



Adobe®
Dreamweaver® CC

CLASSROOM IN A BOOK®

The official training workbook from Adobe Systems



Adobe® Dreamweaver®CC

CLASSROOM IN A BOOK®

The official training workbook from Adobe Systems

Adobe® Dreamweaver® CC Classroom in a Book®

© 2014 Adobe Systems Incorporated and its licensors. All rights reserved.

If this guide is distributed with software that includes an end user license agreement, this guide, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by any such license, no part of this guide may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Adobe Systems Incorporated. Please note that the content in this guide is protected under copyright law even if it is not distributed with software that includes an end user license agreement.

The content of this guide is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.

Please remember that existing artwork or images that you may want to include in your project may be protected under copyright law. The unauthorized incorporation of such material into your new work could be a violation of the rights of the copyright owner. Please be sure to obtain any permission required from the copyright owner.

Any references to company names in sample files are for demonstration purposes only and are not intended to refer to any actual organization.

Adobe, the Adobe logo, Dreamweaver, and Classroom in a Book are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Adobe Systems Incorporated, 345 Park Avenue, San Jose, California 95110-2704, USA

Notice to U.S. Government End Users. The Software and Documentation are “Commercial Items,” as that term is defined at 48 C.F.R. §2.101, consisting of “Commercial Computer Software” and “Commercial Computer Software Documentation,” as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.7202, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §§227.7202-1 through 227.7202-4, as applicable, the Commercial Computer Software and Commercial Computer Software Documentation are being licensed to U.S. Government end users (a) only as Commercial Items and (b) with only those rights as are granted to all other end users pursuant to the terms and conditions herein. Unpublished-rights reserved under the copyright laws of the United States. Adobe Systems Incorporated, 345 Park Avenue, San Jose, CA 95110-2704, USA. For U.S. Government End Users, Adobe agrees to comply with all applicable equal opportunity laws including, if appropriate, the provisions of Executive Order 11246, as amended, Section 402 of the Vietnam Era Veterans Readjustment Assistance Act of 1974 (38 USC 4212), and Section 503 of the Rehabilitation Act of 1973, as amended, and the regulations at 41 CFR Parts 60-1 through 60-60, 60-250, and 60-741. The affirmative action clause and regulations contained in the preceding sentence shall be incorporated by reference.

Adobe Press books are published by Peachpit, a division of Pearson Education located in San Francisco, California. For the latest on Adobe Press books, go to www.adobepress.com. To report errors, please send a note to errata@peachpit.com. For information on getting permission for reprints and excerpts, contact permissions@peachpit.com.

Writer: James J. Maivald
Project Editor: Nancy Peterson
Development Editor: Robyn G. Thomas
Copyeditor and Proofreader: Darren Meiss
Production Coordinator and Compositor: Danielle Foster
Technical Reviewers: Candyce Mairs, Clint Funk
Indexer: Jack Lewis
Cover Designer: Eddie Yuen
Interior Designer: Mimi Heft

Printed and bound in the United States of America

ISBN-13: 978-0-321-91941-0

ISBN-10: 0-321-91941-6

9 8 7 6 5 4 3 2 1

CONTENTS




GETTING STARTED 1





| | |
|---|---|
| About Classroom in a Book..... | 1 |
| TinyURLs | 1 |
| Prerequisites | 2 |
| Installing the program | 2 |
| Downloading the Classroom in a Book files | 2 |
| Recommended lesson order | 3 |
| Jumpstart | 3 |
| Setting up the workspace | 4 |
| Windows vs. OS X instructions..... | 5 |
| Finding Dreamweaver information | 5 |
| Checking for updates | 5 |
| Additional resources | 6 |
| Adobe certification | 7 |





1 CUSTOMIZING YOUR WORKSPACE 8





| | |
|--|----|
| Touring the workspace..... | 10 |
| Switching and splitting views | 11 |
| Working with panels..... | 15 |
| Selecting a workspace layout..... | 18 |
| Adjusting toolbars | 20 |
| Personalizing preferences..... | 21 |
| Creating custom keyboard shortcuts | 21 |
| Using the Property inspector..... | 23 |
| Using the CSS Designer | 24 |
| Explore, experiment, learn | 25 |

| | | |
|---|---|-----------|
| 2 | HTML BASICS | 28 |
|  | What is HTML? | 30 |
| | Where did HTML begin? | 30 |
| | Writing your own HTML code | 32 |
| | Frequently used HTML 4 codes | 40 |
| | Introducing HTML5 | 42 |
| 3 | CSS BASICS | 48 |
|  | What is CSS? | 50 |
| | HTML vs. CSS formatting | 51 |
| | HTML defaults | 53 |
| | CSS box model | 56 |
| | Multiples, classes, and IDs, oh my! | 57 |
| | Formatting text | 59 |
| | Formatting objects | 77 |
| | CSS3 overview and support | 92 |
| 4 | CREATING A PAGE LAYOUT | 96 |
|  | Web design basics | 98 |
| | Working with thumbnails and wireframes | 101 |
| | Defining a Dreamweaver site | 105 |
| | Using the Welcome screen | 106 |
| | Previewing your completed file | 107 |
| | Modifying an existing CSS layout | 108 |
| | Adding a background image to the header | 113 |
| | Inserting new components | 117 |
| | Inserting a navigation component | 120 |
| | Changing element alignment | 122 |
| | Modifying the page width and background color | 123 |
| | Modifying existing content and formatting | 126 |
| | Inserting an image placeholder | 130 |
| | Inserting placeholder text | 131 |
| | Modifying the footer | 132 |
| | Validating webpages | 133 |

| | | |
|---|---|------------|
| 5 | WORKING WITH CASCADING STYLE SHEETS | 136 |
|  | Previewing the completed file..... | 138 |
| | Working with the CSS Designer | 139 |
| | Working with type | 143 |
| | Working with background graphics | 156 |
| | Working with classes, IDs, and descendant selectors | 158 |
| | Creating an interactive menu | 165 |
| | Creating faux columns | 175 |
| | Moving rules to an external style sheet | 177 |
| | Creating style sheets for other media | 179 |
| 6 | WORKING WITH TEMPLATES | 186 |
|  | Previewing completed files | 188 |
| | Creating a template from an existing layout | 189 |
| | Inserting editable regions..... | 190 |
| | Producing child pages | 192 |
| | Updating a template..... | 195 |
| | Using library items..... | 197 |
| | Using server-side includes | 203 |
| 7 | WORKING WITH TEXT, LISTS, AND TABLES | 212 |
|  | Previewing the completed file..... | 214 |
| | Creating and styling text..... | 216 |
| | Creating and styling tables..... | 229 |
| | Spell checking webpages | 243 |
| | Finding and replacing text | 244 |
| 8 | WORKING WITH IMAGES | 250 |
|  | Reviewing web image basics..... | 252 |
| | Previewing the completed file..... | 257 |
| | Inserting an image..... | 258 |
| | Adjusting image positions with CSS classes | 260 |
| | Working with the Insert panel..... | 261 |
| | Using Adobe Bridge to insert images | 262 |
| | Inserting non-web file types | 265 |
| | Working with Photoshop Smart Objects..... | 268 |
| | Copying and pasting images from | |

| | | |
|---|---|------------|
| | Photoshop and Fireworks | 270 |
| | Inserting images by drag and drop | 274 |
| | Optimizing images with the Property inspector | 274 |
| 9 | WORKING WITH NAVIGATION | 280 |
|  | Hyperlink basics | 282 |
| | Previewing the completed file | 283 |
| | Creating internal hyperlinks | 286 |
| | Creating an image-based link | 289 |
| | Creating an external link | 290 |
| | Setting up email links | 292 |
| | Targeting page elements | 293 |
| | Checking your page | 298 |
| 10 | ADDING INTERACTIVITY | 300 |
|  | Learning about Dreamweaver behaviors | 302 |
| | Previewing the completed file | 304 |
| | Working with Dreamweaver behaviors | 305 |
| | Working with jQuery Accordion widgets | 312 |
| 11 | WORKING WITH WEB ANIMATION AND VIDEO | 322 |
|  | Understanding web animation and video | 324 |
| | Previewing the completed file | 325 |
| | Adding web animation to a page | 326 |
| | Adding web video to a page | 330 |
| 12 | WORKING WITH FORMS | 338 |
|  | Previewing the completed file | 340 |
| | Learning about forms | 341 |
| | Adding a form to a page | 343 |
| | Inserting text form elements | 345 |
| | Inserting checkboxes | 351 |
| | Creating radio buttons | 353 |
| | Incorporating text areas | 355 |
| | Working with lists | 357 |
| | Adding a Submit button | 360 |
| | Specifying a form action | 362 |
| | Emailing form data | 363 |

| | | |
|---|--|-------------|
| | Styling forms | 367 |
| | More form information..... | 368 |
| 13 | PUBLISHING TO THE WEB | 370 |
|  | Defining a remote site | 372 |
| | Cloaking folders and files | 378 |
| | Wrapping things up | 380 |
| | Putting your site online | 384 |
| | Synchronizing local and remote sites | 386 |
| 14 | DESIGNING FOR MOBILE DEVICES | 390 |
|  | Previewing your completed file | 392 |
| | Designing for mobile devices | 393 |
| | Creating a media query for smart phones..... | 397 |
| | Adding rules to a media query | 400 |
| | Adapting CSS navigation to mobile environments | 404 |
| | Styling the main content..... | 406 |
| | Testing a media query | 407 |
| | Responsive design..... | 409 |
| | Creating style sheets for tablets | 416 |
| | Edge Inspect | 418 |
| | More information..... | 419 |
| | APPENDIX: TINYURLS | 422 |
| | INDEX | 424 |
| 15 | BONUS LESSON: WORKING WITH CODE | BL-2 |
| | Code tools overview | BL-4 |
| | Selecting code..... | BL-5 |
| | Collapsing code | BL-8 |
| | Expanding code | BL-9 |
| | Adding new code..... | BL-9 |
| | Identifying CSS styling | BL-11 |
| | Working with Live Code..... | BL-14 |
| | Using Inspect mode | BL-15 |
| | Working in related files..... | BL-16 |
| | Accessing Split Code view..... | BL-17 |
| | Commenting your code..... | BL-17 |

GETTING STARTED

Adobe® Dreamweaver® CC is the industry-leading web authoring program. Whether you create websites for a living or plan to create one for your own business, Dreamweaver offers all the tools you need to get professional-quality results.

About Classroom in a Book

Adobe Dreamweaver CC Classroom in a Book® is part of the official training series for graphics and publishing software developed with the support of Adobe product experts.

The lessons are designed so that you can learn at your own pace. If you're new to Dreamweaver, you'll learn the fundamentals of putting the program to work. If you are an experienced user, you'll find that *Classroom in a Book* teaches many advanced features, including tips and techniques for using the latest version of Dreamweaver.

Although each lesson includes step-by-step instructions for creating a specific project, you'll have room for exploration and experimentation. You can follow the book from start to finish, or complete only those lessons that correspond to your interests and needs. Each lesson concludes with a review section containing questions and answers on the subjects you've covered.

TinyURLs

In several points in the book, we reference external information available on the Internet. The uniform resource locators (URLs) for this information are often long and unwieldy, so we have provided custom TinyURLs in their place for your convenience. Unfortunately, the TinyURLs sometimes expire over time and no longer function. If you find that a TinyURL doesn't work, look up the actual URL provided in the Appendix.

Prerequisites

Before using *Adobe Dreamweaver CC Classroom in a Book*, you should have a working knowledge of your computer and its operating system. Be sure you know how to use the mouse, standard menus, and commands, and also how to open, save, and close files. If you need to review these techniques, see the printed or online documentation included with your Windows or Mac operating system.

Installing the program

Before you perform any exercises in this book, verify that your computer system meets the hardware requirements for Dreamweaver CC, that it's correctly configured, and that all required software is installed.

Adobe Dreamweaver CC software must be purchased separately; it is not included with the lesson files that accompany this book. For system requirements, go to www.adobe.com/products/dreamweaver/tech-specs.html.

Downloading the Classroom in a Book files

In order to work through the projects in this book, you will need to download the lesson files from peachpit.com. You can download the files for individual lessons, or download them all at one time.

To access the Classroom in a Book files:

● **Note:** Do not copy one lesson folder into any other lesson folder. The files and folders for each lesson cannot be used interchangeably.

- 1 On a Mac or PC, go to www.peachpit.com/redeem and enter the code found at the back of your book.
- 2 If you do not have a Peachpit.com account, you will be prompted to create one.
- 3 The downloadable files will be listed under the Lesson & Update Files tab on your Account page.
- 4 Click the lesson file links to download them to your computer.
- 5 Rename the Lessons folder as **DW-CC**.
This folder will be used as the local site root folder.
- 6 If you want to complete one or more lessons individually, proceed to the “Jumpstart” section for more instructions.

Recommended lesson order

The training in this book is designed to take you from A to Z in basic to intermediate website design, development, and production. Each new lesson builds on previous exercises, using the files and assets to create an entire website. We recommend that you download all lesson files at once and then perform each lesson in sequential order to achieve a successful result and the most complete understanding of all aspects of web design.

The ideal training scenario will start in Lesson 1 and proceed through the entire book to Lesson 15 (Lesson 15 is online and can be found on your Account page at www.peachpit.com). Since each lesson builds essential files and content for the next, once you start this scenario you shouldn't skip any lessons, or even individual exercises. While ideal, this method may not be a practicable scenario for every user. So, if desired, individual lessons can be accomplished using the jumpstart method described in the next section.

Jumpstart

For users who don't have the time or inclination to perform each lesson in the book in sequence, or who are having difficulty with a particular lesson, a jumpstart method is included to facilitate the performance of individual lessons in or out of sequence. Once you start using the jumpstart method, you will have to use this method for all subsequent lessons. For example, if you want to jumpstart Lesson 6, you will have to jumpstart Lesson 7, too. In many instances, essential files needed for subsequent exercises were built in earlier lessons and exercises and may not be present in a jumpstart environment.

Each lesson folder includes all the files and assets needed to complete the exercises contained within that lesson. Each folder contains finished files, staged files, and customized Template and Library files, but not always a complete set of files that may have been used or completed in other lessons. You may think these folders contain seemingly duplicative materials. But these duplicate files and assets, in most cases, cannot be used interchangeably in other lessons and exercises. Doing so will probably cause you to fail to achieve the goal of the exercise.



The jumpstart method for completing individual lessons treats each folder as a stand-alone website. To jumpstart a lesson, copy the lesson folder to your hard drive and create a new site for that lesson using the Site Setup dialog box. Do not define sites using subfolders of existing sites. Keep your jumpstart sites and assets in their original folders to avoid conflicts. One suggestion is to organize the lesson folders, as well as your own site folders, in a single *web* or *sites* master folder near the root of your hard drive. But avoid using the Dreamweaver application folder or any folders that contain a web server, like Apache, ColdFusion, or Internet Information Services (IIS).

Feel free to use the jumpstart method for all lessons, if you prefer.

To set up a jumpstart site, do the following:

- 1 Choose Site > New Site.

The Site Setup dialog box appears.

- 2 In the Site Name field, enter the name of the lesson, such as **lesson06**.
- 3 Click the Browse () icon next to the Local Site Folder field. Navigate to the desired lesson folder among the files you downloaded from your Account page on Peachpit.com and click Select/Choose.
- 4 Click the arrow () next to the Advanced Settings category to reveal the tabs listed there. Select the Local Info category.
- 5 Click the Browse icon next to the Default Images Folder field. When the dialog box opens, navigate to the Images folder contained within the lesson folder and click Select/Choose.
- 6 In the Site Setup dialog box, click Save.
- 7 The name of the currently active website will appear in the Files panel's site pop-up menu. If necessary, press F8 to display the Files panel, and select the desired website you wish to work on from the Show menu.

These steps will have to be repeated for each lesson you wish to jumpstart. For a more complete description of how to set up a site in Dreamweaver, see Lesson 4, “Creating a Page Layout.” Remember, if you use the jumpstart method for all lessons, you may not end up with a complete set of site files in any individual folder when you are finished.

Setting up the workspace

Dreamweaver includes two main workspaces to accommodate various computer configurations and individual workflows. For this book the Expanded workspace is recommended.

- 1 In Dreamweaver CC, locate the Application bar. It appears along the top of the program.
- 2 If the Expanded workspace is not displayed by default, you can select it from the Workspace pop-up menu on the right side of the screen.
- 3 If the default Expanded workspace has been modified—where certain toolbars and panels are not visible (as they appear in the figures in the book)—you can restore the factory setting by selecting Reset ‘Expanded’ from the Workspace pop-up menu.

These Workspace Layout options can also be accessed from the Windows menu.

Most of the figures in this book show the Expanded workspace. When you finish the lessons in this book, experiment with both workspaces to find the one that you prefer, or build your own configuration and save the layout under a custom name.

For a more complete description of the Dreamweaver workspaces, see Lesson 1, “Customizing Your Workspace.”

Windows vs. OS X instructions

In most cases, Dreamweaver performs identically in both Windows and OS X. Minor differences exist between the two versions, mostly due to platform-specific issues out of the control of the program. Most of these are simply differences in keyboard shortcuts, how dialog boxes are displayed, and how buttons are named. Screen shots may alternate between platforms throughout the book. Where specific commands differ, they are noted within the text. Windows commands are listed first, followed by the OS X equivalent, such as Ctrl-C/Cmd-C. Common abbreviations are used for all commands whenever possible, as follows:

| WINDOWS | OS X |
|-----------------|---------------|
| Control = Ctrl | Command = Cmd |
| Alternate = Alt | Option = Opt |

Finding Dreamweaver information

For complete, up-to-date information about Dreamweaver panels, tools, and other application features, choose Help > Dreamweaver Help. Help files are cached locally so you can access them even when you are not connected to the Internet. You can also download a PDF version of the Dreamweaver Help files from the Adobe Help application.

For additional information resources, such as tips, techniques, and the latest product information, visit <http://helpx.adobe.com/dreamweaver.html>.

Checking for updates

Adobe periodically provides software updates. To check for updates in the program, go to the Help menu and select Updates. For book updates and bonus material, visit your Account page on Peachpit.com.

Additional resources

Adobe Dreamweaver CC Classroom in a Book is not meant to replace documentation that comes with the program or to be a comprehensive reference for every feature. Only the commands and options used in the lessons are explained in this book. For comprehensive information about program features and tutorials, please refer to these resources:

Adobe Dreamweaver CC Help and Support: <http://helpx.adobe.com/dreamweaver.html> is where you can find and browse Help and Support content on adobe.com.

Adobe Creative Cloud Learning: For inspiration, key techniques, cross-product workflows, and updates on new features go to the Creative Cloud Learn page <https://helpx.adobe.com/creative-cloud/tutorials.html>. Available only to Creative Cloud subscribers.

Adobe Forums: forums.adobe.com lets you tap into peer-to-peer discussions, questions, and answers on Adobe products.

Adobe TV: tv.adobe.com is an online video resource for expert instruction and inspiration about Adobe products, including a How To channel to get you started with your product.

Adobe Design Center: www.adobe.com/designcenter offers thoughtful articles on design and design issues, a gallery showcasing the work of top-notch designers, tutorials, and more.

Adobe Developer Connection: www.adobe.com/devnet is your source for technical articles, code samples, and how-to videos that cover Adobe developer products and technologies.

Resources for educators: www.adobe.com/education and <http://edex.adobe.com> offer a treasure trove of information for instructors who teach classes on Adobe software. Find solutions for education at all levels, including free curricula that use an integrated approach to teaching Adobe software and can be used to prepare for the Adobe Certified Associate exams.

Also check out these useful links:

Adobe Marketplace & Exchange: www.adobe.com/cfusion/exchange is a central resource for finding tools, services, extensions, code samples, and more to supplement and extend your Adobe products.

Adobe Dreamweaver CC product home page: www.adobe.com/products/dreamweaver

Adobe Labs: labs.adobe.com gives you access to early builds of cutting-edge technology, as well as forums where you can interact both with the Adobe development teams building that technology and with other like-minded members of the community.

Adobe certification

The Adobe training and certification programs are designed to help Adobe customers improve and promote their product-proficiency skills. There are four levels of certification:

- Adobe Certified Associate (ACA)
- Adobe Certified Expert (ACE)
- Adobe Certified Instructor (ACI)
- Adobe Authorized Training Center (AATC)

The Adobe Certified Associate (ACA) credential certifies that individuals have the entry-level skills to plan, design, build, and maintain effective communications using different forms of digital media.

The Adobe Certified Expert program is a way for expert users to upgrade their credentials. You can use Adobe certification as a catalyst for getting a raise, finding a job, or promoting your expertise.

If you are an ACE-level instructor, the Adobe Certified Instructor program takes your skills to the next level and gives you access to a wide range of Adobe resources.

Adobe Authorized Training Centers offer instructor-led courses and training on Adobe products, employing only Adobe Certified Instructors. A directory of AATCs is available at partners.adobe.com.

For information on the Adobe Certified programs, visit www.adobe.com/support/certification/index.html.

4

CREATING A PAGE LAYOUT

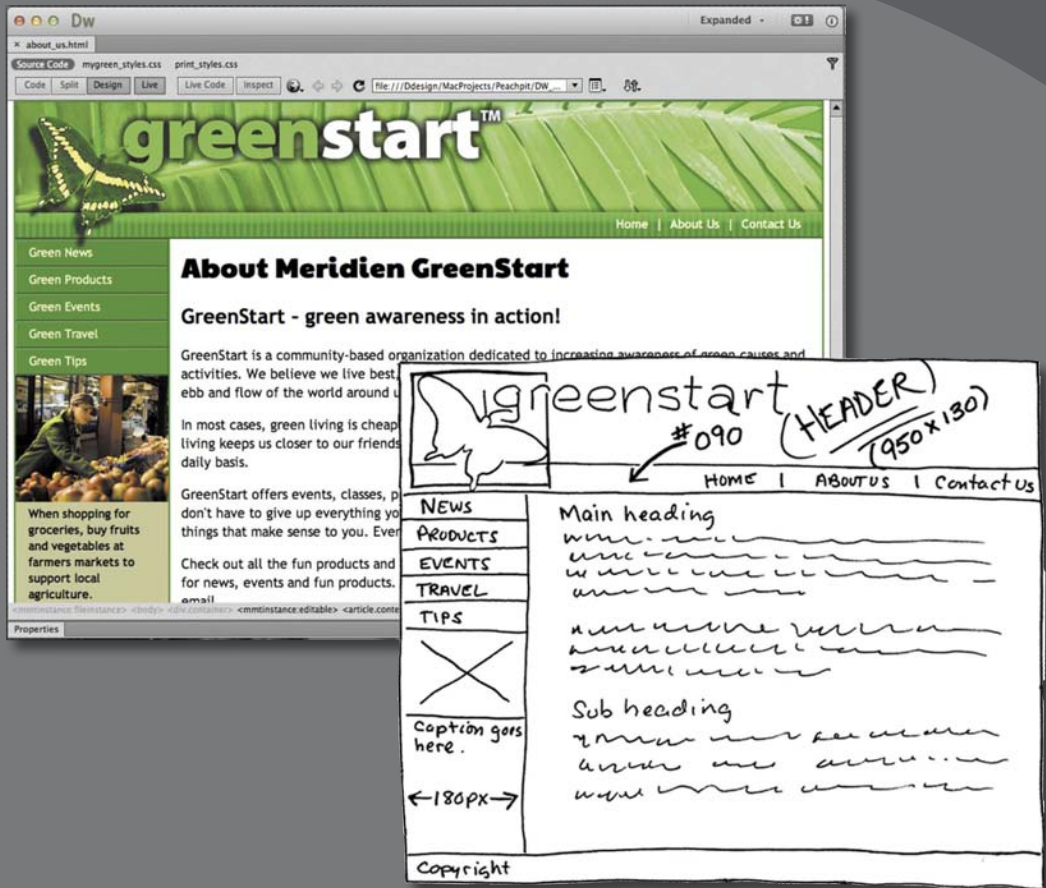
Lesson Overview

In this lesson, you'll learn the following:

- The basics of webpage design
- How to create design thumbnails and wireframes
- How to insert and format new components into a predefined CSS layout
- How to use the CSS Designer to identify applied CSS formatting
- How to check for browser compatibility



This lesson will take about 1 hour and 30 minutes to complete. If you have not already done so, download the project files for this lesson from the Lesson & Update Files tab on your Account page at www.peachpit.com, and store them on your computer in a convenient location, as described in the Getting Started section of this book. Your Accounts page is also where you'll find any updates to the lessons or to the lesson files. Look on the Lesson & Update Files tab to access the most current content. If you are starting from scratch in this lesson, use the method described in the "Jumpstart" section of "Getting Started."



Whether you use thumbnails and wireframes or just a vivid imagination, Dreamweaver can quickly turn design concepts into complete, standards-based CSS layouts.

● **Note:** If you have not already downloaded the project files for this lesson to your computer from your Account page, make sure to do so now. See “Getting Started” at the beginning of the book.

Web design basics

Before you begin any web design project for yourself or for a client, you need to answer three important questions:

- What is the purpose of the website?
- Who is the audience?
- How do they get here?

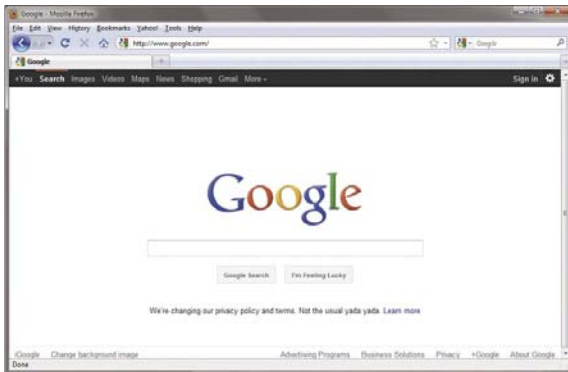
What is the purpose of the website?

Will the website sell or support a product or service? Is your site for entertainment or games? Will you provide information or news? Will you need a shopping cart or database? Do you need to accept credit card payments or electronic transfers? Knowing the purpose of the website tells you what type of content you’ll be developing and working with and what types of technologies you’ll need to incorporate.

Who is the customer?

Is the audience adults, children, seniors, professionals, hobbyists, men, women, everyone? Knowing *who* your market will be is vital to the overall design and functionality of your site. A site intended for children probably needs more animation, interactivity, and bright engaging colors. Adults will want serious content and in-depth analysis. Seniors may need larger type and other accessibility enhancements.

A good first step is to check out the competition. Is there an existing website performing the same service or selling the same product? Are they successful? You don’t have to mimic others just because they’re doing the same thing. Look at Google and Yahoo. They perform the same basic service, but their site designs couldn’t be more different from one another.



How do they get here?

This sounds like an odd question when speaking of the Internet. But, just as with a brick-and-mortar business, your online customers can come to you in a variety of ways. For example, are they accessing your site on a desktop computer, laptop, tablet, or cell phone? Are they using high-speed Internet, wireless, or dial-up service? What browser do they most like to use, and what is the size and resolution of the display? These answers will tell you a lot about what kind of experience your customers will expect. Dial-up and cell phone users may not want to see a lot of graphics or video, whereas users with large flat-panel displays and high-speed connections may demand as much bang and sizzle as you can send at them.

So, where do you get this information? Some you'll have to get through painstaking research and demographic analysis. Some you'll get from educated guesses based on your own tastes and understanding of your market. But a lot of it is actually available on the Internet itself. W3Schools, for one, keeps track of tons of statistics regarding access and usage, all updated regularly:

- **w3schools.com/browsers/browsers_stats.asp**: Provides more information about browser statistics.
- **w3schools.com/browsers/browsers_os.asp**: Gives the breakdown on operating systems. In 2011, they started to track the usage of mobile devices on the Internet.
- **w3schools.com/browsers/browsers_display.asp**: Lets you find out the latest information on the resolutions, or size, of screens using the Internet.

If you are redesigning an existing site, your web hosting service itself may provide valuable statistics on historical traffic patterns and even the visitors themselves.

If you host your own site, you can incorporate third-party tools, such as Google Analytics and Adobe Omniture, into your code to do the tracking for you for free or for a small fee.

As of the beginning of 2013, Windows (80 to 90 percent) still dominates the Internet, with most users favoring Google Chrome (48 percent), followed by Firefox (30 percent), and then various versions of Internet Explorer (13 percent) a distant third. The vast majority of browsers (90 percent) are set to a resolution *higher* than 1024 pixels by 768 pixels. If it weren't for the rapid growth in usage of tablets and smartphones for accessing the Internet, these statistics would be great news for most web designers and developers. But, designing a website that can look good and work effectively for both flat-panel displays and cell phones is a tall order.

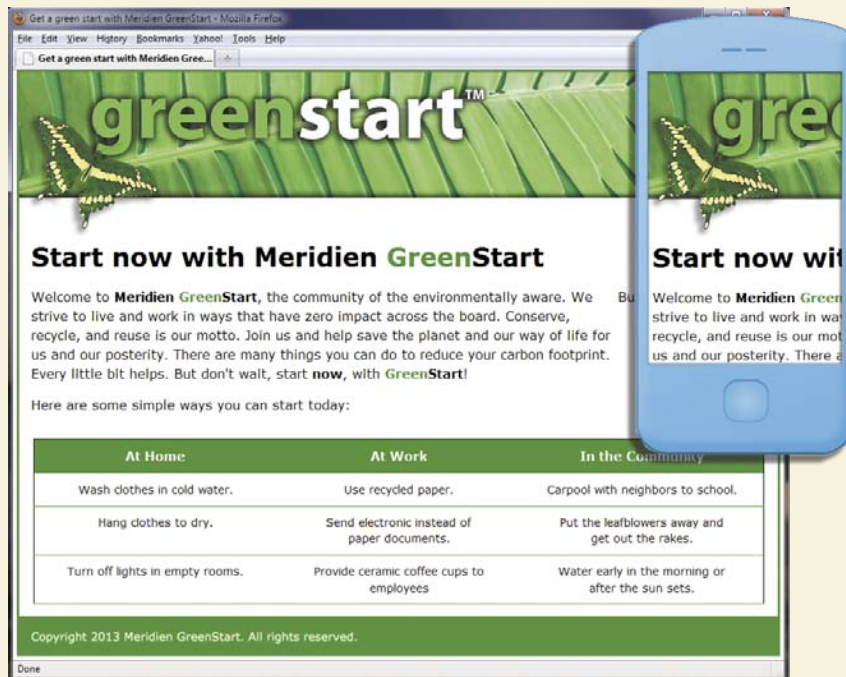
Responsive web design

Each day, more people are using cell phones and other mobile devices to access the Internet. Some people may use them to access the Internet more frequently than they use desktop computers. This presents a nagging problem to web designers. For one thing, cell phone screens are a fraction of the size of even the smallest flat-panel display. How do you cram a two- or three-column page design into a meager 200 to 300 pixels? Another problem is that most device manufacturers have decided to follow Apple's decision to drop support for Flash-based content on their mobile devices.

Until recently, web design usually required that you target an optimum size (height and width in pixels) for a webpage and then build the entire site on these specifications. Today, that scenario is becoming a rare occurrence. Now, you are presented with the decision to either build a site that can adapt to displays of multiple different dimensions or build two or more separate websites to support desktop *and* mobile users at the same time.

Your own decision will be based in part on the content you want to provide and on the capabilities of the devices accessing your pages. Building an attractive website that supports video, audio, and other dynamic content is hard enough without throwing in a panoply of different display sizes and device capabilities. The term *responsive web design* was coined by a Boston-based web developer name Ethan Mercotte in a book by the same name (2011), which describes the notion of designing pages that can adapt to multiple screen dimensions automatically. As you work through the following lessons, you will learn many techniques for responsive web design. By the time you get to Lesson 14, "Designing for Mobile Devices," you'll be fully prepared to tackle this weighty subject.

Many of the concepts of print design are not applicable to the web, because you are not in control of the user's experience. A page carefully designed for a typical flat panel is basically useless on a cell phone.



Scenario

For the purposes of this book you will be working to develop a website for Meridien GreenStart, a fictitious community-based organization dedicated to green investment and action. This website will offer a variety of products and services and require a broad range of webpage types, including dynamic pages using server-based technologies like PHP.

Your customers come from a broad demographic including all ages and education levels. They are people who are concerned about environmental conditions and who are dedicated to conservation, recycling, and the reuse of natural and human resources.

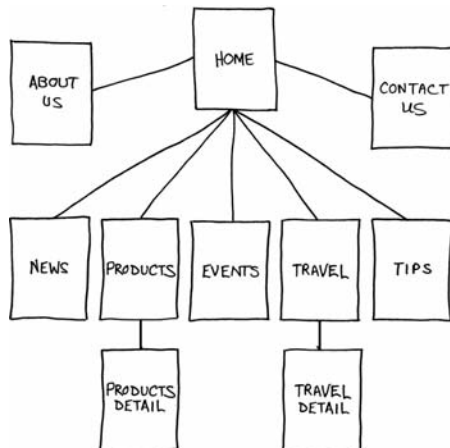
Your marketing research indicates that most of your customers use desktop computers or laptops, connecting via high-speed Internet services, but that you can expect 10 to 20 percent of your visitors via cell phone and other mobile devices. To simplify the process of learning Dreamweaver, we will focus on creating a fixed-width site design. In Lesson 14, “Designing for Mobile Devices,” you will learn how to adapt your fixed-width design to work with smartphones and tablet devices.

Working with thumbnails and wireframes

The next step, after you have nailed down the answers to the three questions about your website purpose, customer demographic, and access model, is to determine how many pages you’ll need, what those pages will do, and finally, what they will look like.

Creating thumbnails

Many web designers start by drawing thumbnails with pencil and paper. Think of thumbnails as a graphical shopping list of the pages you’ll need to create for the website. Thumbnails can also help you work out the basic website navigation structure. Draw lines between the thumbnails showing how your navigation will connect them.



Thumbnails list the pages that need to be built and how they are connected to each other.

Most sites are divided into levels. Typically, the first level includes all the pages in your main navigation menu, the ones a visitor can reach directly from the home page. The second level includes pages you can reach only through specific actions or from specific locations, say from a shopping cart or product detail page.

Creating a page design

Once you've figured out what your site needs in terms of pages, products, and services, you can then turn to what those pages will look like. Make a list of components you want on each page, such as headers and footers, navigation, and areas for the main content and the sidebars (if any). Put aside any items that won't be needed on every page. What other factors do you need to consider?

Identifying the essential components for each page helps in creating an effective page design and structure that will meet your needs.

1. Header (includes banner and logo)
2. Footer (copyright info)
3. Horizontal navigation (for internal reference, i.e., Home, About Us, Contact Us)
4. Vertical navigation (links to products and services)
5. Main content (one-column with chance of two or more)

Do you have a company logo, business identity, graphic imagery, or color scheme you want to accent? Do you have publications, brochures, or current advertising campaigns you want to emulate? It helps to gather them all in one place so you can see everything all at once on a desk or conference table. If you're lucky, a theme will rise organically from this collage.

Once you've created your checklist of the components that you'll need on each page, sketch out several rough layouts that work for these components. Most designers settle on one basic page design that is a compromise between flexibility and sizzle. Some site designs may naturally lean toward using more than one basic layout. But resist the urge to design each page separately. Minimizing the number of page designs may sound like a major limitation, but it's key to producing a professional-looking site. It's the reason why some professionals, like doctors and airline pilots, wear uniforms. Using a consistent page design, or template, conveys a sense of professionalism and confidence to your visitor.

Wireframes allow you to experiment with page designs quickly and easily without wasting time with code.



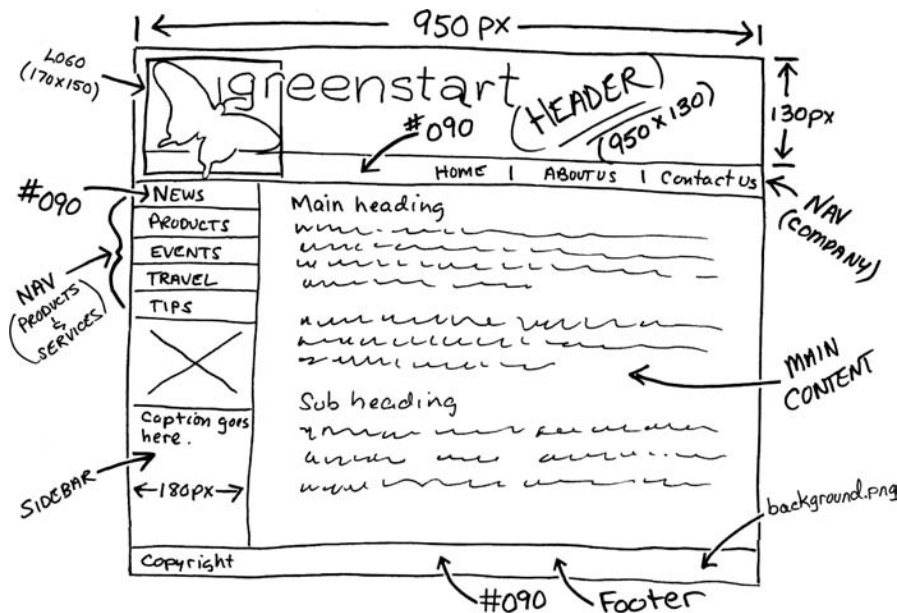
While you figure out what your pages will look like, you'll have to address the size and placement of the basic components. Where you put a component can drastically affect its impact and usefulness. In print, designers know that the upper-left corner of a layout is considered one of the “power positions,” a place where you want to locate important aspects of a design, such as a logo or title. This is because in western culture we read from left to right, top to bottom. The second power position is the lower-right corner because this is the last thing your eyes will see when you're finished reading.

Unfortunately, in web design this theory doesn't work so well because of one simple reason: You can never be certain how the user is seeing your design. Are they on a 20-inch flat panel or a 2-inch cell phone?

In most instances, the only thing you can be certain of is that the user can see the upper-left corner of any page. Do you want to waste this position by slapping the company logo here? Or, make the site more useful by slipping in a navigation menu? This is one of the key predicaments of the web designer. Do you go for design sizzle, workable utility, or something in between?

Creating wireframes

After you pick the winning design, wireframing is a fast way to work out the structure of each page in the site. A wireframe is like a thumbnail, but bigger, that sketches out each page and fills in more details about the components, such as actual link names and main headings. This step helps to catch or anticipate problems before you smack into them when working in the code.



The wireframe for the final design should identify the components and feature markup for content, color, and dimensions.

► **Tip:** For years, designers have started the design process in Fireworks, where they can create a fully functional mockup that can then be exported to a CSS-based HTML layout and then edited in Dreamweaver.

In some cases, creating a mockup in Photoshop, Fireworks, or Illustrator can save hours of tedious coding to receive a needed approval.

Once the basic concepts are worked out, many designers take an extra step and create a full-size mockup or “proof of concept” using a program like Adobe Fireworks, Photoshop, or even Illustrator. It’s a handy thing to do because you’ll find that some clients just aren’t comfortable giving an approval based only on pencil sketches. The advantage here is that all these programs allow you to export the results to full-size images (JPEG, GIF, or PNG) that can be viewed in a browser. Such mockups are as good as seeing the real thing but may take only a fraction of the time to produce.



Jumpstart method

The method we recommend for learning how to use Dreamweaver and to build the website pages and components described in the book is to work consecutively through each of the lessons until you have successfully completed all the exercises. For readers who can follow this model, you will use the site defined in the previous exercise for all lessons in the book.

For readers who cannot work through each lesson in order, or those who need to focus on a specific lesson topic, a Jumpstart method was developed to allow you to start a lesson out of sequence by defining the site on the folder provided for that lesson. Within that folder, components and partially completed files have been staged to permit this type of workflow. This method was described in detail in the “Getting Started” section of the book.

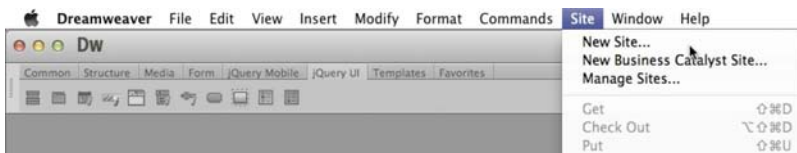
The Jumpstart requires you to define a site, using the steps defined earlier, targeting the desired lesson folder itself as the site root folder. In this case, you would target the Lesson04 folder provided in the online assets and then name the site appropriately, such as *Lesson04*.

Defining a Dreamweaver site

From this point forward, the lessons in this book will function within a Dreamweaver site. You will create webpages from scratch and use existing files and resources that are stored on your hard drive, which combined make up what's called your *local* site. When you are ready to upload your site to the Internet (see Lesson 13, "Publishing to the Web"), you publish your completed files to a web host server, which then becomes your *remote* site. The folder structures and files of the local and remote sites are usually mirrors of each other.

First, let's set up your local site:

- 1 Launch Adobe Dreamweaver CC if necessary.
- 2 Open the Site menu.



The Site menu provides options for creating and managing standard Dreamweaver sites, or to create a Business Catalyst site. Business Catalyst is an online, hosted application that allows you to build and manage rich, dynamic web-based businesses. To learn more about the capabilities of Business Catalyst, check out www.BusinessCatalyst.com.

- 3 Choose New Site.

To create a standard website in Dreamweaver CC, you need only name it and select the local site folder. Site names typically relate to a specific project or client and will appear in the Files panel. This name is intended for your own purposes, so there are no limitations to the name you can choose. Use a name that clearly describes the purpose of the website.

- 4 In the Site Name field, type **DW-CC**.
- 5 Next to the Local Site Folder field, click the folder (📁) icon. When the Choose Root Folder dialog box opens, navigate to the DW-CC folder containing the lesson files you downloaded from the *Adobe Dreamweaver CC Classroom in a Book* online resources.

You could click Save at this time and begin working on your new website, but we'll add one more piece of handy information.

- 6 Click the arrow (▶) next to the Advanced Settings category to reveal the categories listed there. Select Local Info.

Although it's not required, a good policy for site management is to store different file types in separate folders. For example, many websites provide

● **Note:** If you are completing this lesson separately from the rest of the lessons in book, see the detailed "Jumpstart" instructions in the "Getting Started" section at the beginning of the book. Then, follow the steps in this exercise.

● **Note:** If you have not already downloaded the project files for this lesson to your computer from your Account page, make sure to do so now. See "Getting Started" at the beginning of the book.

● **Note:** If you are following the Jumpstart method name the site Lesson04. For each subsequent lesson you will create a new site and use the lesson number for the site name.

individual folders for images, PDFs, video, and so on. Dreamweaver assists in this endeavor by including an option for a Default Images folder. Later, as you insert images from other places on your computer, Dreamweaver will use this setting to automatically move the images into the site structure.

● **Note:** In the Jumpstart method you would target the images folder appearing in the lesson folder itself.

- 7 Next to the Default Images Folder field, click the folder (📁) icon. When the dialog box opens, navigate to the DW-CC > images folder containing the files you downloaded from the *Adobe Dreamweaver CC Classroom in a Book* online resources.

You've entered all the information required to begin your new site. In subsequent lessons, you'll add more information to enable you to upload files to your remote site and test dynamic webpages.

- 8 In the Site Setup dialog box, click Save.

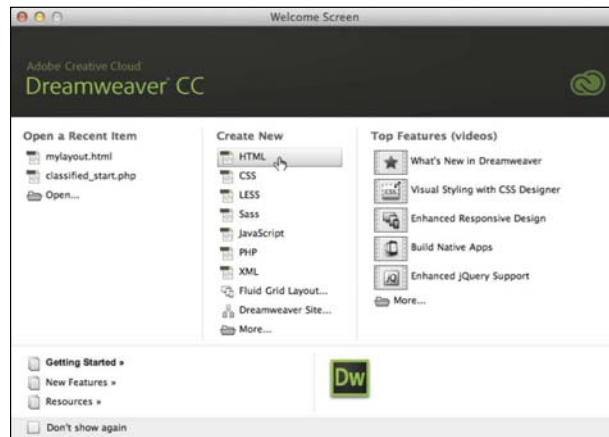
The site name DW-CC now appears in the site list pop-up menu in the Files panel.

Setting up a site is a crucial first step in beginning any project in Dreamweaver. Knowing where the site root folder is located helps Dreamweaver determine link pathways and enables many site-wide options, such as orphaned-file checking and Find and Replace.

Using the Welcome screen

The Dreamweaver Welcome screen provides quick access to recent pages, easy creation of a range of page types, and a direct connection to several key Help topics. The Welcome screen appears when you first start the program or when no other documents are open. Let's use the Welcome screen to explore ways you can create and open documents.

- 1 In the Create New column of the Welcome screen, click HTML to create a new, blank HTML page instantly.



- 2 Choose File > Close.

The Welcome screen reappears.

- 3 In the Open A Recent Item section of the Welcome screen, click the Open button.



This feature allows you to browse for files to open in Dreamweaver.

- 4 Click Cancel.

The Welcome screen shows you a list of up to nine of your recently opened files; however, your installation may not display any used files at this point. Choosing a file from this list is a quick alternative to choosing File > Open when you want to edit an existing page.

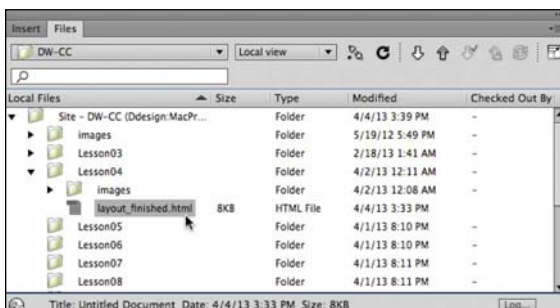
You may use the Welcome screen at any time while working in this book. When you've completed the lessons, you may prefer not to use the Welcome screen, or even to see it. If so, you can disable it by selecting the Don't Show Again option. To re-enable the Welcome screen, access the General category of the Dreamweaver Preferences panel.

Previewing your completed file

To understand the layout you will work on in this lesson, preview the completed page in Dreamweaver.

- 1 In Dreamweaver, press F8 to open the Files panel, and select DW-CC or the Jumpstart name from the site list.
- 2 In the Files panel, expand the Lesson04 folder.
- 3 Double-click **layout_finished.html** to open it.

Note: If you are using the Jumpstart method, you will already be in the Lesson04 folder.



● **Note:** If for some reason you can't or don't want to use an HTML5-based layout, see the sidebar "Alternate HTML 4 workflow" later in this lesson.

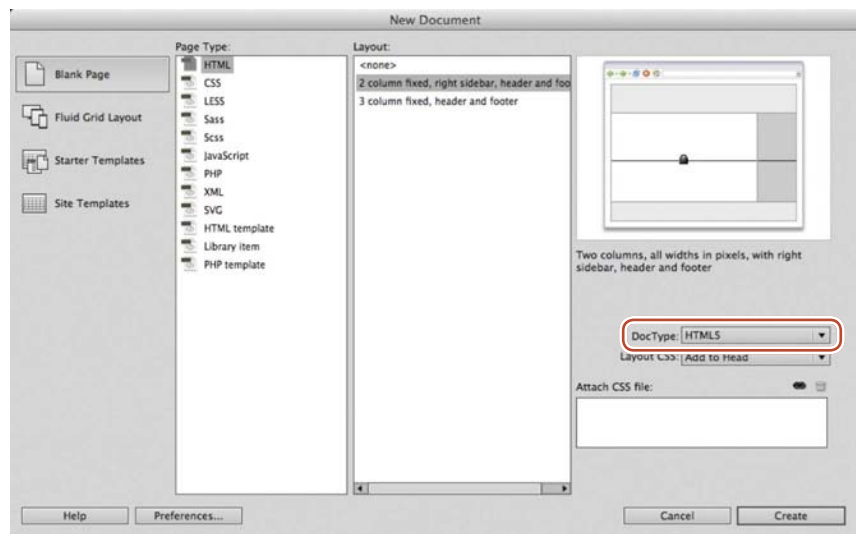
This page represents the completed layout you will create in this lesson. It is based on the wireframe drawings made earlier in this lesson and uses one of the new Dreamweaver HTML5 CSS layouts. Take a few moments to familiarize yourself with the design and components on the page. Can you determine what makes this layout different from existing HTML 4–based designs? You will learn the differences as you work through this lesson.

- 4 Choose File > Close.

Modifying an existing CSS layout

The predefined CSS layouts provided by Dreamweaver are always a good starting point. They are easy to modify and adapt to most projects. Using a Dreamweaver CSS layout, you will create a proof-of-concept page to match the final wireframe design. This page will then be used to create the main project template in subsequent lessons. Let's find the layout that best matches the wireframe.

- 1 Choose File > New.
- 2 In the New Document dialog box, select Blank Page > HTML.



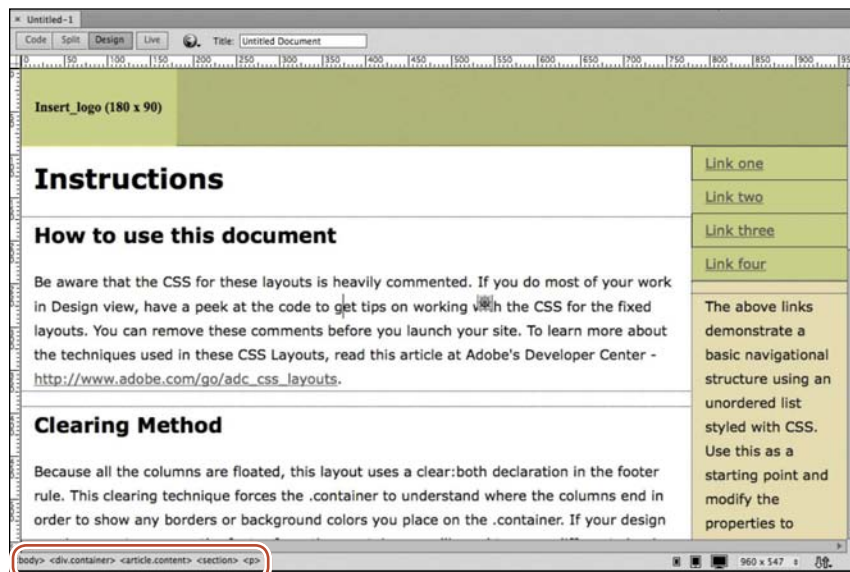
At the time of this writing, Dreamweaver CC offers two HTML5-based CSS layouts. The exact number and features of these layouts may change over time through automatic updates via Creative Cloud. The changes to this list may occur without notice or fanfare, so keep your eyes peeled for new options in this dialog box.

The featured HTML5 layouts use some of the new semantic content elements and will help you get some experience with this evolving standard. Unless you need to support an installed base of older browsers (like IE5 and 6), there's little to worry about using the newer layouts. Let's choose one of the HTML5 layouts that best fits the needs of the new site.

The layout "HTML: 2 column fixed, right sidebar, header and footer" has the most in common with the target design. The only difference is that the sidebar element is aligned to the right of the layout instead of to the left. You will align this element to the left later in this lesson.

- 3 Select **HTML: 2 column fixed, right sidebar, header and footer** from the layout list. Click Open/Create.
- 4 Switch to Design view, if necessary.
- 5 Insert the cursor anywhere in the page content. Observe the names and order of the tag selectors at the bottom of the document window.

● **Note:** If for some reason you can't or don't want to use an HTML5-based layout, see the sidebar "Alternate HTML 4 workflow" later in this lesson.



The display order of elements in the tag selector directly correlates to the page's code structure. Elements appearing to the left are parents, or containers, of all elements to the right. The element farthest to the left is the highest in the page structure. As you can see, the `<body>` element is highest and `<div.container>` is second.

As you click around the page sections, you will be able to determine the HTML structure without having to delve into the Code view window at all. In many ways, the tag selector interface makes the job of identifying the HTML skeleton much easier, especially in complex page designs.

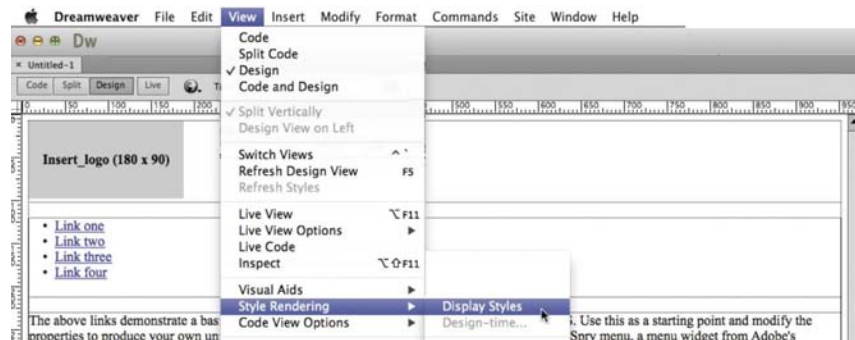
Semantics is all in the name

In HTML5 you will see several new *semantic* elements you may not be familiar with yet, such as `<section>`, `<article>`, `<aside>`, and `<nav>`. In the past, you would have seen `<div>` elements identified and differentiated with `class` or `id` attributes, such as `<div class="header">` or `<div id="nav">`, to make it possible to apply CSS styling. HTML5 has simplified this construction down to `<header>` and `<nav>`. By using elements that are named for specific tasks or types of content, you can streamline code construction while achieving other benefits as well. For example, as search engines, such as Google and Yahoo, are optimized for HTML5, they will be able to locate and identify specific types of content on each page more quickly, making your site more useful and easier to browse.

The page consists of four main content elements, three subsections, and a single element that wraps around all the others. All but one of these are new HTML5 elements, including `<header>`, `<footer>`, `<nav>`, `<aside>`, `<article>`, and `<section>`. The only `<div>` elements in this layout are being used to hold the sidebar content and to hold everything together. Using these new elements means that you can apply complex CSS styling while reducing the complexity of the code overall. You can still use `class` and `id` attributes, but the new semantic elements reduce the need for this technique.

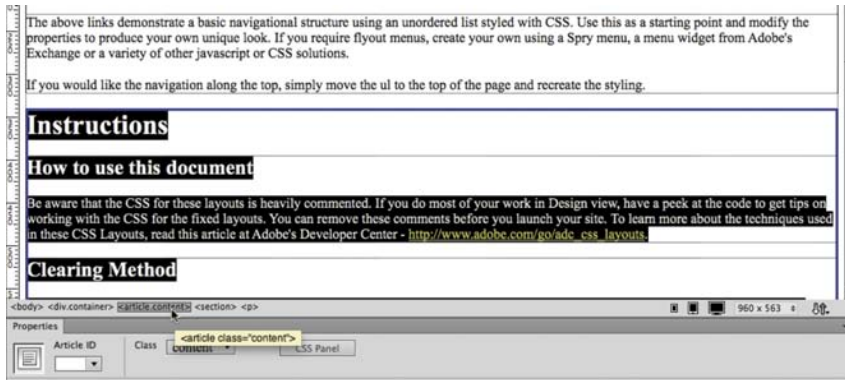
To understand exactly how much this design depends on CSS, sometimes shutting off CSS styling is a good idea.

- 6 Choose View > Style Rendering > Display Styles to disable CSS styling in Design view.



Style display is typically on by default (showing a check mark in the menu). By clicking this option in the menu, you'll toggle CSS styling off temporarily.

7 Note the identity and order of each page component.



Without CSS, the HTML skeleton is exposed for all to see. It's instructive to know what the page will look like if somehow the cascading style sheet is disabled or not supported by a particular browser. Now it's easier to identify the page components and their structure. Although it is not strictly required, items that display higher on the page, like `<header>`, usually are inserted before other elements that appear lower, like `<footer>`.

Another important aspect you should notice is the navigation menu. Without the CSS styling, the navigation menu reverted back to a simple bulleted, or unordered, list with hyperlinks. Not too long ago this menu would have been built with tables, images, and complex rollover animation. If the images failed to load, the menu usually became a jumbled, unusable mess. The hyperlinks continued to work, but without the images there were no words to tell users what they were clicking. But navigation built on text-based lists, on the other hand, will always be usable, even without styling.

8 Choose View > Style Rendering > Display Styles to turn on CSS styling again.

Getting into the habit of saving files before you modify any settings or add content is always a good idea. Dreamweaver doesn't offer a backup or recovered-file feature; if it crashes before you save, all your work in any open, unsaved file will be lost. Save your files on a regular basis to prevent the loss of data and important changes to your files.

Alternate HTML 4 workflow

HTML5 is coming on strong all over the Internet and for most applications the suggested workflow will work perfectly well. But HTML5 is not the current web standard and some pages or components may not display properly on certain older browsers and devices. If you'd rather work with code and structures that are more tried and true, feel free to substitute HTML 4–based components for the HTML5 elements.

However, if you create this layout you'll have to adapt the steps within all the following lessons and exercises to the new components and structures. For example, HTML5 uses the new semantic elements, such as the following:

```
<header>...</header>
<footer>...</footer>
<section>...</section>
<article>...</article>
<nav>...</nav>
```

For an HTML 4–compatible layout, you would substitute a generic `<div>` element and use a `class` attribute that identifies the component this way:

```
<div class="header">...</div>
<div class="footer">...</div>
<div class="section">...</div>
<div class="article">...</div>
<div class="nav">...</div>
```

You'll also have to adapt the CSS styling for the new HTML 4 elements by modifying or rebuilding the HTML5-based selector names (header, footer, nav, and so on).

That way, the CSS rule `header {color:#090}` becomes `.header { color:#090 }` instead.

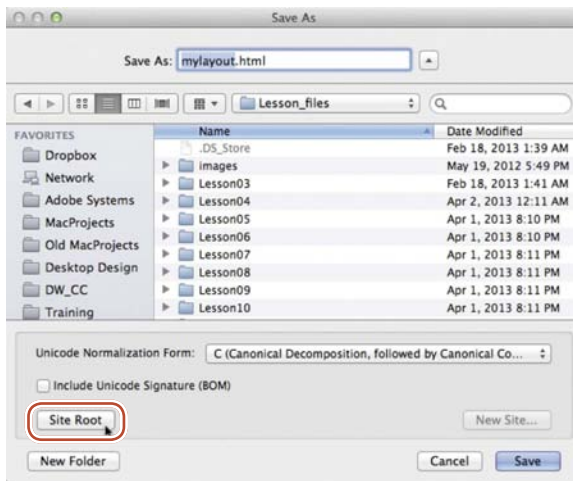
With all the caveats out of the way, the ugly truth is that even when you use standard HTML 4 code and components, older browsers and certain devices will still fail to render some of them properly. Some web designers believe that the longer we persist in using the older code, the longer the older software and devices will hang around making our lives difficult and delaying the inevitable adoption of HTML5. These designers say we should abandon the older standards and force users to upgrade as soon as possible.

The final decision is yours or your company's to make. In most cases, the problems you experience with HTML5 will be minor flaws—a font that's too big or too small—not a complete meltdown.

For more information about the differences between HTML 4 and HTML5, check out the following links:

- <http://tinyurl.com/html-differences>
- <http://tinyurl.com/html-differences-1>
- <http://tinyurl.com/html-differences-2>

- 9 Choose File > Save. In the Save As dialog box, navigate to the site root folder, if necessary. Name the file **mylayout.html** and click Save.



Note: Dreamweaver may try to save this file back to the Lesson04 folder; if this is not the site root folder, click the Site Root button to navigate to the proper location.

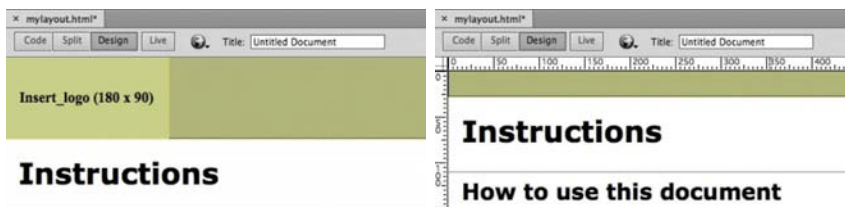
Dreamweaver normally saves HTML files to the default folder specified in the site definition, but double-check the destination to make sure your files end up in the right place. All HTML pages created for the final site will be saved in the site root folder.

Adding a background image to the header

CSS styles are the current standard for all web styling and layout. In the following exercises, you'll apply background colors and a background image to a page section, adjust element alignment and the page width, and modify several text properties. All these changes are accomplished using Dreamweaver's CSS Designer panel (new with Dreamweaver CC).

If you start at the top of the page and work down, the first step would be to insert the graphical banner that appears in the final design. You could insert the banner directly into the header, but adding it as a background image has the advantage of leaving that element open for other content. It will also allow the design to be more adaptable to other devices, like cell phones and other mobile devices.

- 1 If necessary, switch to Design view. Select the image placeholder **Insert_logo** (180x90) in the header. Press Delete.

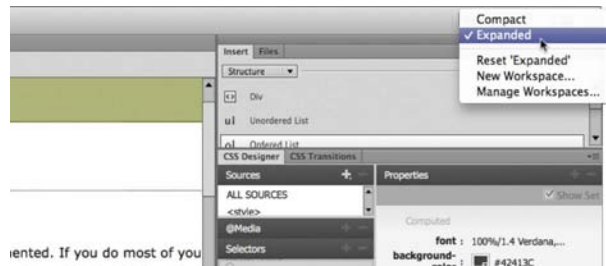


When you delete the image placeholder, the empty header will collapse to a fraction of its former size because it has no CSS height specification. You can identify all the formatting assigned to the layout components by using the new CSS Designer.

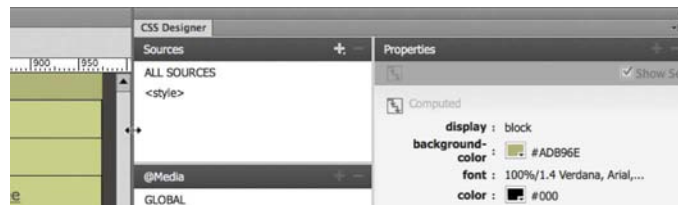
- 2 If necessary, choose Window > CSS Designer to display the panel.

To maximize the effectiveness of the CSS Designer you can use one of the predefined workspaces.

- 3 From the Workspace menu, choose Expanded.

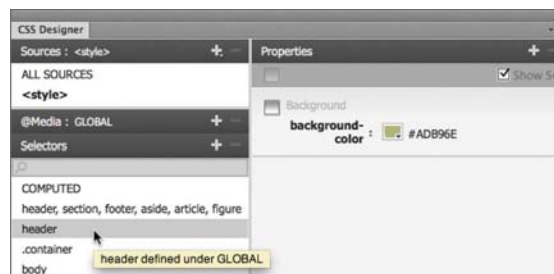


The Dreamweaver workspace changes to display the CSS Designer in a two-column layout. This design will give you extra room to work with the CSS styling. If you want to increase the width of the CSS Designer, you can drag the edge of the document window to the left.



From the tag selector display, you can see that the element contained the image placeholder. Examine the CSS Designer panel. Can you identify any CSS rules that may format the header element?

- 4 In the CSS Designer panel, select the header rule in the Selectors pane. Examine the CSS properties applied to the element.



The Properties pane of the CSS Designer displays any existing specifications. For header it shows only a background color assigned to it. The panel also allows you to create new specifications. It works in two basic modes. If you are familiar with CSS syntax, you can create specifications by typing them directly. Otherwise, you can display a complete list of available CSS properties in the panel and define them as you go. You will experiment with both methods in this lesson.

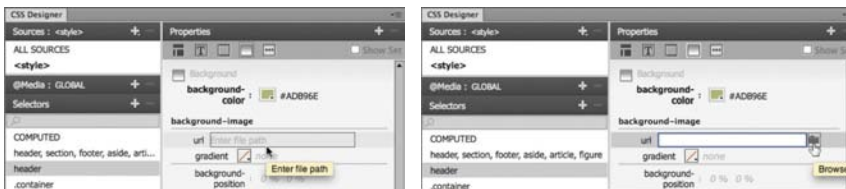
First, you will add a background image to the header element and then adjust its size. To see a complete list of CSS properties, make sure the Show Set option is not enabled.

- 5 If necessary, deselect the Show Set option in the Properties pane.



When Show Set is deselected, the Properties pane displays the list of available CSS specifications. The list is organized into five categories: Layout, Text, Border, Background, and Others. To focus the display on a particular category, you can use the navigation icons at the top of the Properties pane.

- 6 Click the Background (■) category icon. In the background-image section, click the text *Enter file path* beside the URL property. Click the Browse button next to the URL field.



- 7 In the Select Image Source dialog box, navigate to the default images folder and select **banner.jpg** and note the dimensions of the image in the preview.

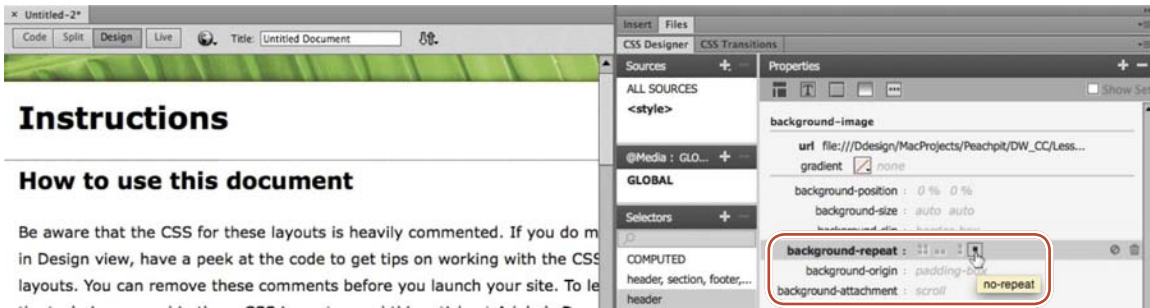
The image is 950 by 130 pixels.

- 8 Click OK/Open to select the background image.

Background images repeat both vertically and horizontally by default. This isn't a problem at the moment, but to ensure that this behavior doesn't cause any undesirable effects in the future, you'll need to change the repeat specification.

Note: You may need to modify your folder display to see the pixel dimensions of the image. In Windows, hover the cursor over the image to display its size. On the Mac, select column view within the dialog box.

- 9 In the background-repeat options, click the no-repeat icon.



The background image appears in the `<header>` element. The element is wide enough but not tall enough to display the entire background image. Since background images aren't truly inserted into an element, they have no effect (positive or negative) on the size of it. To ensure that the `<header>` is large enough to display the entire image, you need to add a height specification to the header rule.

- 10 If necessary, select the header rule in the Selectors pane. In the Properties pane, click the Layout (📐) icon.

The Properties pane provides a list of CSS layout specifications you can set.

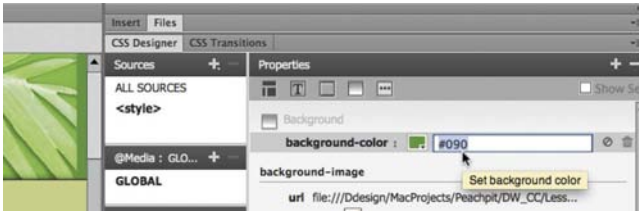
- 11 In the Layout category Height field, choose px from the measurement pop-up list. Enter 130 and press Enter/Return.



The height of the `<header>` element immediately adjusts, displaying the full banner image. Note that the image is slightly narrower than the container. You'll adjust the width of the layout later. You don't want to set the width on the `<header>` element itself. You learned in Lesson 3, "CSS Basics," that the width of block elements, like `<header>`, defaults to the entire width of their parent element. Let's add some finishing touches to the element.

You may have noticed that the `<header>` element already contained a background color that doesn't really match your site color scheme. Let's apply one that does.

- 12 In the Properties pane, click the Background category icon. Replace the existing background-color specification with **#090** and press Enter/Return to complete the change.



A bit of the background color is peeking out from the right edge of the banner, but once you adjust the width of the layout you won't see this color at all unless the background image fails to load. Adding background colors like this is a common precaution since certain devices or browsers may not load images and/or background graphics by default.

- 13 Choose File > Save.

Inserting new components

The wireframe design shows two new elements that don't exist in the current layout. The first contains the butterfly image, the second the horizontal navigation bar. Did you notice that the butterfly actually overlaps both the header and the horizontal navigation bar? There are several ways to achieve this effect. In this case, an absolutely positioned (AP) div will work nicely.

Note: To better understand how this technique works, try this step in Split view.

- 1 Insert the cursor into the header, if necessary. Select the `<header>` tag selector. Press the Left Arrow key once.

This procedure inserts the cursor in the HTML code before the opening `<header>` tag. If you had pressed the Right Arrow key, the cursor would move outside the closing `</header>` tag instead. Remember this technique—you'll use it frequently in Dreamweaver when you want to insert the cursor in a specific location before or after a code element without resorting to Code view. Always remember that your webpage is actually created by elements defined by HTML code and CSS. Knowing how to create, edit, and insert elements in the proper way will result in clean and error-free code.

AP-divs used to be a popular feature in previous versions of Dreamweaver, but the built-in workflow was deprecated in the latest version. This change is primarily in response to the industry-wide move away from fixed width and absolutely positioned components and toward flexible or fluid designs. But, for this application in the current site an AP-div is still a valid option. Later, we'll explore ways to deal with this element for mobile devices.

● **Note:** AP-divs were used extensively in the past to create highly structured, fixed-layout web designs. This technique has declined dramatically in recent years as the need to support cell phones and other mobile devices has increased. For certain applications, AP-divs are still handy.

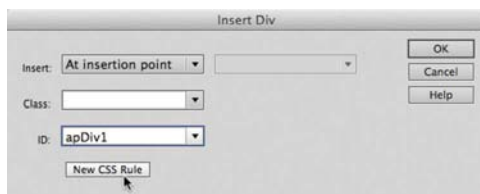
- 2 Choose Insert > Div.

The Insert Div dialog box appears. The AP-div will be the only one on this page. The positioning and formatting will be unique to it. Let's use an ID to name the element.

- 3 Enter **apDiv1** in the ID field.

The Insert Div dialog box allows you to create the CSS rule immediately to format the AP-div.

- 4 Click the New CSS Rule button.

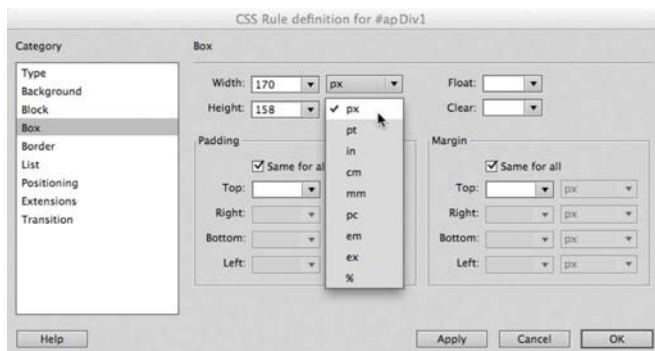


The New CSS rule dialog box appears. The ID apDiv1 appears automatically in the Selector Name field. ID-based selectors have the highest specificity, so the formatting you create in this dialog box will not affect any other elements.

- 5 Click OK to create the CSS rule.

The CSS Rule Definition dialog box for #apDiv1 appears. This dialog box allows you to quickly create the CSS specifications you will need for the AP-div.

- 6 Select the Box category. Enter **170** in the Width field. Choose px as the unit of measurement. Enter **158** in the Height field.



These settings set the width and height of the div that will hold the butterfly logo image.

- 7 Deselect the Same For All option for Margin.
- 8 Enter **15 px** in the Top and Left margin fields.

These settings help to position the AP-div in the proper location from the top and left of the layout. But the most important of all the settings are in the Positioning category.

- 9 Select the Positioning category. Choose Absolute from the Position pop-up menu.

By selecting Absolute you are effectively removing the element from the regular document flow. An absolutely positioned element can be placed almost anywhere within its parent structure regardless of other elements on the page.

Once the `div` has been positioned absolutely, you then have to decide whether it will appear above or below other elements. The property that controls this attribute is *z-index*. Normally, all elements in a layout appear at the same level; they all have a *z-index* of zero (0). But the AP-div needs to float above the other elements. By giving the AP-div a *z-index* greater than zero, you will ensure that it appears higher than the other elements.

- 10 Enter **1** in the *z-index* field. Click OK to complete the rule definition.

The CSS Rule Definition dialog box closes, displaying the Insert Div dialog box again.

- 11 Click OK to insert the AP-div.



The AP-div appears in the layout displaying the placeholder text *Content for id “apDiv1” Goes Here*, which is selected and ready to be replaced.

- 12 Press Delete to remove the placeholder text.
- 13 Choose Insert > Image > Image. Navigate to the default images folder and select **butterfly-ovr.png**.

- 14 Click OK/Open.



The butterfly logo appears in the AP-div. Thanks to the absolute positioning and z-index, the butterfly appears above the banner and other layout elements.

Best web practices call for the use of alternate text to describe images for accessibility purposes. You can enter this attribute directly in the Property inspector.

- 15 In the Property inspector enter **GreenStart Logo** in the Alternate text field. Save the file.

The `<div#apDiv1>` is complete. Now, let's add another new component that will hold the horizontal navigation shown in the site design specs. The vertical navigation menu will hold links to the organization's products and services. The horizontal navigation will be used to link back to the organization's home page, mission statement, and contact information.

Inserting a navigation component

In HTML 4, you probably would have inserted the links into another `<div>` element and used a `class` or `id` attribute to differentiate it from the other `<div>` elements in the file. Instead, HTML5 provides a new element geared specifically toward such components: `<nav>`.

- 1 Insert the cursor into the header. Click the `<header>` tag selector. Press the Right Arrow key.

The cursor now appears after the ending `</header>` tag.

- 2 Choose Insert > Structure > Navigation.

The Insert Navigation dialog box appears.

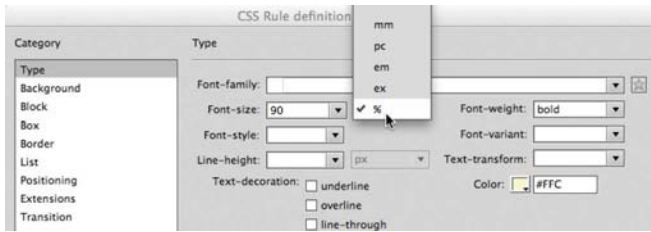
- 3 Enter **top-nav** in the Class field. Click New CSS Rule.

The New CSS Rule dialog box appears.

- 4 Click OK to create the **top-nav** class.

The "CSS Rule Definition for *.top-nav*" dialog box appears.

- 5 In the Type category, enter **90** in the Font-size field and choose the percentage sign (%) from the pop-up. Enter **#FFC** in the Color field. Choose bold from the Font-weight pop-up.



- 6 Type **#090** in the background-color field.
- 7 In the Block category, choose Right from the Text-align pop-up.
- 8 In the Box category, deselect the Same For All check box for Padding. Enter **5 px** in the Top padding fields. Enter **20 px** in the Right padding fields. Enter **5 px** in the Bottom padding fields.
- 9 In the Border category, deselect the Same For All check boxes for Style, Width, and Color. Enter the following values only in the corresponding Bottom border fields: **solid, 2 px, #060**
- 10 Click OK in the CSS Rule Definition dialog box. Click OK in the Insert Navigation dialog box
- A <nav> element appears displaying the placeholder text *Content for class "top-nav" Goes Here*. The new element and placeholder text is already formatted based on the specifications you created in the CSS .top-nav rule.
- 11 Type **Home | About Us | Contact Us** to replace the placeholder text. In the Property inspector, choose Paragraph from the Format pop-up menu.

► **Tip:** To enter separate values in the Bottom field, remember to deselect the Same For All check boxes in each section first.

● **Note:** The <nav> element is new in HTML5. If you need to use HTML 4 code and structures, see the sidebar "Alternate HTML 4 workflow" earlier in this lesson.



You will convert this text to actual hyperlinks in Lesson 9, "Working with Navigation." For now, let's create a new CSS rule to format this element.

- 12 Press Ctrl-S/Cmd-S to save the file.

As you can see, adding new components to the CSS layouts is fairly easy, which is what makes them a good starting point for a new project. In the following exercises, you will explore other ways you can customize the predefined layout.

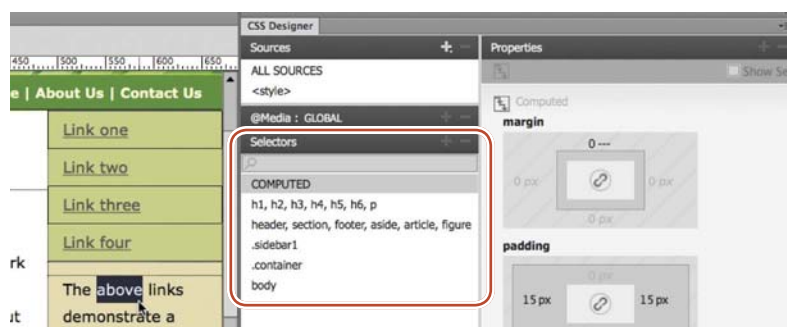
Changing element alignment

The proposed design calls for the sidebar to appear on the left side of the page, but this layout puts it on the right. However, adjusting the layout is a lot easier than you may think. The first step is to determine what existing CSS rule is responsible for the current alignment.

- 1 If necessary, choose Window > CSS Designer to display the panel.

The CSS Designer provides the capability to format HTML components by creating and editing CSS rules. But you can also use it to inspect existing styling. If you click a selector in the list, the Properties pane displays formatting contained in the rule.

- 2 In Design view, insert the cursor anywhere in the right sidebar.
- 3 Examine the Selectors pane of the CSS Designer.

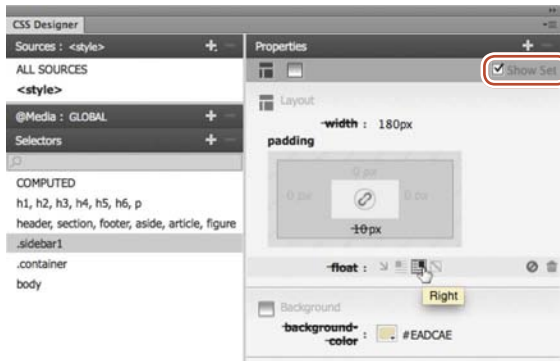


The Selectors pane displays a list of all the CSS rules that have an effect, even minimally, on the targeted element, with the rule at the top of the list having the strongest influence. The list also features the option COMPUTED. Selecting this option displays the aggregated formatting from all rules in the list.

- 4 In the Selectors pane, select the body rule. In the Properties pane, select the Show Set option.

The Show Set option limits the Properties pane display only to properties set by the selected rule.

- 5 Click each rule in the list until you find the one controlling the float property.



The `.sidebar1` rule applies the `float:right` property.

- 6 Select the `.sidebar1` rule, and change the `float` property from `right` to `left`.

The sidebar moves to the left side of the layout.

- 7 Save the file.

With each modification the layout is getting closer to the site design.

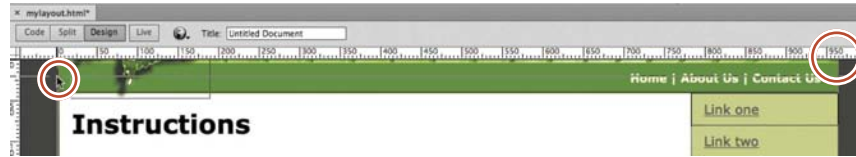
Modifying the page width and background color

Before you convert this file into the project template, let's tighten up the formatting and the placeholder content. For example, the page width has to be modified to match the banner image. But first, you'll have to identify the CSS rule that controls the page width.

- 1 If necessary, choose **View > Rulers > Show** or press **Alt-Ctrl-R/Option-Cmd-R** to display the rulers in the Design window.

You can use the rulers to measure the width and height of HTML elements or images. The orientation of the rulers defaults to the upper-left corner of the Design window. To give you more flexibility, you can set this zero point anywhere in the Design window.

- 2 Position the cursor over the axis point of the horizontal and vertical rulers. Drag the crosshairs to the upper-left corner of the header element in the current layout. Note the width of the layout.



Using the ruler, you can see that the layout is between 960 and 970 pixels wide.

● **Note:** When you select each tag selector, observe how the CSS Designer display updates to show you any applied styling.

- 3 Insert the cursor into any content area of the layout.
Observe the tag selector display to locate any elements that may control the width of the entire page; it would have to be an element that contains all the other elements. The only elements that fit this criterion are `<body>` and `<div.container>`.
- 4 Click each of the tag selectors displayed at the bottom of the document window. Examine the Properties pane display in the CSS Designer for each element. If necessary, select the Show Set option to truncate the display to applied properties only.

Can you identify the rule that controls the width of the entire page?

The `.container` rule seems to match the description; it contains the `width: 960px` declaration. By now you should be getting good at CSS forensics using the tag selector interface and the CSS Designer.

You can edit rules individually as you did in previous exercises, or you can use the display within the COMPUTED option.

- 5 Click the `<div.container>` tag selector. In the Properties pane, select the COMPUTED option. Change width to **950 px** and press Enter/Return to complete the specification.



The `<div.container>` element now matches the width of the banner image, but you may have experienced an unintended consequence when you changed the overall width. In our example, the main content area shifted down below the sidebar. To understand what happened, you'll have to do a quick investigation.

- 6 In the Sources pane of the CSS Designer panel, select `<style>`.

This displays all the CSS rules defined in the current page. At the moment, all rules are embedded in the `<head>` section.

- 7 Click the `.content` rule and check its properties. Note its width: 780 pixels.

- 8 Click the `.sidebar1` rule and check its width: 180 pixels.

Combined, the two `<div>` elements total 960 pixels, the same as the original width of the layout. The elements are too wide to sit side by side in the main container and thereby prompted the unexpected shift. This type of error is common in web design and is easily fixed by adjusting the width of either of the two child elements.

- 9 In the CSS Designer panel, click the `.content` rule. In the Properties section of the panel, change the width to **770 px**.

The `<div.content>` element returns to its intended position. This was a good reminder that the size, placement, and specifications of page elements have important interactions that can affect the final design and display of your elements and of the entire page.

The current background color of the page detracts from the overall design. Let's remove it.

- 10 In the CSS Designer, select the `body` rule. In the Background category, change the background-color to `#FFF` and press Enter/Return.

Note how the absence of the background color gives the impression that the page's content area drifts off into the wide expanse. You could give `<div.container>` a different background color, or you could simply add a border to give the content elements a definitive edge. Let's add a thin border to the element.

- 11 In the CSS Designer, select the `.container` rule. If necessary, select Show Set in the Properties pane.

- 12 Click the Add CSS Property (+) icon.

An empty field appears in the Properties pane.

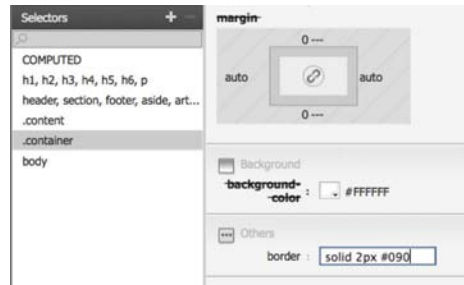
- 13 Type **border** and press Enter/Return to create the new property.

An empty value field appears.

► **Tip:** You can select the desired property from the hint list any time while you are typing. Use your mouse and double-click or arrow down the list and press Enter/Return.

► **Tip:** In many cases, you can enter values manually as shown or select them from options displayed within the Properties pane.

- 14 Type **solid 2px #090** and press Enter/Return to create the new value.



A dark green border appears around `<div.container>`.

- 15 Save the file.

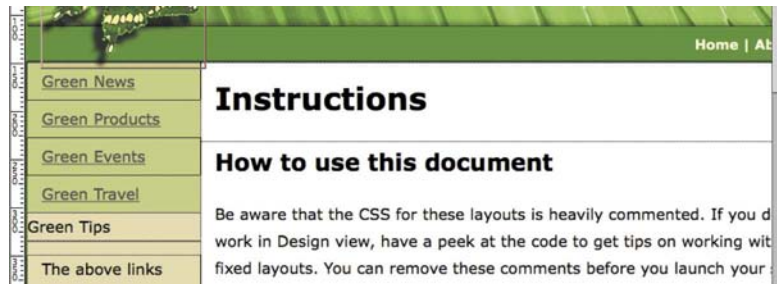
Modifying existing content and formatting

As you can see, the CSS layout comes equipped with a vertical navigation menu. The generic hyperlinks are simply placeholders waiting for your final content. Let's change the placeholder text in the menu to match the pages outlined in the thumbnails created earlier and modify the colors to match the site color scheme.

- 1 Select the placeholder text *Link one* in the first menu button. Type **Green News**. Change *Link two* to read **Green Products**. Change *Link three* to read **Green Events**. Change *Link four* to read **Green Travel**.

One of the advantages of using bulleted lists as navigational menus is that inserting new links is easy.

- 2 With the cursor still at the end of the words *Green Travel*, press Enter/Return. Type **Green Tips**.



The new text appears in what looks like a button structure, but the background color doesn't match and the text doesn't align with other menu items. You could probably figure out what's wrong in Design view, but in this case, the problem may be identified faster in Code view.

- 3 Click the `` tag selector for the new link item, and select Code view. Observe the menu items and compare the first four with the last one. Can you see the difference?

```
147 <div class="sidebar1">
148 <nav>
149   <ul>
150     <li><a href="#">Green News</a></li>
151     <li><a href="#">Green Products</a></li>
152     <li><a href="#">Green Events</a></li>
153     <li><a href="#">Green Travel</a></li>
154     <li>Green Tips</li>
155   </ul>
156 </nav>
```

The difference is obvious in Code view. The last item is tagged with the `` element like the others—as part of the bulleted list—but it doesn’t feature the markup `` used in the other items to create the hyperlink placeholder. For *Green Tips* to look like the other menu items, you have to add a hyperlink, or at least a similar placeholder.

- 4 Select the text *Green Tips*. In the Link field of the HTML Property inspector, type `#` and press Enter/Return.

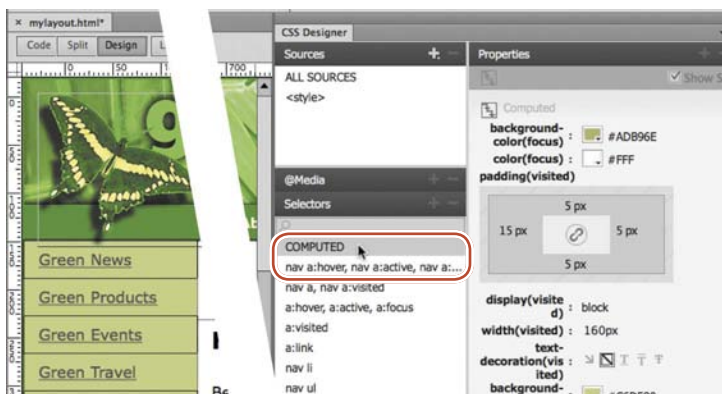
The code in all the items is identical now.

- 5 Switch to Design view.

All the menu items are identically formatted now. You’ll learn more about how to format text with CSS to create a dynamic HTML menu in Lesson 5, “Working with Cascading Style Sheets.”

The current menu color doesn’t match the site color scheme. To change the color, let’s use the CSS Designer to find the CSS rule that controls this formatting.

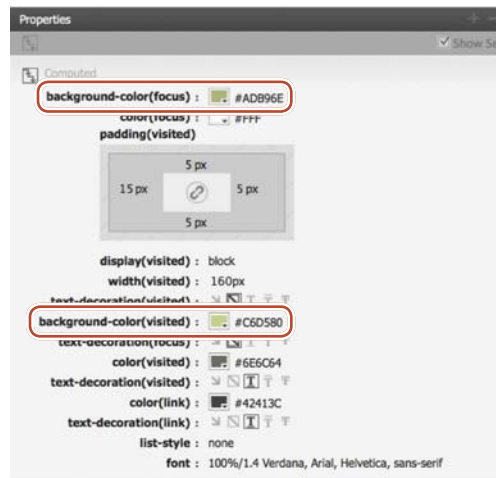
- 6 Insert the cursor into any of the menu items. If necessary, select the COMPUTED option in the Properties pane of the CSS Designer. If necessary, select the Show Set option.



The Properties pane displays properties assigned to the navigation menu. By using the COMPUTED option, you are seeing the aggregated styling of all the applicable rules in one place, making it a simple matter to make the desired changes.

But be careful, in some cases, the styling shown may not affect the element directly, but via inheritance. If you change a specification within the COMPUTED option, the change affects the applicable rule. Be aware that such modifications may produce undesirable results, changing other elements on the page. Keep your eyes peeled for any unintended consequences.

Before you make any changes, it's important to understand how some of the rules are being applied to the page elements. If you look closely at the COMPUTED option in the Properties pane, you'll notice that more than one background color is applied to the element. How can that be?



The reason you see two background colors is because the item is a hyperlink that actually changes formatting in response to user interaction. When the user moves his mouse over the menu item, the background color changes. It can also change when the user visits a link. Although in this case, the menu shows the same color for visited or unvisited links. In Lesson 5, “Working with Cascading Style Sheets,” you will explore how these different effects are applied. For now, let’s just change the default state of the link.

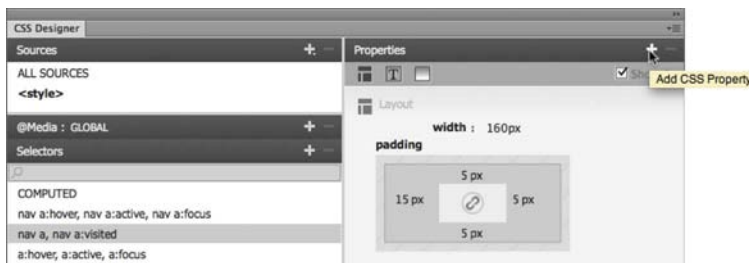
- 7 In the Properties pane, change background-color (visited) to #090 and press Enter/Return.

The background-color of the menu items now matches the horizontal <nav> element. But the black text is difficult to read against the green background color. As you see in the horizontal menu, a lighter color would be more appropriate. Let’s change the text color for the link, too.

If you examine the COMPUTED properties for the ``, you will notice four different “color” properties set for it. Confused? Each of the properties is being applied to and, in some cases, inherited by the text in the menu item. However, before you change the color of any of these properties you should examine the individual rules to identify the correct one.

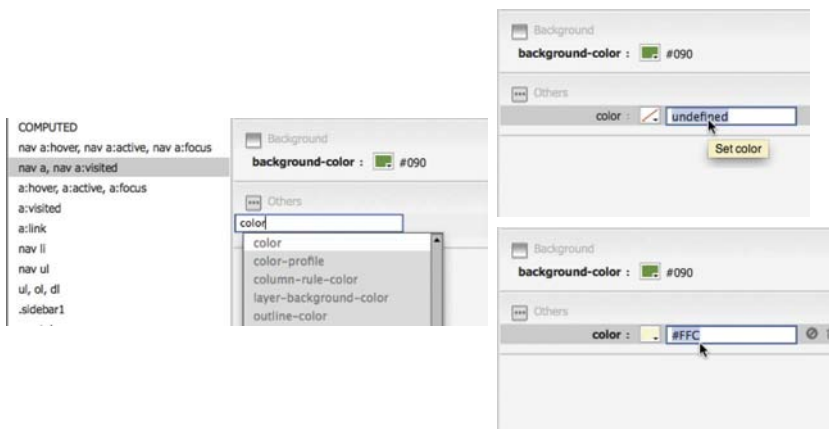
In this instance, the rules that are most important are #1 and #2 in the list. They both apply to the link text in the vertical menu. For now you only need to change the default color of the links themselves.

- 8 Select the `nav a`, `nav a:visited` rule. Examine the properties assigned to it. The rule has no color specification. The link text color is being inherited from another rule. Let’s set a new color specification.
- 9 Click Add CSS Property (+).



A new property field appears.

- 10 Type `color` and press Enter/Return.



A new color property appears with an empty value.

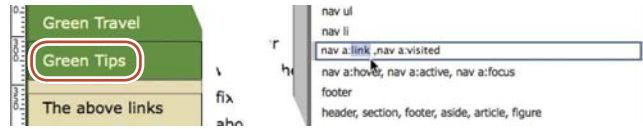
● **Note:** The CSS notation `a:link` is one of four pseudo-selectors used to format various default hyperlink behaviors. You will learn more about these pseudo-selectors in Lesson 5, “Working with Cascading Style Sheets.”

- 11 Click the *undefined* label and enter `#FFC` and press Enter/Return.

The link text does not change color as expected. Unfortunately, Dreamweaver missed a problem in the style sheet. The hyperlink display in Design view is currently honoring the formatting in the `a:link` rule, which applies default formatting to hyperlinks on the page. But the CSS Designer shows the `nav a`, `nav a:visited` rule, which turns off the text decoration, higher in the list. Confused?

The selectors `a` and `a:link` are supposed to be equivalent; they both format the default state of hyperlinks. Yet, in a battle between `a` and `a:link`, `a:link` will always win. So, then why is the `nav a`, `nav a:visited` rule listed higher than `a:link`? Because the rule combines two selectors: `nav a` and `nav a:visited`. Although the property `a:visited` is equal in specificity to `a:link`, combining two selectors gives the rule a higher rating than a rule with just one (even though part of the rule actually has lower specificity). Whatever the reason, the links are still formatted incorrectly. Luckily, there’s an easy fix.

- 12 In the CSS Designer, select `<style>` in the Sources panel. In the Selectors pane, click the `nav a`, `nav a:visited` rule. If you click the selector a second time, the name becomes editable.
- 13 Change `nav a` to `nav a:link` and press Enter/Return to complete the selector.



The link text in the vertical menu now displays in the desired color, and the underline disappears.

- 14 Save the file.

Inserting an image placeholder

The sidebar will feature photos, captions, and short blurbs on environmental topics. Let’s insert a placeholder image and caption below the vertical menu. Dreamweaver no longer provides a built-in feature for creating image placeholders, but you can create one using the Quick Tag Editor or by inserting the code directly into the Code view window. Use the following steps to create the image placeholder.

- 1 Insert the cursor into the text directly below the vertical menu. Click the `<p>` tag selector.

The placeholder image should not be inserted within the `<p>` element. If it were, it would inherit any margins, padding, and other formatting applied to the paragraph, which could cause it to disrupt the layout.

- 2 Press the Left Arrow key.

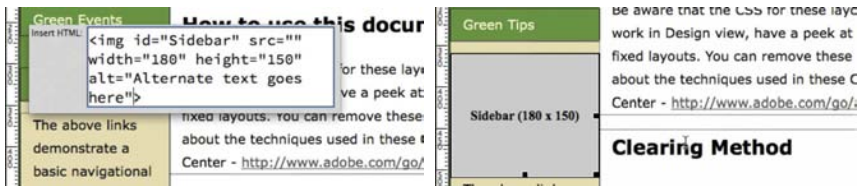
As you have seen in earlier exercises, the cursor moves to the left of the opening `<p>` tag in the code but stays within the `<aside>` element.

- 3 Press Ctrl-T/Cmd-T to open the Quick Tag Editor.

The Quick Tag Editor appears with the text cursor inserted within tag brackets.

- 4 Type `img` and press the spacebar.

- 5 Type `id="Sidebar" src="" width="180" height="150" alt="Alternate text goes here"` and press Return/Enter to complete the image placeholder.



► **Tip:** Use Split view whenever you're unsure where the cursor is inserted.

An image placeholder appears in `<div.sidebar1>` below the vertical menu. When you use this layout to create pages for your actual site, you will replace the placeholder image with an actual image, and update the attributes of this element, as necessary.

- 6 Select all the text below the image placeholder. Type **Insert caption here**.

The caption placeholder replaces the text.

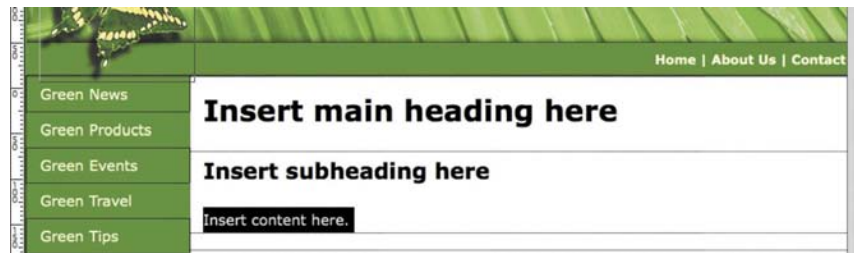
- 7 Press Ctrl-S/Cmd-S to save.

Inserting placeholder text

Let's simplify the layout by replacing the existing headings and text in the main content area.

- 1 Double-click to select the heading *Instructions*. Type **Insert main heading here** to replace the text.
- 2 Select the heading *How to use this document*. Type **Insert subheading here** to replace the text.

- 3 Select the placeholder text in that same <section> element.
Type **Insert content here** to replace it.

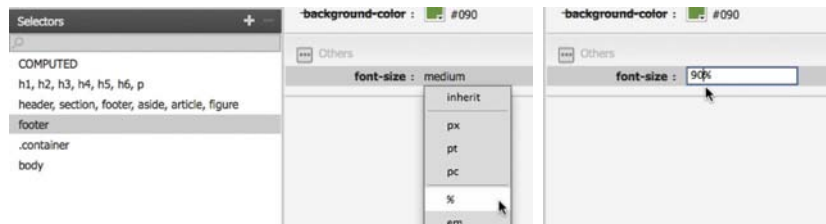


- 4 Insert the cursor in the next section. Click the <section> tag selector. Press Delete. Select and delete the remaining two <section> elements and their contents.
- 5 Press Ctrl-S/Cmd-S to save.

Modifying the footer

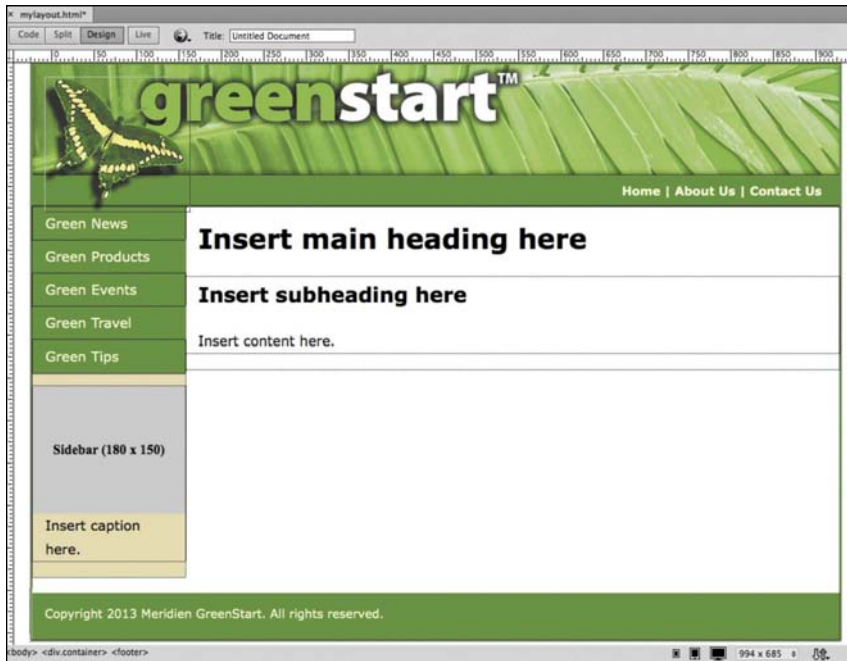
Let's reformat the footer and insert the copyright information.

- 1 In the Sources pane of the CSS Designer panel, select <style>. In the Selectors pane, select the footer rule.
- 2 Change the background-color to #090
- 3 Click the Add CSS Property (+) icon. Type **font-size** and press Enter/Return.
- 4 Click the value field to edit it. Choose % from the pop-up menu. Type **90** in the value field, and press Enter/Return.



- 5 Click the Add CSS Property icon. Type **color** and press Enter/Return.
- 6 Click the value field to edit it. Type **#FFC** in the value field and press Enter/Return.
- 7 Select the placeholder text in the footer. Type **Copyright 2013 Meridien GreenStart. All rights reserved.**
- 8 Delete the <address> element at the bottom of the footer.

- 9 Press Ctrl-S/Cmd-S to save.



The basic page layout is complete.

Validating webpages

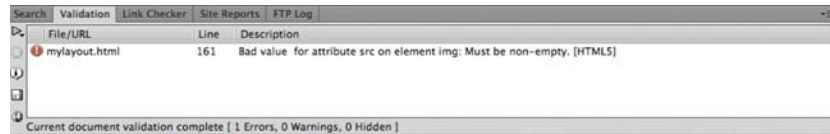
The CSS layouts included with Dreamweaver have been thoroughly tested to work flawlessly in all modern browsers. However, during the lesson you made major modifications to the original layout. These changes could have ramifications in the quality of the code. Before you use this page as your project template, you should check to make sure the code is correctly structured and meets web standards.

- 1 If necessary, open **mylayout.html** in Dreamweaver.
- 2 Choose File > Validation > Validate Current Document (W3C).



A W3C Validator Notification dialog box appears, indicating that your file will be uploaded to an online validator service provided by the W3C. Before clicking OK, you will need to have a live Internet connection.

3 Click OK to upload the file for validation.



After a few moments you receive a report indicating any errors in your layout. The only error you should see is the empty `src` attribute for the image placeholder.

Congratulations. You created a workable basic page layout for your project template and learned how to insert additional components, image placeholders, text, and headings; adjust CSS formatting; and check for browser compatibility. In the upcoming lessons, you will continue to work on this file to complete the site template, tweak the CSS formatting, and set up the template structure.

Review questions

- 1 What three questions should you ask before starting any web design project?
- 2 What is the purpose of using thumbnails and wireframes?
- 3 What is the advantage of inserting the banner as a background image?
- 4 How can you insert the cursor before or after an element without using Code view?
- 5 How does the CSS Designer assist in designing your website layout?
- 6 What advantages does using HTML5-based markup provide?

Review answers

- 1 What is the purpose of the website? Who is the customer? How did they get here?
These questions, and their answers, are essential in helping you develop the design, content, and strategy of your site.
- 2 Thumbnails and wireframes are quick techniques for roughing out the design and structure of your site without having to waste lots of time coding sample pages.
- 3 By inserting the banner or other large graphics as a background image, you leave the container free for other content.
- 4 Select an element using its tag selector, and press the Left Arrow or Right Arrow key to move the cursor before or after the selected element.
- 5 The CSS Designer serves as a CSS detective. It allows you to investigate what CSS rules are formatting a selected element and how they are applied.
- 6 HTML5 has introduced new semantic elements that help to streamline code creation and styling. These elements also allow search engines, like Google and Yahoo, to index your pages more quickly and effectively.

INDEX

SYMBOLS

, (commas), multiple selectors, 169–170
: (colons), generating email forms in PHP, 365
. (period), creating email forms in PHP, 365
/* */ (slash and asterisk), comments in CSS, 71
; (semicolon), CSS rule syntax for declarations, 61
\$ (dollar sign), PHP variables, 364–365
{ } (braces), CSS rule syntax, 60

NUMBERS

1-bit (monochrome) palette, 254
3D visual effect, for menus, 173–174
4-bit color space, 254
8-bit color space, 254–255
16-bit (high-color) palette, 255
24-bit (true color) palette, 255
32-bit color space (alpha transparency), 255–257

A

<a> tag selector
 building menu buttons, 173
 creating descendant selectors, 167–168
 creating dynamic hyperlinks, 168–170
 creating hyperlink text color, 317–319
 creating hyperlinks, 41, 282
 customizing Accordion behavior, 317–318
A:active, 142, 146, 168–172, 183
AATCs (Adobe Authorized Training Centers), 7
Abbreviations, Windows vs. OS X commands, 5
Absolute hyperlinks, 282
Absolute measurements, CSS
 defined, 77
 height, 82–85
 margins and paddings, 85–88
 positioning elements, 88–90
 width, 78–82
ACA (Adobe Certified Associate), 7
Accessibility, adding, 345–348
Accordion. *See* jQuery Accordion widgets
ACE (Adobe Certified Expert), 7
ACI (Adobe Certified Instructor), 7

Actions, adding to forms, 360–363
Active Server Pages. *See* ASP (Active Server Pages)
Add Behavior icon, 302, 308–310
Add Condition icon, 399, 417
Add CSS Property icon
 creating text indents, 228
 modifying background color, 125
 modifying existing content, 129–130
 modifying footer, 132
Add Media Query icon, 397, 417
Add New Server icon, 373, 377
Add Selector icon
 adding behaviors to hyperlinks, 311
 adding rules to media query, 402
 creating custom classes, 159
 creating descendant selectors, 167
 creating dynamic hyperlinks, 168
 creating headings, 220
 creating hyperlink rollover effect, 170
 customizing Accordion behavior, 316, 319
 inserting forms on page, 344
 specifying font size, 156
 styling forms, 367–368
 using Edge Web Fonts, 151
Adobe Authorized Training Centers (AATCs), 7
Adobe Bridge, 262–265
Adobe Certified Associate (ACA), 7
Adobe Certified Expert (ACE), 7
Adobe Certified Instructor (ACI), 7
Adobe Edge Animate, 326–330
Adobe Edge Inspect, 418–419
Adobe Edge Web Fonts, 146, 150–153
Adobe Exchange panel, 302–303
Adobe Extension Manager CC, 302
Adobe Flash
 FLV as fallback format for older browsers, 333
 history of and present time, 324
Adobe Muse, 31
A: focus, CSS, 169
Alerts, FTP connection to remote site, 374
Alignment
 adjusting tables for vertical, 240–241
 changing element, 122–123
A:link pseudo-class, hyperlinks, 130, 169

Alphabetical order option, form lists, 358

Alt text

- creating image-based link, 290
- when images won't display, 258–259

Animation. *See* web animation and video

AP-divs, 117–120

Application bar, workspace setup, 4

<article> tag selector, 45, 60, 219, 381

ASP (Active Server Pages)

- installing testing server, 376
- web form mail code, 366
- as web programming language, 363

Attributes

- creating class, 58
- form field, 347–348
- ID, 58, 347

Auto Complete, HTML5 forms, 350

Auto Focus, HTML5 forms, 348

A:visited pseudo-class, hyperlinks, 169

B

Background color

- changing, 116–117, 127–130
- of footer, 132–133
- formatting CSS, 91
- hiding unwanted page
 - components, 181–183
- hyperlink rollovers, 171–172
- of hyperlinks, 172
- of navigation components, 121
- of page, 125–126
- of table cells, 235–236

Background graphics

- adding image to header, 113–117
- creating faux columns with, 175–176
- creating responsive table for mobile device, 414–415
- for faux columns, 175–176
- formatting in CSS, 91
- hiding unwanted images, 181–183
- hiding/replacing for mobile device, 403–404
- with Photoshop Smart Objects, 268–269
- working with, 156–158

Backup, Dreamweaver not offering, 111

Behaviors

- adding to hyperlinks, 310–312, 316
- applying Swap Image, 307–309
- applying Swap Image Restore, 309
- overview of, 301–304
- removing applied, 310
- working with, 305–307

Behaviors panel, 302, 307–310

Berners-Lee, Tim, 30–31

Bit depth, 254

Block elements

- defined, 35
- inline elements vs., 40
- positioning, 88–90
- width of, 78

<blockquote> element, 41, 224

Body rule

- changing font size, 81, 83–84
- creating page layout, 122, 125
- working with CSS, 141, 144

<body> element, HTML

- basic HTML code structure, 31–32, 36, 38
- defined, 35–36
- modifying existing CSS layout, 109
- modifying page width and background color, 124

Borders

- creating text indents, 227–228
- formatting in CSS, 90–91
- styling tables, 233–234, 235
- varying color of, 173–174

Box model, CSS, 56–57

Braces, CSS rule syntax, 60

Bridge, Adobe, 262–265

Brightness and Contrast tool, graphics, 278

Browse For File icon, 287–289

Browse icon, 4, 156–157, 331–333

Browsers

- FLV as fallback video format in older, 333
- most popular, 55, 99
- non-support for most vector formats, 252
- searching for particular URL, 290–291
- statistics for access and usage, 99
- targeting page elements, 294
- template-based pages in, 189
- text formatting on, 62, 70–72

Buggy video, troubleshooting, 333

Bullets, formatting list as, 224

Buttons, form

- defined, 341
- HTML5 workflow and, 346
- radio. *See* radio buttons
- Submit button, 360

C

Cache, remote FTP site setup, 376

Captions

- adding and formatting, 241–242
- adding to images, 259
- creating image placeholder, 131
- working with behaviors, 306

Cascade theory, formatting text in CSS, 61–62

Cascading style sheets. *See* CSS (cascading style sheets)

Cell phones. *See* mobile device design

Cellspacing attribute, styling tables, 233

CERN (Conseil Européen pour la Recherche Nucléaire), 30

Certification, Adobe, 7

Check Links Sitewide, 298

Checkbox Group icon, 352

Checkboxes

- converting radio buttons into, 354
- creating with <input> tag, 345
- inserting group of, 351–353
- radio buttons vs., 353
- tabbing to form fields, 361

Checkboxes Add button, 352

Child pages

- producing template, 192–195
- in templates, 190–192
- updating templates, 195–197

Classes

- creating attributes in CSS, 58
- creating custom, 159–161
- positioning images with CSS, 260–261

Client-based functions, vs. server-side, 293

Clipboard, importing text, 217–218

Cloaking local files/folders, 385

Code hinting

- creating HTML, 38
- creating text indents, 226–227
- dynamic hyperlink effects, 169
- emailing form data, 364

- Code Navigator, 72–74, 75–77, 329
- Code view
 - built-in enhancements of, 10
 - Coding toolbar in, 20
 - overview of, 12
- Coding toolbar, 20
- ColdFusion
 - installing testing server for, 376
 - using RDS to connect to sites, 372
 - web form mail code, 366
 - as web programming language, 363
- Colons (:), generating email forms in PHP, 365
- Color
 - background. *See* background color
 - border, 173–174
 - customizing Accordion behavior, 317–320
 - dynamic hyperlink effects, 168–170
 - raster image quality and, 254–255
 - table, 235–236
 - unlimited depth of JPEG, 256
- Columns, table
 - controlling width, 236–238
 - creating faux, 175–176
 - creating from scratch, 228–231
 - creating responsive mobile device, 414
 - styling cells, 234–236
- Commands, Windows vs. OS X, 5
- Commas (,), multiple selectors, 169–170
- Comma-separated values (CSV) file, 238–240
- Comments, 140
- Commit icon, 267, 275–276
- Compact workspace, selecting layout, 18–19
- Components, page
 - configuring mobile devices, 412–413
 - hiding unwanted, 180–183
 - inserting new, 117–120
- Compression, JPEG/PNG images, 256
- COMPUTED option, Selectors pane of CSS Designer, 25
- Connect to Remote Server icon, putting site online, 384
- Connection, testing FTP remote site, 374–375
- Conseil Européen pour la Recherche Nucléaire (CERN), 30
- Content
 - adding to jQuery Accordion, 313–315
 - completing field, in HTML5, 350
 - creating style sheets for other media, 179–184
 - modifying existing, 126–130
 - producing child pages, 193–194
 - separating from its presentation, 57
 - styling main mobile device, 406–407
 - targeting page, 293–295
 - templates for easily producing site, 189
 - updating templates, 195–197
- Copy and paste
 - creating lists, 221–222
 - Find And Replace command, 246
 - images from Photoshop and Fireworks, 270–273
 - stripping out paragraph returns from text, 216
 - tables from other programs, 232
- Copyright information, modifying footer, 132–133
- Crop icon, 276
- Crop tool, 278
- CSS (cascading style sheets)
 - CSS3 overview, 92–94
 - customizing jQuery Accordion, 315–320
 - disabling in Design view, 110
 - formatting in Design view, 11
 - formatting with, 35
 - modifying existing layout, 108–113
 - review Q & A, 95
 - styling forms, 367–368
 - styling in Code view, 25
 - styling tables, 232–234
 - troubleshooting in Inspect mode, 14
 - troubleshooting with CSS Designer, 24–25
- CSS (cascading style sheets) basics
 - box model, 56–57
 - creating class attributes, 58
 - creating ID attributes, 58
 - defined, 49
 - formatting multiple elements at once, 57
 - formatting objects. *See* object formatting, CSS
 - formatting text. *See* text formatting, CSS
 - HTML defaults, 53–55
 - HTML vs. CSS, 51–52
 - overview of, 48
 - understanding, 50–51
- CSS (cascading style sheets), working with
 - adding visual appeal to menus, 173–174
 - background graphics, 156–158
 - CSS Designer, 139–143
 - customizing classes, 159–161
 - customizing IDs, 161–164
 - descendant selectors, 167–168
 - dynamic hyperlink effects, 168–170
 - faux columns, 175–176
 - final adjustments, 176
 - hyperlink rollover effect, 170–172
 - interactive menu, 165
 - modifying existing hyperlink behavior, 172
 - moving rules to external style sheet, 177–178
 - overview of, 136
 - previewing completed file, 138
 - review Q & A, 185
 - style sheets for other media, 179–184
 - troubleshooting CSS conflicts, 166–167
 - working with type. *See* typeface
- CSS Designer
 - adding background image to header, 113–117
 - adding rules to media query, 400–404
 - adjusting image positions with CSS classes, 260–261
 - changing element alignment, 122–123
 - customizing workspace, 24–25
 - formatting text, 75–77
 - identifying cause of gap in animation, 329
 - managing CSS rules, 139–143
 - setting font-family, 147–150
 - styling forms, 367–368
 - using Edge Web Fonts, 151–153
 - working with Image panel, 261–262
- CSV (comma-separated values) file, 238–240
- Custom Font Stacks tab, Manage Fonts dialog box, 148–150
- Customers, website design for, 98–99
- Customizing
 - classes, 159–161, 224–228
 - IDs, 161–164

- jQuery Accordion, 315–320
- media queries, 397
- workspace. *See* workspace, customizing
- Cut and paste
 - creating external link, 290–291
 - creating lists, 221–223
 - creating semantic structures, 219
 - creating text indents, 225–228
 - importing text, 217–218
 - inserting jQuery Accordion widget, 314
 - setting up email links, 292
 - working with behaviors, 306
- D**
- Declarations, CSS rules, 59, 61
- Define Media Query dialog box, 397–399
- Descendant selectors
 - creating, 167–168
 - styling tables, 232–234
 - understanding, 68–70
- Descendant theory, formatting text in CSS, 66–70
- Design view
 - overview of, 11
 - previewing completed file, 138
 - removing/applying class attributes in, 160
 - toggling CSS styling, 110–111
 - viewing server-side includes, 206
- Desktop Size Resolution Switcher, 407–409
- Dithering, 254
- Div ID pop-up menu, 162
- Docking panels, 18
- Document toolbar, 20, 384
- Dollar sign (\$), PHP variables, 364–365
- Dots per inch (dpi), printers/printing presses, 253
- Dpi (dots per inch), printers/printing presses, 253
- Drag and drop
 - image icon, 259
 - images, 274
 - panels, 17
- Duplicate Set button, 21–22
- .dwt extension (templates), 190
- Dynamic content
 - SSIs working with, 205, 207
 - templates for, 190
- Dynamic hyperlink effects, 168–170

- E**
- Edge Animate, Adobe, 326–330
- Edge Inspect, 418–419
- Edge Web Fonts, 146, 150–153
- Edit icon, 373, 377, 379
- Edit Image Settings tool, 278
- Edit tool, graphics, 278
- Editable regions
 - inserting in templates, 190–192
 - producing child pages, 193–195
 - saving template files without, 189–190
 - updating templates, 197
- Elements
 - applying CSS formatting to multiple, 57
 - changing alignment of, 122–123
 - HTML5 new semantic, 110
- Elements, form
 - adding accessibility, 345–346
 - hidden, 341
 - HTML5, 341
 - web-based, 341–342
- tags, 35, 39
- Email
 - generating form for, 363–366
 - inserting HTML5 form field, 349–350
 - setting up links, 292–293
- <embed> element, buggy video, 333
- Embedded formatting, CSS, 59
- Ems
 - fixed height as, 83
 - fixed widths as, 78
 - as relative measurement, 77
 - specifying widths as, 80–82
- Enter/Return
 - creating HTML in Dreamweaver, 38–39
 - inserting paragraph return, 32
- Entities
 - HTML character, 42
 - as HTML code elements, 33
- Exchange panel, 302–303
- Expand button, cloaking local files/folders, 379
- Expand icon, putting site online, 384
- Expanded workspace
 - adding background image to header, 114
 - selecting layout, 18–19
 - setting up, 4–5
 - using CSS Designer in, 139
- Explicit encryption (FTP over SSL/TLS), 372

- Exs, 77, 78
- Extension Manager CC, 302
- Extensions
 - creating home page, 380–381
 - identifying templates by .dwt, 190
 - inserting server-side includes, 205–206
 - saving server-side includes, 206
- External hyperlinks, 282, 290–291
- Eye icon, 314
- F**
- F5, refreshing SSIs, 209
- Faux columns, CSS, 175–177
- Fields, form
 - adding labels, 349
 - creating field set, 350–351
 - creating with <input> tag in HTML5, 345
 - defined, 341
 - inserting forms on page, 343–344
 - naming conventions, 347
 - setting attributes, 347–348
 - tabbing to, 361
- <fieldset> tag selector, forms, 350–353, 360
- File > Publish command, 327
- File formats
 - raster image, 255–257
 - vector graphic, 252
- File Transfer Protocol (FTP), remote sites, 372–376
- Files
 - cloaking for remote site, 378–380
 - inserting non-web, 265–267
 - previewing completed. *See* previewing completed file
 - saving regularly, 111, 113
- Files panel
 - opening, 107
 - site name appearing in, 4, 105–106
 - in workspace, 10
- Filter panel, 264
- Find And Replace command, 244–247
- Fireworks, 104, 270–273
- Fireworks HTML Option, Insert panel, 261
- Fixed height, object formatting in CSS, 82–84
- Fixed width, 78, 173–176
- Flash. *See* Adobe Flash
- Floating panels, 16, 18

- FLV (Flash video) file format
 - as fallback format for older browsers, 333
 - history of Adobe Flash, 324
 - HTML5 video support for, 332
- Folders, cloaking for remote site, 378–380
- Folders panel, 263
- Font size
 - inserting navigation component, 121
 - sample CSS rule construction, 59
 - setting heights in ems, 84
 - setting widths in ems, 80–82
 - styling tables with CSS, 233
- `` tag, deprecated with CSS, 52
- Font-family
 - building font stacks with web fonts, 154
 - setting, 147–150
 - using Edge Web Fonts, 152–153
 - working with type, 145–146
- Fonts
 - building font stacks with web, 154
 - setting font-family, 147–150
 - specifying size of, 81, 83–84, 155–156
 - typeface vs., 147
 - using Edge Web Fonts, 150–153
 - using web-hosted, 146
 - working with type, 143–147
- Footer
 - adding rules to media query, 400–401
 - creating page design, 102
 - modifying, 132–133
 - modifying existing CSS layout, 109–112
 - working with background graphics, 158
- `<footer>` tag selector
 - adding background image to, 158
 - defined, 44–45
 - styling main content for mobile devices, 406–407
- For attribute, form field labels, 349
- Form icon, 343
- `<form>` tag, 341, 343–344
- Formats
 - adding web video to page, 331–332
 - creating headings, 220
- Formatting
 - applying inline, 35
 - with CSS. *See* CSS (cascading style sheets) basics

- of heading tags, 221
- HTML vs. CSS-based, 51–52
- lists, 221–224
- modifying existing, 126–130
- text with HTML, 34

Forms

- adding tabbing to, 361
- elements of, 342
- emailing data, 363–366
- incorporating text areas, 355–357
- inserting, 343–344
- inserting checkboxes, 351–353
- inserting list menus, 357–359
- inserting radio buttons, 353–355
- inserting text elements, 345–351
- more information on, 368
- overview of, 338, 341–342
- previewing completed file, 340–341
- review Q & A, 369
- specifying action, 362–363
- styling with CSS, 367–368
- submission of, 343, 360

FTP (File Transfer Protocol), remote sites, 372–376

FTP over SSL/TLS (explicit encryption), 372

FTP over SSL/TLS (implicit encryption), 372

G

Get icon, putting site online, 385

GET method, processing form data, 362

Getting started

- about this book, 1
- additional resources, 6
- Adobe certification, 7
- checking for updates, 5
- downloading files, 2
- finding Dreamweaver information, 5
- jumpstarting lessons, 3–4
- prerequisites, 2
- program installation, 2
- recommended lesson order, 2–3
- TinyURLs, 1, 424
- Windows vs. OS X instructions, 5
- workspace setup, 4–5

GIF (graphic interchange format), 255, 265

Graphic interchange format (GIF), 255, 265

Graphics. *See* background graphics

Groups

- checkbox, 351–353
- panel, 16–17
- radio button, 353–355

H

`<h1>` tag selector

- adding rules to media query, 400–401
- adding web animation to page, 327
- creating headings, 219–221

`<h2>` tag selector

- creating headings, 219–221
- inserting jQuery Accordion widget, 313–315
- selecting placeholder text, 217

`<h3>` tag selector

- adding behaviors to hyperlinks, 310
- applying behaviors, 307
- applying Swap Image Restore behavior, 309
- creating headings, 219–221
- customizing Accordion behavior, 316, 318

`<head>` element, HTML

- as basic code structure, 31, 35–36
- CSS formatting vs., 51
- defined, 35–36, 41
- media queries and, 398, 400

Header

- adding background image to, 113–117

- adding rules to media query, 400–404

- creating page design, 102
- inserting editable region in template, 191

- inserting new page design, 117–120

- inserting tables from other sources, 239

- modifying existing CSS layout, 109–112

- styling table cells, 234–236

`<header>` tag selector. *See also* `<th>` tag selector (table header)

- adding rules to media query, 401–404

- creating link destination using ID, 295–296

- defined, 44–45

- inserting navigation component, 120–121

- Headings
 - adding long dash to, 193
 - creating, 219–221
 - formatting text with HTML, 34
 - working with behaviors, 306
 - Height
 - customizing Accordion behavior, 320
 - inserting new components, 118
 - object formatting in CSS, 82–85
 - working with type for line, 144
 - Hidden form elements, 341
 - Hiding unwanted page components, 180–183
 - High-color (16-bit) palette, 255
 - Home page, creating, 380–382
 - HTML (Hypertext Markup Language)
 - common defaults, 53–55
 - Dreamweaver as industry-leading editor, 26
 - as focus of Code view, 12
 - formatting in Design view, 11
 - formatting tables with attributes, 229
 - limitations of formatting
 - commands in, 50
 - in Live view, 13
 - structure of, 31
 - troubleshooting with Live Code, 14
 - HTML (Hypertext Markup Language) basics
 - adding structure, 35–36
 - applying inline formatting, 35
 - creating in Dreamweaver, 37–40
 - formatting text with, 34
 - frequently used HTML 4 codes, 41–42
 - history of, 30
 - inserting tags, 33–34
 - introducing HTML5, 42–46
 - overview of, 28
 - recommended books on
 - HTML 4, 40
 - review Q & A, 47
 - syntax, 32–33
 - understanding, 28–29
 - writing code, 32
 - HTML Property inspector, 23
 - <html> element, HTML, 31–32, 35–36, 41
 - HTML5
 - animation and video support, 324
 - animation on older computers/
 - software and, 330
 - buggy video fix, 333
 - defaults, 54–55
 - email field on forms, 349–350
 - form elements, 341
 - form fields, 345
 - form workflow, 346
 - history of, 42–43
 - HTML 4 layout vs., 112
 - inserting forms, 343–344
 - modifying existing CSS layout, 108–113
 - semantic elements, 110
 - tags, 44–45
 - techniques and technology, 46
 - text fields, 348
 - web animation, 326–329
 - web video, 330–332, 334–336
 - what's new, 43
 - Hyperlinks
 - absolute, 283
 - adding behaviors, 310–312
 - avoiding spaces, 297
 - basic behaviors, 316
 - checking for broken, 298
 - customizing Accordion behavior, 315–320
 - dynamic effects, 168–170
 - editing/removing existing, 286
 - external, 282
 - formatting default behavior, 130
 - home page, 382
 - internal, 282, 286–289
 - modifying existing behavior, 172
 - pseudo-classes and, 169
 - relative, 282
 - removing unneeded styles, 183–184
 - rollover effect, 170–172
 - setting up email, 292–293
 - targeting page elements, 294–297
 - testing interactivity, 138
 - updating, 190, 382–383
 - Hypertext Markup Language. *See* HTML (Hypertext Markup Language)
 - Hypertext preprocessor (PHP), 363–366, 376
- I**
- IDs
 - adding to HTML table, 296–297
 - all radio buttons having same, 353
 - creating attributes, 58
 - creating link destination, 295–296
 - custom, 161–162
 - forms featuring unique, 343
 - identifying image to be swapped, 305
 - inserting form field in Design
 - view, 347
 - targeting link destination, 297
 - targeting page elements, 293–295
 - using once and forgetting, 163–164
 - Illustrator, 104
 - Image Optimization dialog box, 266–267
 - Image option, Insert panel, 261–262
 - Image placeholder
 - adding background image to
 - header, 113–117
 - inserting, 130–131
 - inserting images, 258–259
 - producing child pages for
 - template, 194
 - Image Property inspector, 23
 - Image-based link, creating, 289–290
 - Images. *See also* background graphics
 - copying/pasting from Photoshop/Fireworks, 270–273
 - exercise in, 277–278
 - graphic tools for, 278
 - inserting, 258–260
 - inserting by drag and drop, 274
 - inserting non-web file types, 265–267
 - inserting with Bridge, 262–265
 - optimizing with Property Inspector, 274–277
 - overview of, 250
 - Photoshop Smart Objects, 268–269
 - positioning with CSS classes, 260–261
 - previewing completed file, 257
 - raster graphics, 252–255
 - raster image file formats, 255–257
 - review Q & A, 279
 - setting width in responsive design, 409–411
 - vector graphics, 252
 - working with Insert panel, 261–262
 - Images category icon, 259
 - Images icon, 274
 - element, 260, 410
 - Implicit encryption (FTP over SSL/TLS), 372
 - Import Tabular Data dialog box, 238–239

- Importing text, 216–218
- Indents, text, 224–228
- Inheritance theory, formatting text in
 - CSS, 63–66
- Inline elements
 - block elements vs., 40
 - Code Designer displaying, 75–77
 - Code Navigator not displaying, 73
 - CSS, 59
 - formatting, 35
 - HTML, 35
 - positioning, 88
 - setting image width in responsive design, 410
 - trumping formatting applied by other means, 71
- `<input>` tag, forms, 345, 353
- Insert > Character > Em Dash, 193
- Insert panel, 10, 313
- Inserting
 - forms, 343–344
 - HTML tags, 33–34
 - hyperlinks, 190–192
 - images, 258–260
 - images by drag and drop, 274
 - images with Adobe Bridge, 262–265
 - images with Insert panel, 261–262
 - jQuery Accordion widget, 313–315
 - non-web file types, 265–267
 - server-side includes, 205–207
- Inspect mode, 14
- Interactive menus, 165
- Interactivity
 - with Accordion. *See* JQuery Accordion widgets
 - adding behaviors to hyperlinks, 310–312
 - applying behaviors, 307–309
 - applying Swap Image Restore behavior, 309
 - overview of, 300
 - previewing completed file, 304–305
 - removing applied behaviors, 310
 - review Q & A, 321
 - understanding behaviors, 301–304
 - working with behaviors, 305–307
- Internal hyperlinks
 - creating, 286–289
 - overview of, 282
 - targeting ID-based link destinations, 297

- targeting page elements, 293–295
- IP (Internet Protocol) address, remote FTP site setup, 373

J

- JavaScript code, behavior as predefined, 302
- JPEG (Joint Photographic Experts Group) format, 256, 264, 266
- jQuery Accordion widgets
 - creating external link, 290–291
 - customizing, 315–320
 - inserting, 313–315
 - overview of, 312–313
 - in responsive mobile device design, 415–416
- Jumpstart method
 - facilitating lessons with, 3–4
 - page layout using, 104

K

- Keyboard shortcuts, 21–22

L

- `<label>` tag selector
 - adding text areas to forms, 356
 - adding to form fields, 349
 - inserting form field in Design view, 347
- Layout, selecting workspace, 18–19
- Leading, type, 144
- Lessons (in this book)
 - downloading files, 2
 - jumpstart method of facilitating, 3–4
 - recommended order of, 2–3
 - Windows vs. OS X instructions, 5
 - workspace setup, 4
- `` tag selector, list item, 224
- Library category icon, 198
- Library folders, 379–380
- Library items
 - creating, 198–201
 - overview of, 197
 - server-side includes vs., 203
 - updating, 202–203
- Line numbers, turning on, 140
- Linebreaks
 - inserting checkbox into form, 353
 - making SSIs more readable, 206
 - reordering radio buttons, 354
- Link Checker panel, 298

- Link icon, 171–172
- Linked formatting, CSS, 59
- List Values button, Property inspector, 358
- Lists
 - creating, 221–224
 - as form element, 341, 357–359
- Live Code, 14
- Live view, 13–14
- Local (network) connection, to remote sites, 372
- Local sites
 - in Dreamweaver workflow, 372
 - synchronizing remote and, 386–387
- Local Web Fonts tab, Manage Fonts dialog box, 148
- Local web server
 - establishing remote site on, 377–378
 - installing testing server, 376
 - testing server-side includes on, 206
- Locked icon, 188, 401
- Logos, from vector shapes, 252
- Look and feel of site, 189
- Lossless compression, PNGs, 256
- Lossy compression, JPEGs, 256

M

- Maintain Synchronization
 - Information, remote FTP site, 373, 376–380
- Manage Fonts dialog box, 145–146, 148–149, 152–154
- Manage Sites dialog box, 373, 377–378
- Marcotte, Ethan, 409
- Margins
 - CSS rule syntax, 60
 - formatting in CSS, 85–86
 - inserting new components, 119
 - styling tables with CSS, 233
- Markup, HTML, 31
- Max-width property, relative widths, 79–80
- Media, style sheets for other
 - creating print-media style sheet, 179–180
 - hiding unwanted page components, 180–183
 - overview of, 179
 - removing unneeded styles, 183–184
- @Media pane, CSS Designer, 24, 397, 402–403, 411–412, 415–416

- Media queries
 - adding in or out, 400
 - mobile device design and, 394–397
 - mobile device navigation, 404–406
 - rules for, 400–404
 - smart phone, 397–399
 - style sheets for tables, 416–418
- Media type property, mobile device design, 393–394
- Microsoft Internet Explorer, HTML5 video in, 335
- Minimizing, panels, 15–16
- Min-width property, relative widths, 79–80
- Mobile device design
 - adapting jQuery Accordions to responsive, 415–416
 - adapting page components, 412–413
 - adding CSS navigation, 404–405
 - creating responsive HTML table, 414–415
 - creating style sheets for tablets, 416–418
 - media queries and, 394–397
 - media queries for smart phones, 397–399
 - media queries in or out, 400
 - media query rules, 400–404
 - media type property and, 393–394
 - mobile-ready vs. mobile-optimized sites, 406
 - overview of, 390
 - previewing completed file, 392–393
 - resources for, 419
 - responsive web design, 409–411
 - review Q & A, 420
 - styling main content, 406–407
 - testing media queries, 407–409
 - testing with Edge Inspect, 418–419
- Mobile Size (468 x 800) Resolution Switcher, 396–397, 408–409, 414
- Mobile-optimized sites, vs. mobile-ready, 406
- Mockups
 - modifying existing CSS layout, 108–113
 - page design, 104
- Monochrome (1-bit) palette, 254

- MP4 file format, web video, 331
- Multiple choice, form lists, 359
- Muse, Adobe, 31

N

- Name attribute, form fields, 347
- Named anchors, IDs deprecating, 293–295
- Naming conventions
 - CSS rules, 236
 - editable regions in templates, 191
 - form fields, 347
 - home page, 380–381
 - library item, 198–199
 - new site, 105
 - radio buttons vs. checkboxes, 353–354
 - saving page as template, 189–190
 - saving server-side includes, 206
- Navigation
 - adding CSS to mobile environment, 404–405
 - checking page, 298
 - creating external link for, 290–291
 - creating home page with, 380–382
 - creating image-based link for, 289–290
 - creating internal hyperlinks for, 286–289
 - email link setup for, 292–293
 - hyperlinks and, 282–283
 - inserting component, 120–121
 - modifying existing CSS layout, 111
 - overview of, 280
 - previewing completed file, 283–285
 - review Q & A, 299
 - targeting page elements, 293–297
 - updating links, 382–383
- Network web server, remote site on, 377–378
- Nonbreaking space
 - adding Submit button to form, 360
 - HTML character entity for, 42
 - method for inserting HTML, 33
- Notepad (Windows), writing HTML code, 32

O

- Object formatting, CSS
 - borders and backgrounds, 90–91
 - height, 82–85
 - margins and padding, 85–88
 - overview of, 77
 - positioning, 88–90
 - width, 78–82
- <object> tag, HTML5 fix for buggy video, 333
- Ogg file format, web video, 331–332
- tag selector, selecting entire list, 223–224
- Online resources. *See* resources, online
- OnMouseOut attribute, 309
- OnMouseOver attribute, 308–309
- Ordered lists, 223–224
- OS X instructions, vs. Windows, 5

P

- <p> tag selector
 - adding form lists, 357
 - adding web video, 330–332
 - applying inline formatting, 35
 - creating external links, 290
 - creating headings, 220, 223–224
 - creating stand-alone paragraph with, 41
 - CSS rule syntax and, 60
 - inserting HTML code, 33
 - inserting image placeholder and, 131
 - inserting jQuery Accordion widget, 314–315
 - specifying font size, 155
 - targeting page elements, 294
 - working with behaviors, 306
- Padding
 - adding, 86–88
 - creating hyperlink rollover effect, 171–172
 - CSS rule syntax and, 60
- Page layout
 - adding background image to header, 113–117
 - adding forms to, 343–344
 - creating, 102–103
 - creating design, 102–103
 - creating template from existing, 189–190
 - creating thumbnails, 101–102
 - creating wireframes, 103–104

- Page layout (*continued*)
 - element alignment, 122–123
 - inserting image placeholder, 130–131
 - inserting new components, 117–120
 - inserting placeholder text, 131–132
 - jumpstart method, 104
 - modifying existing content and formatting, 126–130
 - modifying existing CSS, 108–113
 - modifying footer, 132–133
 - navigation component, 120–121
 - overview of, 96
 - page width and background color, 123–126
 - previewing completed file, 107–108
 - responsive web design and, 100
 - review Q & A, 135
 - setting up Dreamweaver site, 105–106
 - tables eschewed for, 229
 - validating webpages, 133–134
 - web design basics, 98–101
 - Welcome screen, 106–107
- Palettes, raster image quality and, 254–255
- Panels
 - dragging, 17
 - floating, 16
 - grouping, 17
 - jQuery Accordion widget, 312
 - minimizing, 15–16
 - overview of, 15
 - personalizing preferences, 21
 - stacking and docking, 18
- Paper forms, online vs., 341
- Paragraphs
 - coding stand-alone. *See* <p> tag selector
 - formatting text with HTML, 34
 - inserting returns, 32–33
- Password, remote FTP site setup, 373
- Path element notation (./), text-based hyperlinks, 287
- Percentage (%)
 - inserting navigation component, 121
 - as relative measurement, 77, 79–80
 - setting fixed height as, 84–85
 - setting fixed width as, 78
 - working with font size, 144
- Period (.), creating email forms in PHP, 365
- Photoshop
 - copying and pasting images from, 270–273
 - creating full mockup in, 104
 - inserting PSD document into web page, 265–267
- Photoshop Smart Objects, 267–269
- PHP (hypertext preprocessor), 363–366, 376
- Pixels, of raster graphics, 253–255
- Pixels per inch (ppi), resolution of raster images, 253
- Placeholder image. *See* image placeholder
- Placeholder text
 - adding to HTML5 email form field, 350
 - creating lists, 221–222
 - inserting, 131–132
 - inserting navigation component, 121
 - inserting new components, 119
 - modifying existing content/formatting, 126–130
 - selecting with <h2> tag, 217
 - specifying font size, 155–156
- PNG (portable network graphic) file format, 256–257, 264
- Point To File icon, 287–288
- Positioning
 - images with CSS classes, 260–261
 - inserting new components, 119–120
 - object formatting in CSS, 88–90
 - vector graphics, 252
- POST method, processing form data, 362
- Poster field
 - adding web video, 332
 - HTML5 animation on older computers/software, 330
- Ppi (pixels per inch), resolution of raster images, 253
- Preferences
 - fixed font size vs. user, 144
 - personalizing, 21
 - turning on to upload dependent files to remote site, 384
 - viewing SSIs in Dreamweaver, 207
- Prelaunch checklist, 383
- Prerequisites, for this book, 2
- Presentation
 - creating style sheets for other media, 179–184
 - separating content from its, 57
- Presets, inserting non-web files, 266
- Previewing completed file
 - creating interactivity, 304–305
 - form design, 340–341
 - mobile device design, 392–393
 - navigation, 283–285
 - page layout, 107–108
 - working with CSS, 138
 - working with images, 257
 - working with templates, 188–189
 - working with text, lists and tables, 214–216
- Print-media style sheet
 - creating, 179–180
 - hiding unwanted page components, 180–183
 - removing unneeded styles, 183–184
- Properties pane, CSS Designer, 24–25
- Property inspector
 - accessing new HTML5 form attributes, 346
 - adding behaviors to hyperlinks, 310–312
 - adding ID to HTML table, 296–297
 - adding lists to forms, 358
 - adding text areas to forms, 356
 - adding web video to page, 331
 - creating image-based link, 289–290
 - creating internal hyperlinks, 286–288
 - creating link destination using ID, 295–296
 - customizing workspace, 23
 - HTML5 video options, 334
 - optimizing images, 274–277
 - setting form field attributes, 347–348
 - setting up email links, 292
 - specifying form action, 362
 - working with behaviors, 306
- PSD files, 265–269
- Pseudo-classes, hyperlink, 169
- Pseudo-selectors, 130
- Publishing to web
 - cloaking folders and files, 378–380
 - completing sidebar content, 383

- creating home page, 380–382
- defining remote site, 372–378
- defining testing server, 376
- overview of, 370–371
- prelaunch checklist, 383
- putting site online, 384–386
- synchronizing local and remote sites, 386–388
- updating links, 382–383
- Put icon, putting site online, 384–385

Q

- Quality setting, inserting non-web file types, 266
- Quick Tag Editor, 130–131, 225–226

R

- Radio Button Group icon, 354
- Radio buttons
 - adding to forms, 353–355
 - creating with `<input>`, 345
 - as form elements, 341
 - new HTML5 form workflow, 346
 - tabbing to form fields, 361
- Raster graphics
 - color, 254–255
 - file formats, 255–257
 - overview of, 252–253
 - resolution, 253–254
 - scaling, 271
 - size, 254
- RDS (Remote Development Services), ColdFusion, 372
- Recovered-file feature, Dreamweaver
 - not offering, 111
- References. *See* resources, book;
- resources, online
- Refresh icon, 13
- Relative hyperlinks, 282
- Relative measurements, CSS
 - adding margins and paddings, 85–88
 - defined, 77
 - height settings, 82–85
 - positioning elements, 88–90
 - setting fixed widths, 78
 - setting relative widths, 79–80
 - width settings, 78–82
- Relative widths, 79
- Remote sites
 - defined, 372
 - installing testing server, 376
 - on local/network web server, 377–378
 - overview of, 372
 - setting up with FTP, 373–376
- Remove CSS Property icon, 171
- Remove Event icon, 310
- Repeat function
 - for background graphics, 157–158
 - creating faux columns, 175–176
- Requirements, installing
 - Dreamweaver CC, 2
- Resample tool, 278
- Reset Button icon, 360
- Reset icon, 267, 275–276
- Resolution
 - JPEGs, PNGs and, 256
 - raster image quality and, 253
- Resources, book
 - CSS formatting and application, 94
 - HTML 4, 40
- Resources, online
 - Adobe Certified programs, 7
 - Adobe Edge Inspect, 418
 - Adobe Edge Web Fonts, 150
 - creating SVG graphics, 252
 - CSS box model, 57
 - font services, 150
 - forms, 368
 - HTML, 40
 - HTML 4 vs. HTML5, 112
 - HTML5, 46
 - HTML5 fields and attributes, 341
 - HTML5 video, 336
 - installing testing server, 376
 - mobile device design, 419
 - program features/tutorials, 6
 - Responsive Web Design* (Marcotte), 409
 - W3Schools, 99
 - web form mail code, 366
 - web programming languages, 363
- Responsive web design
 - adding jQuery Accordions to mobile devices, 415–416
 - creating HTML table for mobile devices, 414–415
 - mobile devices and, 409–411
 - overview of, 100
- Responsive Web Design* (Marcotte), 409
- Rollover effect
 - creating interactive menu, 165
 - using behaviors, 304
- Rollover Image option, Insert panel, 261
- Root directory, remote FTP site
 - setup, 374, 378–380
- Root element, HTML, 35

- Rows, table
 - `<tr>` tag for, 41
 - controlling column width, 237
 - creating tables from scratch, 228–231
 - inserting tables from other sources, 239
 - responsive design for mobile devices, 414–415
 - styling cells, 235–236
- Rules, CSS
 - cascade theory, 61–62
 - Code Designer displaying, 75–77
 - Code Navigator displaying, 72–74
 - CSS Designer managing, 139–143
 - descendant theory, 66–70
 - element alignment, 122–123
 - external style sheets, 177–178
 - gap in animation, 329
 - hiding unwanted page components, 180–183
 - HTML5 video options, 334–336
 - image width in responsive design, 410
 - inheritance theory, 63–66
 - main content for mobile devices, 406–407
 - media queries, 400–404
 - mobile device navigation, 404–406
 - naming conventions, 220, 236
 - navigation components, 120–121
 - new components, 118–120
 - order of, 235
 - ordering of, 220
 - overview of, 59
 - page width/background color, 123–126
 - removing unneeded styles, 183–184
 - smart phones, 399
 - specificity theory, 70–72
 - syntax, 60
 - syntax for, 60
 - troubleshooting conflicts, 166–167

S

- Saving
 - custom workspace, 21
 - files on regular basis, 111, 113
 - moving CSS to external file, 178
 - non-web file types, 266–267
 - page layout as template, 189–190
 - server-side includes, 205–206

- Scalable vector graphic (SVG)
 - formats, 252
 - <section> elements
 - creating lists, 222–223
 - creating semantic structures, 218–219
 - creating text indents, 225–228
 - as HTML5 semantic element, 217
 - inserting tables from other sources, 238
 - Secure File Transfer Protocol (SFTP), for remote sites, 372
 - Select element, 357–359, 361
 - Select Image Source dialog box, 308
 - Selectors
 - commas for multiple, 169
 - CSS rules and, 59
 - descendant, 68–70, 167–168
 - Edge Web Fonts using, 151–153
 - managing rules in CSS Designer, 140–143
 - rule syntax for compound, 60
 - Selectors pane, CSS Designer, 24–25
 - Semantic structures
 - creating, 218–219
 - defined, 44
 - importing text, 217
 - new in HTML5, 110
 - Semicolon (;), CSS rule syntax for declarations, 61
 - Server-side includes. *See* SSIs (server-side includes)
 - SFTP (Secure File Transfer Protocol), for remote sites, 372
 - Sharpen tool, 278
 - Shift/Tab keys, 230
 - Show Log option, Update Pages dialog box, 196
 - Show Set, Properties pane of CSS Designer
 - changing element alignment, 122–123
 - defined, 25
 - modifying background color, 125–126
 - modifying existing formatting, 127
 - overview of, 76
 - setting font-family, 147–150
 - Sidebar
 - adding navigation to mobile devices, 404–406
 - completing content on, 383
 - creating faux columns, 175–176
 - final tweaks to, 176–177
 - inserting editable region in template for, 191
 - producing child pages, 194
 - Site menu, 105
 - Site Setup dialog box
 - cloaking process in, 379–380
 - defining Dreamweaver site, 106
 - jumpstarting lessons, 3–4
 - remote site setup, 373–376
 - Size. *See also* font size
 - copying and pasting images
 - from Fireworks/Photoshop, 270–273
 - optimizing images with Property Inspector, 274–277
 - raster image quality and, 254
 - resizing vector graphics, 252
 - Slash and asterisk (/.**/), comments in CSS, 71
 - Smart Object icon, 268–269
 - Smart phones. *See* mobile device design
 - Sources pane, CSS Designer, 24
 - Specificity theory, formatting text in CSS, 70–72
 - Spell checking webpages, 243–244
 - Split view
 - creating HTML in
 - Dreamweaver, 37
 - HTML vs. CSS-based formatting, 51
 - overview of, 12–13
 - SSIs (server-side includes)
 - creating, 204–205
 - inserting, 205–207
 - requiring testing server, 210
 - updating, 208–210
 - using in large sites, 203–204
 - Stacks, panel, 16–18
 - Staging servers, 372
 - Standard toolbar, 20
 - tag, 35, 39
 - Submission, form
 - adding Submit button, 360
 - overview of, 343
 - preventing empty fields in, 348
 - SVG (scalable vector graphic)
 - formats, 252
 - Swap Image behavior
 - adding to hyperlink, 310–312
 - applying, 307–309
 - applying behavior to hyperlink by removing, 310
 - identifying image using ID, 305
 - Swap Image Restore behavior, 309–310
 - SWF file format
 - buggy video in HTML5 and, 333
 - history of Adobe Flash, 324
 - HTML5 video support for, 332
 - Synchronization, local/remote site, 386–387
 - Syntax
 - CSS rule, 60
 - HTML, 32–33
- ## T
- Tab Index attribute, form elements, 361
 - Tab key, 230
 - Table Property inspector, 23
 - <table> tag selector
 - adding and formatting captions, 241–242
 - adding ID, 296–297
 - creating responsive table for mobile device, 414
 - creating tables from scratch, 231
 - styling tables, 232–234
 - targeting page elements, 294
 - Tables
 - adding and formatting captions, 241–242
 - adjusting vertical alignment, 240–241
 - controlling column width, 236–238
 - copying and pasting, 232
 - creating from scratch, 228–231
 - creating responsive mobile device design, 414–415
 - HTML-based formatting using, 50–51
 - inserting from other sources, 238–240
 - styling cells, 234–236
 - styling with CSS, 232–234
 - working with behaviors, 306
 - Tablet Size (768 x 1024) Resolution Switcher, 396–397, 407–409, 416–418
 - Tablets. *See* mobile device design
 - Tags, HTML
 - applying inline formatting, 35
 - defaults, 53–55
 - formatting text with, 34
 - frequently used, 40–41
 - inserting, 33–34
 - structure of, 31
 - what's new in HTML5, 44–45

- Targeting page elements, navigation, 293–297
- `<td>` tag selector (table data), 234–236, 240–241
- Template category icon, 195–197
- Template folder, 401
- Templates
 - cloaking to prevent upload to remote server, 379–380
 - creating library items, 198–201
 - creating lists, 221
 - creating server-side includes, 204–205
 - from existing layout, 189–190
 - importing text, 213
 - inserting editable regions, 190–192
 - inserting server-side includes, 205–207
 - overview of, 187
 - previewing completed files, 188–189
 - producing child pages, 192–195
 - review Q & A, 211
 - server-side includes vs., 203
 - updating, 195–197
 - updating library items, 202–203
 - updating links, 382–383
 - updating server-side includes, 208–210
 - using server-side includes, 203–204
- Testing
 - Accordion behavior, 320
 - email links, 292
 - FTP connections to remote sites, 374
 - installing server for, 376
 - media query, 407–409
 - media query in mobile device design, 407–409
 - mobile device design, 418
 - server-side includes, 206
- Text
 - creating headings, 219–221
 - creating indents, 224–228
 - creating lists, 221–224
 - creating semantic structures, 218–219
 - Find And Replace command, 244–247
 - importing, 216–218
 - inserting elements into forms, 345–351
 - spell checking webpages, 243–244

- working with behaviors, 306
- working with type. *See* typeface
- Text area, forms, 341, 346, 355–357
- Text Area icon, 355
- Text Field icon, 347
- Text fields
 - defined, 345
 - as form elements, 341
 - HTML5, 346, 348
 - tabbing to form fields, 361
 - using, 345–347
 - working with form field labels, 349
- Text formatting, CSS
 - cascade theory, 61–62
 - Code Navigator, 72–74
 - CSS Designer, 75–77
 - descendant theory, 66–70
 - inheritance theory, 63–66
 - overview of, 59
 - rule syntax, 60
 - specificity theory, 70–72
- Text formatting, HTML, 34
- `<textarea>` tag selector, 41, 356
- TextEdit (Mac), writing HTML code, 32
- `<th>` tag selector (table header)
 - adjusting vertical alignment, 240–241
 - controlling column width, 237–238
 - styling table cells, 234–236
- Third party services
 - fonts, 150
 - web-hosted fonts, 146
- Thumbnails, creating, 101–102
- TinyURLs, 1, 424
- Titles
 - creating editable template, 191–192
 - creating lists, 221
 - creating text indents, 224
 - editable region for, 190
 - HTML code structure for, 31, 36, 39
 - producing child pages, 193
- Toggle Size Constrain icon, 267, 275–276
- Toolbars, 20–21
- Tools, 278
- `<tr>` tag selector, 246
- Traffic patterns, 99
- Triggering element, behaviors, 302
- Troubleshooting
 - CSS conflicts, 166–167
 - CSS styling, 24–25
 - CSS styling in Inspect mode, 14

- CSS styling in Live Code, 13
- FTP remote site connection, 375
- True color (24-bit) palette, 255
- Typeface
 - building font stacks with web fonts, 154
 - fonts vs., 147
 - setting font-family, 147–150
 - specifying font size, 155–156
 - using Edge Web Fonts, 150–153
 - working with, 143–147
- Typekit, 150

U

- `` tag selector, 314
- Undo, 278
- Unordered lists, formatting as bullets, 224
- Update From Original tool, images, 278
- Update Pages dialog box, image-based link, 290
- Update Template Files dialog box, 288–289
- Updates
 - Adobe software, 5
 - internal hyperlink, 288–289
 - library item, 202–203
 - link, 190, 382–383
 - Photoshop Smart Objects for image, 269
 - server-side include, 208–210
 - template, 195–197
- URL (uniform resource locator)
 - establishing remote site on local/network web server, 377
 - processing form data with GET, 362
 - putting site online, 386
 - remote FTP site setup, 373
 - searching for particular, 290
 - TinyURLs, 1, 424
- Usability feature, forms, 348
- Username, remote FTP site setup, 373

V

- Validation, webpage, 133–134
- Values, form
 - adding lists, 358
 - adding Submit button, 360
 - differentiating radio buttons, 353–354
- Variables, generating email forms with PHP, 364–366

- Vector graphics, 252
- Vertical alignment, adjusting tables, 240–241
- Video. *See* web animation and video
- <video> tag selector, 332–336
- View menu
 - displaying desired toolbar, 20
 - splitting screen horizontally, 13
- Views
 - Code view, 12
 - Design view, 11
 - Inspect mode, 14
 - Live Code, 14
 - Live view, 13
 - Split view, 12–13

W

- W3C (World Wide Web Consortium)
 - CSS development, 52
 - HTML5 development, 42–43
 - online resource for, 46
- W3C Validator Notification dialog box, 133–134
- W3Schools, 99
- Web animation and video
 - adapting for mobile devices, 411–412
 - adding web animation to page, 326–330
 - adding web video to page, 330–332
 - buggy video, 333
 - HTML5 video options, 334–336
 - overview of, 322–323

- previewing completed file, 325–326
- review Q & A, 337
- understanding, 324
- Web fonts
 - building font stacks with, 154
 - Edge Web Fonts, 151–153
 - third party, 146, 150
- Web programming languages, 363
- Web video. *See* web animation and video
- WebDav (Web Distributed Authoring and Versioning), 372
- WebM file format, 331
- Web-safe color palette, 254
- Welcome screen, page layout, 106–107
- What You See Is What You Get (WYSIWYG) editor, 9–10
- White space, creating with HTML tags, 33–34
- Width
 - adding mobile device navigation, 404–406
 - controlling column, 236–238
 - creating responsive table for mobile device, 414–415
 - inserting tables from other sources, 240
 - modifying page, 123–126
 - object formatting in CSS, 78–82
 - resizing for mobile device with media queries. *See* media queries
 - styling tables with CSS, 233

- Windows instructions, vs. OS X, 5
- Wireframes, 102, 103–104
- Workflow
 - HTML5 form, 346
 - local vs. remote site in Dreamweaver, 372
- Workgroup environment, templates enabling, 189
- Workspace, customizing
 - keyboard shortcuts, 21–22
 - layout, 18–19
 - overview, 8–9
 - panels, 15–18
 - preferences, 21
 - review Q & A, 26–27
 - saving, 21
 - switching and splitting views, 11–14
 - toolbars, 20
 - touring interface, 10–11
 - using CSS Designer, 24–25
 - using Property inspector, 23
- Workspace, setting up, 4–5
- World Wide Web Consortium (W3C)
 - CSS development, 52
 - HTML5 development, 42–43
 - online resource for, 46
- WYSIWYG (What You See Is What You Get) editor, 9–10

Z

- Z-index property, 119–120