3rd 4th 5th

versus positioning 2nd 3rd

floating images

weblogs 2nd

floating photos

catalog view 2nd 3rd

contact sheet view 2nd

font-size

em-based margins

fonts

changing

monospace fonts

sizing 2nd 3rd

footers

creating

Zen Garden designs 2nd 3rd 4th 5th 6th 7th 8th

formatting

text

in catalog view 2nd 3rd

 PREV

[< Day](#) [Day Up >](#)

[[SYMBOL](#)] [[A](#)] [[B](#)] [[C](#)] [[D](#)] [[E](#)] [[F](#)] [[G](#)] [[H](#)] [[I](#)] [[J](#)] [[L](#)] [[M](#)] [[N](#)] [[O](#)] [[P](#)] [[R](#)] [[S](#)] [[T](#)] [[U](#)] [[V](#)] [[W](#)] [[Z](#)]

[gallery view \(photo collections\)](#)

[deleting slide effects](#) [2nd](#)

[layout tweaks](#) [2nd](#)

[tables](#) [2nd](#)

[viewing titles](#) [2nd](#) [3rd](#)

[generated content](#)

global styles

[applying to tabs](#) [2nd](#) [3rd](#) [4th](#)

[setting](#) [2nd](#)

[goals.](#) [See [project goals](#)]

graphics

 adding

[to home pages](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

groundwork

[for Zen Garden designs](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

laying

[for CSS-driven drop-down menus](#) [2nd](#) [3rd](#) [4th](#)

[for home pages](#) [2nd](#) [3rd](#) [4th](#)

[laying for tabs](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

grouping

 PREV

[< Day](#) [Day Up >](#)

[[SYMBOL](#)] [[A](#)] [[B](#)] [[C](#)] [[D](#)] [[E](#)] [[F](#)] [[G](#)] [[H](#)] [[I](#)] [[J](#)] [[L](#)] [[M](#)] [[N](#)] [[O](#)] [[P](#)] [[R](#)] [[S](#)] [[T](#)] [[U](#)] [[V](#)] [[W](#)] [[Z](#)]

[h4 elements](#)

headers

page headers

[Zen Garden designs](#) [2nd](#) [3rd](#) [4th](#)

hiding

styles

[from Explorer for Windows](#)

[submenus](#)

highlighting

[alternate rows](#) [2nd](#)

[designing home pages](#) [2nd](#) [3rd](#)

hits

[range of](#)

home pages

[designing](#)

[adding graphics](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

[borders](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

[content and sidebars](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#)

[finishing touches](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

[natural highlighting](#) [2nd](#) [3rd](#)

[placing images behind the masthead](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

[shifting images](#)

[sidebar links](#) [2nd](#) [3rd](#) [4th](#)

[styling sidebars](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

[laying groundwork for](#) [2nd](#) [3rd](#) [4th](#)

[other options for creating](#) [2nd](#)

[project goals for](#)

horizontal scrollbars

[avoiding](#)

hotspots

[shrunken hotspots](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

hover styles

hovering

[arbitrary hovering](#)

entries

[CSS-driven drop-down menus](#) [2nd](#) [3rd](#)

[links](#) [2nd](#) [3rd](#) [4th](#)

[nonlink elements](#)

HTML designs

[converting to CSS](#)

[goals for](#) [2nd](#)

[laying groundwork](#) [2nd](#) [3rd](#)

[rebuilding the design.](#) [See CSS, rebuilding the design]

[skeletal structure after removing HTML](#) [2nd](#) [3rd](#) [4th](#)

[stripping out HTML-based presentation](#) [2nd](#) [3rd](#)

[HTML Tidy](#)

HTML-based designs

[converting to CSS](#) [2nd](#) [3rd](#) [4th](#)

 PREV

[< Day](#) [Day Up >](#)

[[SYMBOL](#)] [[A](#)] [[B](#)] [[C](#)] [[D](#)] [[E](#)] [[F](#)] [[G](#)] [[H](#)] [[I](#)] [[J](#)] [[L](#)] [[M](#)] [[N](#)] [[O](#)] [[P](#)] [[R](#)] [[S](#)] [[T](#)] [[U](#)] [[V](#)] [[W](#)] [[Z](#)]

IE/Win

[bugs in](#)
[jogs in text flow](#)
[margin calculations](#) 2nd
[negative margins](#)
[spacing](#)

image text

[versus real text](#) 2nd
[image-replacement techniques](#) 2nd 3rd
[versus images](#) 2nd

images

adding
[to page headers](#)
[to Zen Garden designs](#) 2nd 3rd 4th 5th 6th 7th
[adding to links](#) 2nd 3rd 4th

and entry title styles

[weblogs](#) 2nd 3rd

background images

[aligning](#)
[for summaries](#) 2nd 3rd 4th 5th 6th

borders around

[deleting](#)

dropping

[into body background](#)

floating images

[weblogs](#) 2nd

JPEG images

[translucency effects](#) 2nd 3rd 4th

placing

[Zen Garden designs](#)

[placing behind mastheads](#) 2nd 3rd 4th 5th 6th

PNG. [See [PNG](#)][2nd](#) [See [PNG](#)][3rd](#) [See [PNG](#)][4th](#) [See [PNG](#)][5th](#) [See [PNG](#)][6th](#) [See [PNG](#)][7th](#) [See [PNG](#)]

]

[shifting](#)

[shifting versus editing](#)

[sizing](#)

tabs

[size of](#)

[versus image-replacement techniques](#) 2nd

images. [See also [photo collections](#)]

[indenting](#)

indicating

[current tab](#) 2nd 3rd 4th 5th

inline boxes

[and markers](#)

[inline elements](#)

Internet Explorer

[bullets](#)

Internet Explorer 5

[differences in browsers](#)

Internet Explorer for Windows. [See [Explorer for Windows](#)]

italicizing

[sidebars](#)

[italicizing text](#) 2nd

 PREV

[< Day](#) [Day Up >](#)

 PREV

[< Day](#) [Day Up >](#)

[[SYMBOL](#)] [[A](#)] [[B](#)] [[C](#)] [[D](#)] [[E](#)] [[F](#)] [[G](#)] [[H](#)] [[I](#)] [[J](#)] [[L](#)] [[M](#)] [[N](#)] [[O](#)] [[P](#)] [[R](#)] [[S](#)] [[T](#)] [[U](#)] [[V](#)] [[W](#)] [[Z](#)]

jog

[IE/Win](#)

JPEG images

[translucency effects](#) [2nd](#) [3rd](#) [4th](#)

 PREV

[< Day](#) [Day Up >](#)

 PREV

[< Day](#) [Day Up >](#)

[[SYMBOL](#)] [[A](#)] [[B](#)] [[C](#)] [[D](#)] [[E](#)] [[F](#)] [[G](#)] [[H](#)] [[I](#)] [[J](#)] [[L](#)] [[M](#)] [[N](#)] [[O](#)] [[P](#)] [[R](#)] [[S](#)] [[T](#)] [[U](#)] [[V](#)] [[W](#)] [[Z](#)]

labels

[financial reports](#) [2nd](#) [3rd](#) [4th](#)

labels. [See also [titles](#)]

laying

groundwork

[for CSS-driven drop-down menus](#) [2nd](#) [3rd](#) [4th](#)

[for home pages](#) [2nd](#) [3rd](#) [4th](#)

[for Zen Garden designs](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

laying groundwork

[for tabs](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

letter-spacing attribute [2nd](#)

line-height

line-height values

[default line-height values](#)

links [2nd](#)

[adding texture to](#) [2nd](#) [3rd](#) [4th](#)

blocking

[CSS-driven drop-down menus](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

borders

boxed links [2nd](#) [3rd](#)

color shades

colors

[cut off text](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

enclosing

[in list-based menus](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

[straddling the lines](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

floating [2nd](#) [3rd](#) [4th](#) [5th](#)

hotspots [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

hover effects

hovering [2nd](#) [3rd](#)

padding

permalink

[weblogs](#) [2nd](#) [3rd](#)

[separating](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

shifting borders to

shrink-wrapping [2nd](#) [3rd](#)

sidebar links [2nd](#) [3rd](#) [4th](#)

skip-links

[removing](#) [2nd](#) [3rd](#)

styling

[in list-based menus](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#) [8th](#)

wrapping problems

liquid designs

[versus fixed designs](#)

list items

[adding hover effects](#)

list-based menus

arrow styles

[adding](#) [2nd](#) [3rd](#) [4th](#)

boxed links [2nd](#) [3rd](#)

[laying ground work for project](#) [2nd](#) [3rd](#)

links

[enclosing](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

[straddling the lines](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

markup

 PREV

[< Day](#) [Day Up >](#)

 PREV

< Day Day Up >

[[SYMBOL](#)] [[A](#)] [[B](#)] [[C](#)] [[D](#)] [[E](#)] [[F](#)] [[G](#)] [[H](#)] [[I](#)] [[J](#)] [[L](#)] [[M](#)] [[N](#)] [[O](#)] [[P](#)] [[R](#)] [[S](#)] [[T](#)] [[U](#)] [[V](#)] [[W](#)] [[Z](#)]

manipulating

position of backgrounds. [See [translucency effects](#)]

margins

auto margins

in catalog view [2nd](#) [3rd](#) [4th](#)

auto-margins

clear element

collapsing

em-based margins

IE/Win margin calculations [2nd](#)

negative margins

IE/Win

spacing and centering photos [2nd](#)

using when positioning

versus padding

Zen Garden designs

markers

and inline boxes

markup

for list-based menus

examining [2nd](#) [3rd](#) [4th](#)

for photo collections

weblogs [2nd](#)

whitespace in

mastheads

placing images behind [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

rebuilding after converting HTML to CSS [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#)

translucency effects [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#) [8th](#)

menus

CSS-driven drop-down menus. [See [CSS-driven drop-down menus](#)]

list-based menus. [See [list-based menus](#)]

reorienting [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

fixing problems with [2nd](#) [3rd](#) [4th](#)

submenu corrections [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

monospace fonts

 PREV

< Day Day Up >

 PREV

[< Day](#) [Day Up >](#)

[[SYMBOL](#)] [[A](#)] [[B](#)] [[C](#)] [[D](#)] [[E](#)] [[F](#)] [[G](#)] [[H](#)] [[I](#)] [[J](#)] [[L](#)] [[M](#)] [[N](#)] [[O](#)] [[P](#)] [[R](#)] [[S](#)] [[T](#)] [[U](#)] [[V](#)] [[W](#)] [[Z](#)]

naming

[classes](#)

[nav](#)

[natural highlighting](#) [2nd](#) [3rd](#)

nav

[naming](#)

navbars

[positioning](#)

[rebuilding after converting HTML to CSS](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

navigation

tabs

[styling](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

[Nederlof, Peter](#)

negative margins

[IE/Win](#)

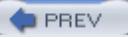
negative profit styles

[financial reports](#) [2nd](#)

[nesting](#)

nonlink elements

[hovering](#)

 PREV

[< Day](#) [Day Up >](#)

[PREV](#)

[< Day](#) [Day Up >](#)

[[SYMBOL](#)] [[A](#)] [[B](#)] [[C](#)] [[D](#)] [[E](#)] [[F](#)] [[G](#)] [[H](#)] [[I](#)] [[J](#)] [[L](#)] [[M](#)] [[N](#)] [[O](#)] [[P](#)] [[R](#)] [[S](#)] [[T](#)] [[U](#)] [[V](#)] [[W](#)] [[Z](#)]

octothorpe (#)

[permalinks](#)

[online styles.](#) [See [screen styles](#)]

Opera

[borders](#)

[overflowing text](#)

overlapping text

[avoiding](#)

[PREV](#)

[< Day](#) [Day Up >](#)

 PREV

[< Day](#) [Day Up >](#)

[SYMBOL] [A] [B] [C] [D] [E] [F] [G] [H] [I] [J] [L] [M] [N] [O] [P] [R] [S] [T] [U] [V] [W] [Z]

padding 2nd 3rd

links

pixel-based padding

sidebars

uneven padding

versus margins

page headers

adding images to

Zen Garden designs 2nd 3rd 4th

percentages

sizing fonts

permalink

weblogs 2nd 3rd

photo collections

catalog view 2nd

aligning photos 2nd 3rd 4th

borders and auto margins 2nd 3rd 4th

floating photos 2nd 3rd

formatting text 2nd 3rd

contact sheet view

floating photos 2nd

slide effects 2nd 3rd 4th 5th

spacing and centering photos 2nd 3rd 4th

gallery view

deleting slide effects 2nd

layout tweaks 2nd

tables 2nd

viewing titles 2nd 3rd

markup

preparation for project

project goals 2nd

style guide for 2nd

suggestions for adjustments 2nd 3rd 4th

picture collections. [See photo collections]

pixel-based padding

placing

images

behind mastheads 2nd 3rd 4th 5th 6th

Zen Garden designs

planning

for CSS-driven drop-down menus 2nd 3rd 4th

PNG

adding

to Zen Garden designs 2nd 3rd 4th 5th 6th 7th

browsers

PNG (Portable Network Graphics) 2nd 3rd 4th 5th 6th 7th

pngquant

pop-ups

preparing for 2nd 3rd 4th 5th

Portable Network Graphics. [See PNG]2nd [See PNG]3rd [See PNG]4th [See PNG]5th [See PNG]6th [See PNG]7th [See PNG]

positioning

backgrounds

translucency effects. [See translucency effects]

images

 PREV

[< Day](#) [Day Up >](#)

 PREV

[< Day](#) [Day Up >](#)

[[SYMBOL](#)] [[A](#)] [[B](#)] [[C](#)] [[D](#)] [[E](#)] [[F](#)] [[G](#)] [[H](#)] [[I](#)] [[J](#)] [[L](#)] [[M](#)] [[N](#)] [[O](#)] [[P](#)] [[R](#)] [[S](#)] [[T](#)] [[U](#)] [[V](#)] [[W](#)] [[Z](#)]

raised effects

[creating](#)

[range of hits](#)

real text

[versus image text](#) [2nd](#)

rebuilding

[designs that were converted from HTML](#). [See CSS, rebuilding the design]

reducing

[text size](#)

removing

[skip-links](#) [2nd](#) [3rd](#)

[removing](#). [See [deleting](#)]

reorienting

[menus](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

[fixing problems with](#) [2nd](#) [3rd](#) [4th](#)

[submenu corrections](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

[requirements div](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#) [8th](#)

reviews

[rebuilding after converting HTML to CSS](#) [2nd](#) [3rd](#) [4th](#)

rounding errors

[in alignment](#)

row-separator lines

[financial reports](#)

rows

[alternate row markup](#)

[highlighting alternate rows](#) [2nd](#)

[rules](#)

 PREV

[< Day](#) [Day Up >](#)

 PREV

[< Day](#) [Day Up >](#)

[[SYMBOL](#)] [[A](#)] [[B](#)] [[C](#)] [[D](#)] [[E](#)] [[F](#)] [[G](#)] [[H](#)] [[I](#)] [[J](#)] [[L](#)] [[M](#)] [[N](#)] [[O](#)] [[P](#)] [[R](#)] [[S](#)] [[T](#)] [[U](#)] [[V](#)] [[W](#)] [[Z](#)]

screen styles

[financial reports](#)

[baseline styles](#) [2nd](#)

[deficit styles](#) [2nd](#)

[layout tweaks](#) [2nd](#) [3rd](#)

[profit styles](#) [2nd](#)

[table labels](#) [2nd](#) [3rd](#) [4th](#)

[table markup](#) [2nd](#)

[total styles](#)

scrollbars

horizontal scrollbars

[avoiding](#)

separating

[links](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

shifting

borders

[to links](#)

[images](#)

[versus editing images](#)

showing

submenus

[in response to user actions](#)

showing. [See [viewing](#)]

shrink-wrapping

[links](#) [2nd](#) [3rd](#)

[sidebar links](#) [2nd](#) [3rd](#) [4th](#)

sidebars

[adding arrow styles to](#) [2nd](#) [3rd](#) [4th](#)

[background images](#)

[designing home pages](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#)

[italicizing](#)

[padding](#)

[rebuilding after converting HTML to CSS](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

[styling](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

[Zen Garden designs](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#) [8th](#) [9th](#) [10th](#) [11th](#) [12th](#) [13th](#)

sizing

[fonts](#) [2nd](#) [3rd](#)

[images](#)

tabs

[automatically](#)

[text](#)

[skip-links](#)

[removing](#) [2nd](#) [3rd](#)

slide effects

[contact sheet view](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

[deleting from gallery view](#) [2nd](#)

spacing

[IE/Win](#)

[text characters](#) [2nd](#)

spacing photos

[contact sheet view](#) [2nd](#) [3rd](#) [4th](#)

[specificity](#) [2nd](#)

stopping

[text from wrapping](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#)

straddling

 PREV

[< Day](#) [Day Up >](#)

 PREV

[< Day](#) [Day Up >](#)

[[SYMBOL](#)] [[A](#)] [[B](#)] [[C](#)] [[D](#)] [[E](#)] [[F](#)] [[G](#)] [[H](#)] [[I](#)] [[J](#)] [[L](#)] [[M](#)] [[N](#)] [[O](#)] [[P](#)] [[R](#)] [[S](#)] [[T](#)] [[U](#)] [[V](#)] [[W](#)] [[Z](#)]

tables

[alternate row markup](#)

alternate rows

[highlighting](#) [2nd](#)

[cellspacing attribute](#)

financial reports

[labels](#) [2nd](#) [3rd](#) [4th](#)

[screen styles](#) [2nd](#)

[gallery view \(photo collections\)](#) [2nd](#)

[in photo collections](#)

[troubleshooting](#)

tabs

[adding texture](#) [2nd](#) [3rd](#) [4th](#)

[autosizing](#)

[creating](#)

[style sheets, changing](#) [2nd](#) [3rd](#)

[cut off text](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

[hovering links](#) [2nd](#) [3rd](#)

images

[size of](#)

[indicating current tab](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

[laying ground work for](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

[optional ideas for using](#) [2nd](#)

[project goals](#) [2nd](#)

[putting in](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

raised effects

[creating](#)

[refining look of](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#)

[shrunken hotspots](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

[solid background colors](#)

[styling](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

text

[aligning](#) [2nd](#) [3rd](#)

[avoiding overlapping with photos](#)

[capitalization](#)

[centering](#)

color

[changing](#)

[colors](#) [2nd](#)

[cut off text](#)

[links](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

[drifting to the left](#)

[entry text styles](#)

[weblogs](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

fonts

[changing](#)

[monospace fonts](#)

formatting

[in catalog view](#) [2nd](#) [3rd](#)

[handling](#)

[italicizing](#) [2nd](#)

[overflowing text](#)

[real text versus image text](#) [2nd](#)

[reducing size of](#)

[sizing](#)

 PREV

[< Day](#) [Day Up >](#)

 PREV

[< Day](#) [Day Up >](#)

[[SYMBOL](#)] [[A](#)] [[B](#)] [[C](#)] [[D](#)] [[E](#)] [[F](#)] [[G](#)] [[H](#)] [[I](#)] [[J](#)] [[L](#)] [[M](#)] [[N](#)] [[O](#)] [[P](#)] [[R](#)] [[S](#)] [[T](#)] [[U](#)] [[V](#)] [[W](#)] [[Z](#)]

[ul elements](#)

utilities

[pngquant](#)

 PREV

[< Day](#) [Day Up >](#)

 PREV

[< Day](#) [Day Up >](#)

[[SYMBOL](#)] [[A](#)] [[B](#)] [[C](#)] [[D](#)] [[E](#)] [[F](#)] [[G](#)] [[H](#)] [[I](#)] [[J](#)] [[L](#)] [[M](#)] [[N](#)] [[O](#)] [[P](#)] [[R](#)] [[S](#)] [[T](#)] [[U](#)] [[V](#)] [[W](#)] [[Z](#)]

vertical-align attribute

viewing

titles

[in gallery view](#) [2nd](#) [3rd](#)

visibility

[making backgrounds visible](#)

 PREV

[< Day](#) [Day Up >](#)

 PREV

[< Day](#) [Day Up >](#)

[[SYMBOL](#)] [[A](#)] [[B](#)] [[C](#)] [[D](#)] [[E](#)] [[F](#)] [[G](#)] [[H](#)] [[I](#)] [[J](#)] [[L](#)] [[M](#)] [[N](#)] [[O](#)] [[P](#)] [[R](#)] [[S](#)] [[T](#)] [[U](#)] [[V](#)] [[W](#)] [[Z](#)]

warnings

[backgrounds](#)

[Web browser styles](#). [See [screen styles](#)]

webblogs

[suggestions for adjustments](#) [2nd](#) [3rd](#)

[weblogs](#) [2nd](#)

[baseline styles](#) [2nd](#)

[entry text styles](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

[floating images](#) [2nd](#)

[markup](#) [2nd](#)

[permalinks](#) [2nd](#) [3rd](#)

[preparation for project](#)

[project goals](#) [2nd](#)

[style sheet \(listing 8.2\)](#) [2nd](#)

[title style](#) [2nd](#) [3rd](#) [4th](#) [5th](#)

[entry titles](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

whitespace

[in markup](#)

wrapping

[text](#)

[alignment problems](#)

[stopping process of](#) [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#)

[wrapping text](#) [2nd](#) [3rd](#)

 PREV

[< Day](#) [Day Up >](#)

 PREV

[< Day](#) [Day Up >](#)

[[SYMBOL](#)] [[A](#)] [[B](#)] [[C](#)] [[D](#)] [[E](#)] [[F](#)] [[G](#)] [[H](#)] [[I](#)] [[J](#)] [[L](#)] [[M](#)] [[N](#)] [[O](#)] [[P](#)] [[R](#)] [[S](#)] [[T](#)] [[U](#)] [[V](#)] [[W](#)] [[Z](#)]

z-index

Zen Garden designs [2nd](#)

adding PNG [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#)

creating [2nd](#) [3rd](#)

adding images [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#)

footers [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#) [8th](#)

page headers [2nd](#) [3rd](#) [4th](#)

styling content [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#) [8th](#) [9th](#) [10th](#) [11th](#)

styling sidebars [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#) [7th](#) [8th](#) [9th](#) [10th](#) [11th](#) [12th](#) [13th](#)

styling summaries [2nd](#) [3rd](#) [4th](#) [5th](#) [6th](#)

laying groundwork for [2nd](#) [3rd](#) [4th](#) [5th](#)

project goals [2nd](#)

suggestions for

 PREV

[< Day](#) [Day Up >](#)