

# **LEARN**

# **Adobe Dreamweaver CC**

## **for Web Authoring**

Adobe Certified Associate Exam Preparation

Kim Cavanaugh  
with Rob Schwartz



# **LEARN**

# **Adobe Dreamweaver CC**

## **for Web Authoring**

Adobe Certified Associate Exam Preparation

Kim Cavanaugh  
with Rob Schwartz



**LEARN ADOBE DREAMWEAVER CC FOR WEB AUTHORING**  
**ADOBE CERTIFIED ASSOCIATE EXAM PREPARATION**  
**Kim Cavanaugh**  
**with Rob Schwartz**

Copyright © 2016 by Peachpit Press

Adobe Press books are published by Peachpit, a division of Pearson Education.  
For the latest on Adobe Press books and videos, go to [www.adobepress.com](http://www.adobepress.com).  
To report errors, please send a note to [errata@peachpit.com](mailto:errata@peachpit.com)

**Adobe Press Editor:** Victor Gavenda  
**Senior Editor, Video:** Karyn Johnson  
**Development Editor:** Robyn G. Thomas  
**Technical Reviewers:** Candyce Mairs and Megan Ahearn  
**Senior Production Editor:** Tracey Croom  
**Copyeditor:** Scout Festa  
**Proofreader:** Elizabeth Welch  
**Compositor:** Kim Scott, Bumpy Design  
**Indexer:** Valerie Haynes Perry  
**Cover and Interior Design:** Mimi Heft  
**Illustration:** Markovka, Fotolia.com

**NOTICE OF RIGHTS**

All rights reserved. No part of this book with video training may be reproduced or transmitted in any form by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. The images and media files provided for download are copyrighted by the authors and Peachpit. You have the non-exclusive right to use these programs and files. You may use them on one computer at a time. Adobe product screenshot(s) reprinted with permission from Adobe Systems Incorporated. For information on getting permission for reprints and excerpts, contact [permissions@peachpit.com](mailto:permissions@peachpit.com).

**NOTICE OF LIABILITY**

The information in this book is distributed on an “As Is” basis, without warranty. While every precaution has been taken in the preparation of the book, neither the authors nor Peachpit shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the instructions contained in this book or by the computer software and hardware products described in it.

**TRADEMARKS**

Adobe, the Adobe logo, Adobe Certified Associate, Dreamweaver, Edge Animate, Edge, Fireworks, Flash, Illustrator, PhoneGap, Photoshop, and Premiere Pro are registered trademarks of Adobe Systems Incorporated in the United States and/or other countries. All other trademarks are the property of their respective owners.

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Peachpit was aware of a trademark claim, the designations appear as requested by the owner of the trademark. All other product names and services identified throughout this book are used in editorial fashion only and for the benefit of such companies with no intention of infringement of the trademark. No such use, or the use of any trade name, is intended to convey endorsement or other affiliation with this book.

ISBN-13: 978-0-13-439642-2

ISBN-10: 0-13-439642-1

9 8 7 6 5 4 3 2 1

Printed and bound in the United States of America

*To mi amor, Silvia.*

*This page intentionally left blank*

# Acknowledgements

Thanks to Rob Schwartz for having the vision and persistence to get this new series of books and videos off the ground.

To Chris Flick, for all the tremendous artwork and the flavor that his cartoons brought to the hands-on projects in this book.

To my daughter, Kathryn Cavanaugh, for her help in checking my work to be sure I was reaching an audience much closer to her age than to mine.

To the great team at Adobe Press, Victor Gavenda, Candyce Mairs, Scout Festa, Tracey Croom, Liz Welch, and especially Robyn Thomas, for keeping this complicated project on time and for keeping the author on task.

To all of the awesome educators in the Adobe Education Leaders program. Your enthusiasm for your craft and dedication to your students continues to inspire me.

*This page intentionally left blank*

## About the Authors

**Kim Cavanaugh** has been teaching and writing about web, graphic, and interactive design for over 15 years. With four books, three interactive online courses, and over two hundred tutorials on Adobe Dreamweaver, Fireworks, WordPress, and others to his credit, Kim's particular skill has always been in opening new opportunities to technology newcomers by making his material easy to follow while remaining technically rich and challenging.

Kim is a founding board member of the Adobe Education Leaders program, is a certified Adobe Education Trainer, and has taught classes on programming and design to students at every level from middle school to college. During the day, Kim is the instructional webmaster at one of the largest school districts in the US.



**Rob Schwartz** (author of book chapters 8 and 9) is an award-winning teacher (currently at Sheridan Technical College in Hollywood, FL) with over 15 years experience in technical education. Rob holds several Adobe Certified Associate certifications, and is also an Adobe Certified Instructor. As an Adobe Education Leader, Rob won the prestigious Impact Award from Adobe, and in 2010 Rob was the first Worldwide winner of the Certiport Adobe Certified Associate Championship. Find out more about Rob at his online curriculum website at [brainbuffet.com](http://brainbuffet.com).





# Contents

Getting Started x

- 1 Welcome to the World Wide Web 3**
  - World Wide What? 4
  - Working in the Web Design World 14
  - PROJECT 1.1 Create Your First Dreamweaver Website 19**
  - PROJECT 1.2 Under the Hood—Examining Code in a Web Page 23**
  - Challenge! Customize the Starter Page and Make It Your Own 32
  - Conclusion 35
  
- 2 Dreamweaver CC Essentials 37**
  - How Does Dreamweaver Help? 38
  - Get Your Files Here 38
  - Define the Chapter 2 Website 39
  - Dreamweaver Workspace Overview 40
  - The Document Window 42
  - The Property Inspector 48
  - Panels and Panel Groups 52
  - Coding Tools in Dreamweaver CC 59
  - Conclusion 61
  
- 3 Building Your First Website 63**
  - Your First Client—Chris the Cartoonist 64
  - Get Your Files Here 65
  - Define the Chapter 3 Website 65
  - PROJECT 3.1 Translating Client Project Requirements 66**
  - PROJECT 3.2 Modifying Page Properties 71**
  - PROJECT 3.3 Setting Font Formats and Links 77**
  - PROJECT 3.4 Adding Content Containers and Attributes 83**
  - PROJECT 3.5 From Blueprint to Assembly: Positioning Page Elements with CSS 89**
  - Challenge! Create Your Own Page Comps 96
  - Conclusion 97
  
- 4 Getting the Big Picture 99**
  - Bring On the Art! 100
  - Get Your Files Here 100
  - Define the Chapter 4 Website 101
  - Image File Formats for the Web 101
  - PROJECT 4.1 Prototyping and Designing with Background Images 103**
  - PROJECT 4.2 Inserting and Editing Images 112**
  - PROJECT 4.3 Floating Images into Position 121**
  - PROJECT 4.4 Creating Structured Layouts with Images 125**
  - PROJECT 4.5 Styling the Header Area 133**
  - Challenge! Create Your Own Design with Photoshop 137
  - Conclusion 137

- 
- 5 Working with Web Page Content 139**
    - Bring on the Content! 140
    - Get Your Files Here 140
    - Define the Chapter 5 Website 140
    - PROJECT 5.1 Styling the Navigation Sidebar 141**
    - PROJECT 5.2 Styling Navigation Elements 147**
    - PROJECT 5.3 Styling Structured Data 155**
    - PROJECT 5.4 Building and Designing Interactive Forms 160**
    - PROJECT 5.5 Finalizing the Prototype for Production 166**
    - Conclusion 171
  
  - 6 Moving from Prototype to Production 173**
    - Let's Go Live! 174
    - Get Your Files Here 174
    - Define the Chapter 6 Website 175
    - Reviewing CSS Syntax 175
    - PROJECT 6.1 Putting the Style Sheet into Production 177**
    - PROJECT 6.2 Building Out the Site with Dreamweaver Templates 183**
    - PROJECT 6.3 Adding Optional Content with Library Items 189**
    - PROJECT 6.4 Publishing and Updating a Dreamweaver Site 195**
    - Conclusion 201
  
  - 7 Exploring the Brave New World of HTML5 and Responsive Design 203**
    - Let's Go Mobile! 204
    - Get Your Files Here 204
    - Define the Chapter 7 Website 205
    - Understanding HTML5, CSS3, and Responsive Design 205
    - PROJECT 7.1 Exploring Responsive Design with Media Queries 208**
    - PROJECT 7.2 Exploring Responsive Design with jQuery and Bootstrap 215**
    - PROJECT 7.3 Inserting HTML5 Video 225**
    - PROJECT 7.4 Exploring CSS3 Animation 229**
    - Your Final Challenge 238
    - Conclusion 238
  
  - ACA Objectives Covered 240**
  
  - Glossary 247**
  
  - Index 261**
  
  - 8 Leveling Up with Design 8-2**  
(included in ebooks and Web Edition\*)
  
  - 9 Working with Outsiders 9-2**  
(included in ebooks and Web Edition\*)

*\*To access your free copy of this book's Web Edition containing more than 10 hours of video, see the instructions on pp. xiv–xv.*

# Getting Started

## Book Learning Objectives

- Learn the inner workings of the World Wide Web and how web content is published and delivered.
- Learn the key terms and common language of web developers.
- Learn to use Adobe Dreamweaver to create web pages and websites.
- Understand and write HTML using Dreamweaver's tools to visualize, write, edit, and test your web pages.
- Use modern web standards to lay out web pages and control their appearance with Cascading Style Sheets (CSS).
- Become proficient with the Dreamweaver interface and the tools used to insert and format text, hyperlinks, images, tables, and multimedia files.
- Use Dreamweaver to design web pages that look and work great on computers, tablets, and smartphones.
- Publish and update web pages to the World Wide Web.
- Apply principles of design that lead to attractive, engaging, and effective user experiences.
- Learn how to work with clients to develop project plans that lead to successful web projects.

Welcome to *Learn Adobe Dreamweaver CC for Web Authoring*! We use a combination of text and video to help you learn the basics of web programming with Adobe Dreamweaver CC along with other skills that you will need to get your first job in web design. Adobe Dreamweaver CC is a powerful program for working with the HTML, CSS, and JavaScript code that make up the modern web experience. Dreamweaver is widely used within the web design industry and by many company web production teams to manage website resources, taking advantage of its dynamic authoring environment to create great user experiences for their customers.

## About This Book and Video

*Learn Adobe Dreamweaver CC for Web Authoring* was created by a team of expert instructors, writers, and editors with years of experience in helping beginning learners get their start with the cool creative tools from Adobe Systems. Our aim is not

only to teach you the basics of the art of web page design with Dreamweaver, but to give you an introduction to the associated skills (like design principles and project management) that you'll need for your first job.

We've built the training around the objectives for the Web Authoring Using Adobe Dreamweaver CC (2015) Adobe Certified Associate (ACA) Exam, and if you master the topics covered in this book and videos, you'll be in good shape to take the exam. But even if certification isn't your goal, you'll still find that this training will give you an excellent foundation for your future work in web design. To that end, we've structured the material in the order that makes most sense for beginning learners (as determined by experienced classroom teachers), rather than following the more arbitrary grouping of topics in the ACA Objectives.

To aid you in your quest, we've created a unique learning system that uses video and text in partnership. You'll experience this partnership in action in the Web Edition, which lives on your Account page at *peachpit.com*. The Web Edition contains 10 hours of video—the heart of the training—embedded in an online ebook that supports the video training and provides background material. The ebook material is also available separately for offline reading as a printed book or as an ebook in a variety of formats. The Web Edition also includes hundreds of interactive review questions you can use to evaluate your progress. Purchase of the book in *any* format entitles you to free access to the Web Edition (instructions for accessing it follow later in this section).

Most chapters provide step-by-step instructions for creating a specific project or learning a specific technique. Many chapters include several optional tasks that let you further explore the features you've already learned. Two valuable chapters acquaint you with other skills and concepts that you'll come to depend on as you use the software in your everyday work. Here is where you'll find coverage of Domains 1 and 2 of the ACA Objectives, which don't specifically relate to features of Dreamweaver but are important components of the complete skill set that the ACA exam seeks to evaluate. Because these chapters aren't part of the path to learning Dreamweaver itself, they aren't included in the printed book. They're included in the ebook editions and in the online Web Edition (to access the free Web Edition, see pages xiv–xv).

Each chapter opens with two lists of objectives. One list lays out the learning objectives: the specific tasks you'll learn in the chapter. The second list shows the ACA exam objectives that are covered in the chapter. A table at the end of the book guides you to coverage of all the exam objectives in the book or video.

## Conventions Used in this Book

This book uses several elements styled in ways to help you as you work through the projects.

 **Video 9.5** About Copyright

Links to videos that cover the topics in depth appear in the margins.

Text that you should enter appears in bold, such as:

In the Link field in the Property inspector, type **http://www.capesnbabes.com**.

Terms that are defined in the glossary appear in bold and in color, such as:

The **web font** that's used in the header of the page is just what the client is looking for. That's a great thing.

 **ACA Objective 2.1**

The ACA objectives covered in the chapters are called out in the margins beside the sections that address them.

Notes give additional information about a topic. The information they contain is not essential to accomplishing a task but provides a more in-depth understanding of the topic:

**NOTE** *The default behavior for background images is to tile across and down the container where they are set.*

Working in Dreamweaver means you'll be working with code. We have used several conventions to make working with the code in this book easier to follow and understand.

In many instructions, you will be required to enter HTML code, CSS rules, and properties and other code-based markup. To distinguish the markup from the instructional text, the entries will be styled with a code font, like this:

Examine the code `<h1>Heading goes here</h1>`.

In instances where you must enter the markup yourself, the entry will be formatted in bold, like this:

Type the following code: **`<h1>Heading goes here</h1>`**.

Within the body of descriptions and exercise instructions, elements may be referenced by name or by class or ID attribute. When an element is identified by its tag name, it will appear as `<h1>` or `h1`. When referenced by its class attribute, the name will appear with a leading period (.) in code font, like this: `.content` or `.sidebar1`. References to elements by their ID attribute will appear with a leading hash (#) and in a code font, like this: `#top`. This practice matches the way these elements appear in the Tag selector interface in Dreamweaver.

## OPERATING SYSTEM DIFFERENCES

In most cases, Dreamweaver CC works the same in both Windows and Mac OS X. Minor differences exist between the two versions, mostly due to platform-specific issues. Most of these are simply differences in keyboard shortcuts, how dialogs are displayed, and how buttons are named. In most cases, screenshots were made in the Windows version of Dreamweaver and may appear somewhat differently from your own screen.

Where specific commands differ, they are noted within the text. Windows commands are listed first, followed by the Mac OS equivalent, such as Ctrl+C/Command+C. In general, the Windows Ctrl key is equivalent to the Command (or Cmd) key in Mac OS, and the Windows Alt key is equivalent to the Option (or Opt) key in Mac OS.

As chapters advance, instructions may be truncated or shortened to save space, with the assumption that you picked up the essential concepts earlier in the chapter. For example, at the beginning of a chapter you may be instructed to “press Ctrl+C/Cmd+C.” Later, you may be told to “copy” text or a code element. These should be considered identical instructions.

If you find you have difficulties in any particular task, review earlier steps or exercises in that chapter. In some cases when an exercise is based on concepts covered earlier, you will be referred back to the specific chapter.

## Installing the Software

Before you begin using *Learn Adobe Dreamweaver CC for Web Authoring*, make sure that your system is set up correctly and that you’ve installed the proper software and hardware. This material is based on the original 2015 release of Adobe Dreamweaver CC (version 16.0) and is designed to cover the objectives of the Adobe Certified Associate Exam for that version of the software.

The Adobe Dreamweaver CC software is not included with this book; it is available only with an Adobe Creative Cloud membership, which you must purchase or which must be supplied by your school or another organization. In addition to Adobe Dreamweaver CC, some lessons in this book have steps that can be performed with Adobe Photoshop and other Adobe applications. You must install these applications from Adobe Creative Cloud onto your computer. Follow the instructions provided at [helpx.adobe.com/creative-cloud/help/download-install-app.html](http://helpx.adobe.com/creative-cloud/help/download-install-app.html).

## ADOBE CREATIVE CLOUD DESKTOP APP

In addition to Adobe Dreamweaver CC, some extension activities suggested in the text require the Adobe Creative Cloud desktop application, which provides a central location for managing dozens of apps and services that are included in a Creative Cloud membership. Although the central lessons in this book and the videos do not require a Creative Cloud subscription, you should explore the ways the Creative Cloud desktop application can be used to sync and share files, manage fonts, access libraries of stock photography and design assets, and showcase and discover creative work in the design community.

The Creative Cloud desktop application is installed automatically when you download your first Creative Cloud product. If you have Adobe Application Manager installed, it auto-updates to the Creative Cloud desktop application.

If the Creative Cloud desktop application is not installed on your computer, you can download it from the Download Creative Cloud page on the Adobe website ([creative.adobe.com/products/creative-cloud](http://creative.adobe.com/products/creative-cloud)) or from the Adobe Creative Cloud desktop apps page ([adobe.com/creativecloud/catalog/desktop.html](http://adobe.com/creativecloud/catalog/desktop.html)). If you are using software on classroom machines, be sure to check with your instructor before making any changes to the installed software or system configuration.

## CHECKING FOR UPDATES

Adobe periodically provides updates to software. You can easily obtain these updates through Creative Cloud. If these updates include new features that affect the content of this training or the objectives of the ACA exam in any way, we will post updated material to [peachpit.com](http://peachpit.com).

## Accessing the Free Web Edition and Lesson Files

Your purchase of this book in any format includes access to the corresponding Web Edition hosted on [peachpit.com](http://peachpit.com). The Web Edition contains the complete text of the book augmented with hours of video and interactive quizzes.

To work through the projects in this product, you will first need to download the lesson files from [peachpit.com](http://peachpit.com). You can download the files for individual lessons or download them all in a single file.

If you purchased an ebook from *peachpit.com* or *adobeypress.com*, the Web Edition and files will automatically appear on your Account page. The Web Edition will be on the Digital Purchases tab, and the files will be on the Registered Products tab (see step 5 below).

If you purchased an ebook from a different vendor or you bought a print book, you must register your purchase on *peachpit.com*:

- 1 Go to [www.peachpit.com/register](http://www.peachpit.com/register).
- 2 Sign in or create a new account.
- 3 Enter the ISBN: **9780134396422**.
- 4 Answer the questions as proof of purchase.
- 5 The Web Edition will appear under the Digital Purchases tab on your Account page. Click the Launch link to access the product.

The lesson files can be accessed through the Registered Products tab on your Account page.

- 6 Click the Access Bonus Content link below the title of your product to proceed to the download page. Click the lesson file links to download them to your computer.

## Additional Resources

*Learn Adobe Dreamweaver CC for Web Authoring* is not meant to replace the documentation that comes with the program or to be a comprehensive reference for every feature. For comprehensive information about program features and tutorials, refer to these resources:

**Adobe Dreamweaver Learn & Support:** [helpx.adobe.com/dreamweaver](http://helpx.adobe.com/dreamweaver) is where you can find and browse Help and Support content on *Adobe.com*. Adobe Dreamweaver Help and Adobe Dreamweaver Support Center are accessible from the Help menu in Dreamweaver. Help is also available as a printable PDF document. Download the document at [helpx.adobe.com/pdf/dreamweaver\\_reference.pdf](http://helpx.adobe.com/pdf/dreamweaver_reference.pdf).

**Adobe Forums:** [forums.adobe.com/community/dreamweaver](http://forums.adobe.com/community/dreamweaver) lets you tap into peer-to-peer discussions, questions, and answers on Adobe products.

**Adobe Dreamweaver CC product home page:** [adobe.com/products/dreamweaver](http://adobe.com/products/dreamweaver) provides information about new features and intuitive ways to create responsive web page layouts that display beautifully on any screen.

**Adobe Add-ons:** *creative.adobe.com/addons* is a central resource for finding tools, services, extensions, code samples, and more to supplement and extend your Adobe products.

**Resources for educators:** *adobe.com/education* and *edex.adobe.com* offer a treasure trove of information for instructors who teach classes on Adobe software at all levels.

## Adobe Certification

The Adobe training and certification programs are designed to help designers and other creative professionals improve and promote their product-proficiency skills. Adobe Certified Associate (ACA) is an industry-recognized credential that demonstrates proficiency in Adobe digital skills. Whether you're just starting out in your career, looking to switch jobs, or interested in preparing for success in the job market, the Adobe Certified Associate program is for you! For more information, visit *edex.adobe.com/aca*.

## Resetting the Preferences to Their Default Settings

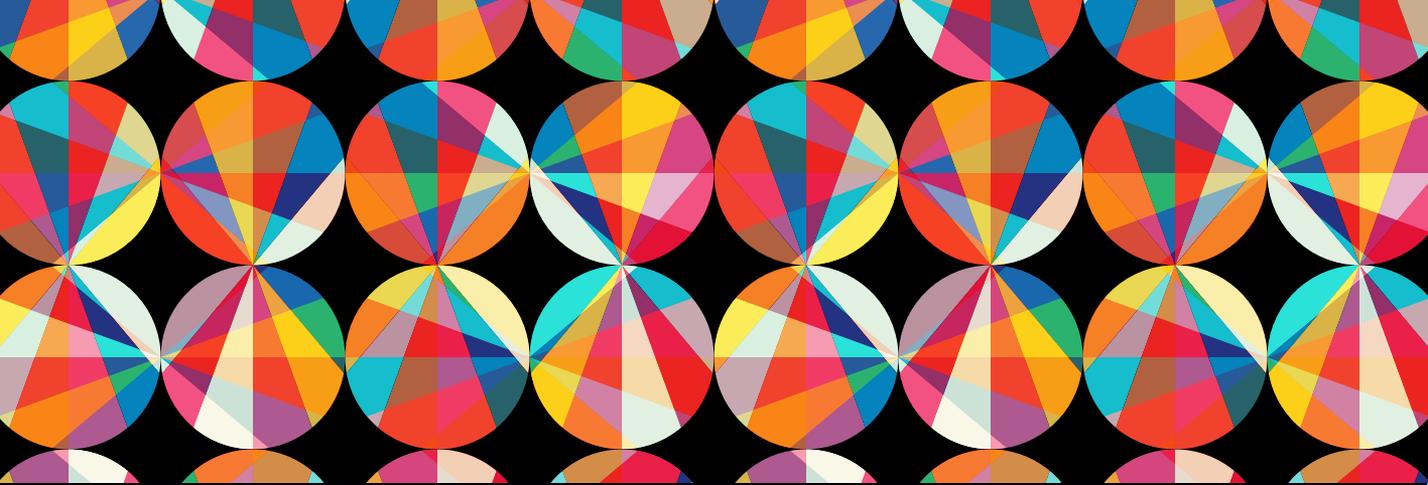
Dreamweaver lets you determine how the program looks and behaves (like tool settings and the default unit of measurement) using the extensive options in Edit > Preferences (Windows) or Dreamweaver CC > Preferences (Mac OS). To ensure that the preferences and default settings of your Adobe Dreamweaver CC program match those used in this book, you can reset your preference settings to their defaults. If you are using software installed on computers in a classroom, don't make any changes to the system configuration without first checking with your instructor.

To reset your preferences to their default settings, follow these steps:

- 1 Quit Adobe Dreamweaver.
- 2 Hold down the key combination Windows key+Alt+Shift (Windows) or Command+Option+Shift (Mac OS), and launch Dreamweaver.
- 3 Continue to hold down the keys until the Reset Preferences dialog appears.
- 4 In the Reset Preferences dialog, click Yes.

The file containing your preferences will be deleted.

*This page intentionally left blank*



## CHAPTER OBJECTIVES

### Chapter Learning Objectives

- Understand basic terminology of the World Wide Web and how web pages are delivered.
- Learn how HTML, Cascading Style Sheets (CSS), media, and other technologies are used for creating modern web pages.
- Understand the basic structure of web pages and how content is contained in HTML tags and the common tags found within a web page.
- Learn how web designers use Dreamweaver to visualize and modify their designs.
- Define a new Dreamweaver website.
- Use Dreamweaver to modify basic elements of a web page such as text, links, images, and page title.

### Chapter ACA Objectives

#### **DOMAIN 2.0**

##### UNDERSTANDING WEB AUTHORING

- 2.1** Demonstrate knowledge of basic web authoring terminology.
- 2.3** Demonstrate knowledge of Hypertext Markup Language (HTML) and the behavior of basic HTML tags.
- 2.4** Demonstrate knowledge of Cascading Style Sheets (CSS) and the behavior of basic CSS tags.

#### **DOMAIN 4.0**

##### WEB AUTHORING USING ADOBE DREAMWEAVER CC

- 4.1** Define a new Dreamweaver site.
- 4.2** Create a web page and define the page layout using HTML, CSS, and div tags.

# CHAPTER 1

## Welcome to the World Wide Web

Chances are pretty good that before you ever got to school or work this morning you were already using the Internet. And why not! We can all send messages to each other using all sorts of different apps and services, check in to see what our friends are up to on our favorite social networking site, play games with people around the world, and use the web to check the weather or traffic, or to find information about how to do almost anything. It's estimated that over 3 *billion* people use the Internet in some way every day.

Behind the scenes, millions of men and women are making it possible for us to stay connected, find information, or be entertained and educated through the magic of the Internet. But of course, it isn't magic at all. The modern Internet experience is built on a vast complex of specialized computers, millions of miles of cables, and a gazillion bits of data flying through the air from cellphone towers and Wi-Fi hotspots.

And now you're about to learn how to *become* one of those people. Looking ahead, you are going to learn how to use Adobe Dreamweaver to design, publish, and maintain web pages and web applications so you can become one of the creators of today's Internet experience through the code you write and the user experiences you design. The skills you'll develop and the tools you'll learn to use are going to put you in the elite company of webmasters, code warriors, and creative professionals who make the Internet what it is today, and what it just might become in the future.

How awesome is that?

# World Wide What?

## ★ ACA Objective 2.1

If you were born after 1994 you've never known a world without the Internet. It was in that year that the very first computer program appeared that allowed the average person to read text *and* view images at the same time. Amazing, right?

The term **browser** was coined to describe a program that allowed one computer to view specially coded documents that could be found on the worldwide *inter-connected network* of computers. The network had already been around for nearly 25 years, but the difference was that instead of using that network just for sharing scientific data and reference materials, these new programs gave the average person access to information and entertainment that was as close as the new computers that were making their way into their homes. Once it became easy for consumers to discover what was possible on this network, the **Internet** became part of our shared culture and our daily lives.

## The Internet of All Things

As you become a web *designer* and not just a web *consumer*, you're going to need to know what's happening behind the scenes when you're checking out the latest viral video or Googling some information. And that means knowing some of the basic terms that describe how the Internet works.

Most people use the terms Internet and **World Wide Web** (aka the web) as if they were the same thing, but in fact there is a difference. The Internet includes *all* the ways that computers can communicate with each other. That includes e-mail, the ability to transfer files from one computer to another, message boards, online games, and file sharing programs. The Internet also allows you to view documents that are written especially for viewing by humans on your desktop computers, tablets, and smartphones. That part of the Internet that lets you view those pages and navigate from one page to another using **hyperlinks** is known as the World Wide Web.

Now before we get too far, you should always keep in mind that computers are fundamentally just machines, and machines are dumb. They do what they are told. Although they can be told to do many, *many* things by way of the programs that are written for them by their human masters, computers require specific instructions for every action they take. Telling a computer to do something involves code of some sort at almost every level. From the moment the computer is switched on to the moment it's actually doing something useful, some sort of code is involved.

And not only is there code, but when it comes to having one computer connect and communicate with another there also has to be an agreement on the language that will be spoken. Without an agreement on the **protocol** that will be used to exchange information, your computer would be just as lost as you would be if you were dropped into an alien world where no one spoke the same language as you. In fact, it was the invention of standard methods for one computer to send and receive information that really made the Internet and the web possible.

**YOUR FIRST STEPS ON AN ALIEN PLANET**



Imagine you've been dropped onto an alien planet (Figure 1.1) and you're all set to meet up with the local population of friendly creatures. How will you communicate with them? Maybe they don't even use a spoken language, and you have to learn to talk with them by combining clicks and whistles and hand gestures—except they have four hands!

The first steps in learning to code are a lot like that.

**Figure 1.1** You have to learn how to communicate before you can begin to share information in a new alien language.

## MAKING CONNECTIONS

You might say that the invention of **Hypertext Transfer Protocol (HTTP)** truly marked the beginning of the web as we know it. You've no doubt seen a web address written like this: `http://google.com`. Back in the olden days you would actually have to *tell* someone to visit a website by sounding it all out to them. "Go to aich tee tee pee colon slash slash google dot com." Seems ridiculous now, right?

But those letters and symbols are really important to the computers on the network. When someone types that into the address bar of their web browser or clicks a link, it's like saying, "Hey, I'd like to see a document that you have on your computer over here on my computer please." Since your web browser is in agreement on the

**Video 1.1** *How the World Wide Web Works*

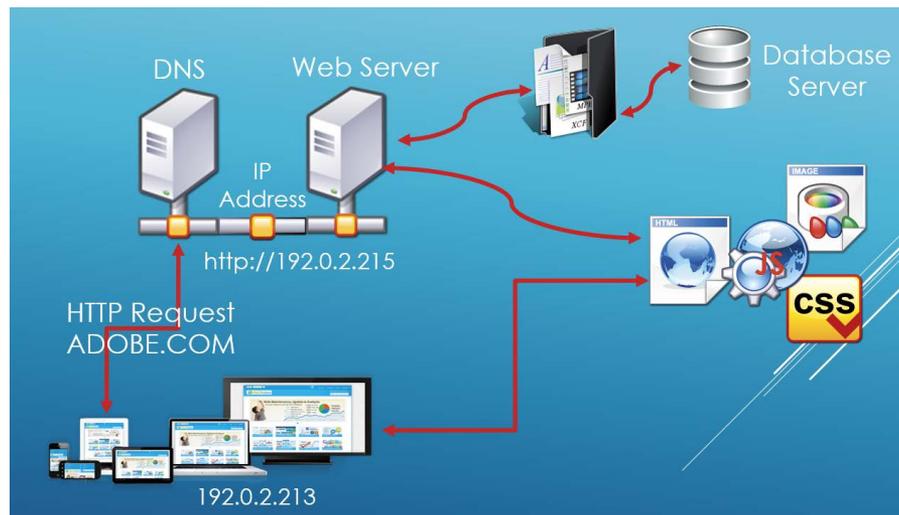
language that will be used for the connection to a remote computer, the two can begin to communicate as your computer (the **client**) receives information from a computer on the network (the **server**).

In order for these handshakes to take place between client and server, the client must identify itself and special devices must know how to route your request to the right server that houses the web page you want to see. The **Internet Protocol (IP)** was established to get that done. Each and every computer on the web is identified with an **IP address**—a combination of numerals that is unique to every single desktop, tablet, or phone that wishes to use the network. An IP address looks like this: 192.0.2.0.

As you can imagine, it would be pretty hard to remember that your favorite website could be found at `http://192.0.2.213`, so a system was created that allowed for different **names** to be assigned to computers that were serving up web pages to clients. This system of **domain names** is what makes it possible for individuals, companies, and government agencies to register a particular name for their exclusive use. And so, when you visit *Adobe.com* or *Microsoft.com* or any other website, the name that is used in your request is translated by a **domain name server**, a special computer on the network that works like a traffic cop, directing the client's request to the correct IP address (**Figure 1.2**).

Whew! And all *that* happens before you ever see anything on your computer screen or phone!

**Figure 1.2** Diagram of connections





## LEVEL UP!

How amazing is the system that connects computers and delivers web pages? Check out this video to see the magic behind the scenes as one web page is delivered to a client in England from a server in Los Angeles.

“There and Back Again: A Packet’s Tale – How Does the Internet Work?”  
[www.youtube.com/watch?v=WwyJGzZmBe8](http://www.youtube.com/watch?v=WwyJGzZmBe8)

## SERVING UP THE GOODIES

Now that you understand the importance of having a common language so computers can speak to each other and make network connections, let’s take a look at the second set of standards that allow web pages to be delivered and displayed.

★ *ACA Objective 2.3*

Learning to write in code is just like learning a foreign language. You have to understand the rules that have been established that let web browsers communicate with servers to display the text and images and links that make up a web page. These rules are known as the **syntax** of the coding language you’re using.

No doubt you’ve heard the term HTML tossed around, and you probably have a basic understanding that it has something to do with web pages. And you’d be right! **Hypertext Markup Language (HTML)** is the basic language that web servers and web browsers use to describe how a web page should be delivered, how the content is structured, and how it should behave.

But HTML is far from the only player needed on the team of technologies that make web pages possible. Although HTML provides the order and structure to a page, the styling of the page—the colors that are used, the position of HTML blocks on the page, the size of text and images, and much more—are controlled by another kind of code, known as **Cascading Style Sheets (CSS)**. Modern web pages use CSS to determine just how the structured elements found in the HTML are displayed by the browser. This combination of structure (HTML) and style (CSS) is what makes it possible for web pages to have almost limitless designs.

The third and final set of code that gets poured into many (if not most) web pages is **JavaScript**. JavaScript is the programming language of the web that adds interactivity to web pages and turns them into more than just a lifeless document. For example, when you click the Like or Share button on a social networking site, it’s

JavaScript that allows your browser to transmit back to the network just how much you love the latest video your friend has shared. Animations, sounds, opening of new windows, and resizing a web page from full screen to phone size are all handled by JavaScript.

Whoa! That's a lot of stuff to know, right? Two different languages *plus* programming? But that's where Dreamweaver is going to come to the rescue. Dreamweaver is a visual editor that allows you to write the code that displays your page with the right styling and interactive elements and provides all the tools you need to create entire websites, full of beautifully styled pages that are responsive to the people who view them.

## TAG. YOU'RE IT

 **Video 1.2**  
Tag: You're It

So. You've cranked up your favorite web browser. Whether you're using Chrome or Safari or Internet Explorer or Firefox—or even opened a web-enabled app on your phone—you know that a transaction is taking place across the web. Your request is being routed to the right web server that contains a web page written in HTML. Now what?

Your browser begins receiving the content of the web page in a very particular way. Web pages are organized using the **Document Object Model**, or DOM. The DOM is just a programmer's way of saying that there should be a standard order to the parts that make up a web page. Web pages need

- A document type
- An area of the page where special instructions for the browser are located but are not displayed on the page
- Content of the page that can be viewed by humans

Each of the areas of a web page is described using HTML tags. **Tags** are the special descriptors that tell the browser how to decipher what's inside them. Tags are enclosed with angle brackets such as `<thetagname>` and are (almost always) written in pairs. The very first tag that a web browser needs to see to display a web page is the HTML tag, written like this:

```
<html> </html>
```

Notice that the second tag has an added forward slash (/) character. This tells the browser that this is the end of the HTML container. Seeing the `</html>` tag tells the browser that it has reached the end of the code for the page.

Now let's see the combination of the DOM and tags in action. Here is all the code a browser needs to display a web page with the text "Hey! How are you doing?"

```
<html>
<head>
<title>How are you doing?</title>
</head>
<body>
<h1>Hey!</h1>
<p>How are you doing?</p>
</body>
</html>
```

## LEVEL UP!

You can make your very own web page using the example HTML code. Just open a text editor, type the code that you see in the example, save the file to your computer with the file extension set to `.html`, and then open the file in your browser. Voilà! Your first web page! I named mine `hey.html` and then pressed CTRL+O (Command+O) in a browser window and navigated to the page to open it.

When you look back at the essential building blocks of every web page, you can see how HTML tags are used to describe them.

This really simple web page includes

- The **document type** defined with the `<html>` `</html>` tags. These tags enclose the entire page.
- The hidden area of the page—the **head**—where instructions to the browser are hidden. The head of the page is enclosed in `<head>` `</head>` tags. Inside the `<head>` tags you see the `<title>` tag with the title of the web page as it would appear in the title bar at the top of the browser. The title won't actually show *on* the page, but it's what you would see in a Google search or if you were to bookmark the page.
- The contents of the page that can be viewed by humans. The visible parts of a web page are enclosed in the `<body>` `</body>` tags.

But what about the tags and content *inside* the body tags? What's with the <h1> and <p> tags?

Did you guess that the <h1> is a heading? And that the <p> tag defines the content of a paragraph? Pat yourself on the back if you did!

HTML is intended to be *pretty* easy to understand with tags that make sense to both the humans who write them and the programs (browsers) that interpret them. And as you begin using Dreamweaver, you'll see how changes you make visually are written into the instructions that the computer will read.

For now, what you should take away from this overview is the understanding of how web pages are structured and coded at the basic level. Now let's see how to make them pretty!



**LEVEL UP: HTML**

The **World Wide Web Consortium (W3C)** is the international authority on web standards and is the organization that maintains all references on how HTML and CSS should be used by web browsers and web designers. You can find an A to Z listing of all HTML tags that are currently in use on the W3C HTML Reference Page at [w3schools.com/tags/default.asp](http://w3schools.com/tags/default.asp).

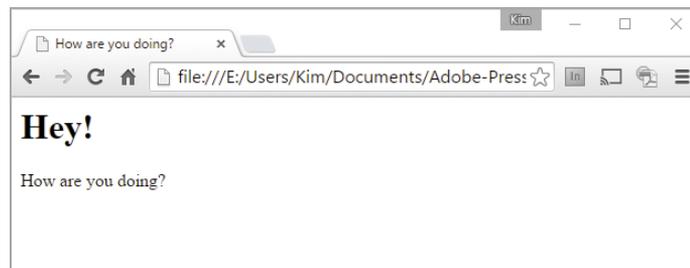
This reference also includes a description of how they are used, whether the tags are currently approved, and those that are no longer recommended.

## TIME TO DRESS THINGS UP

 **Video 1.3** Let's Get Stylish

When you look at **Figure 1.3** (which is the result of the HTML code you saw earlier), you see a web page that has just the very basics in place. A heading and a little text. You've probably not seen a page like that in the real world, and if you did you probably didn't look at it for long!

**Figure 1.3** Simple web pages can be created with just a very little HTML.



Real web pages are made up of many more tags that contain all *kinds* of information. Some of them are pretty easy to figure out. The `<img>` tag, for instance, is for inserting an image. Need a table? The `<table>` tag will see to that. And in the latest version of the HTML specifications, known as **HTML5**, there are even more tags that provide new methods for putting content on the page. Tags like `<section>` and `<article>` and `<summary>` make sense to humans and browsers alike.

Some tags are a little more obscure. For example, when you want to write a hyperlink you enclose it in an `<a>` tag. One of the most common tags you'll learn to use is the `<div>` tag, used to divide the page into different blocks that can be positioned and styled as needed.

HTML tags aren't limited to simply telling a browser that "This is a paragraph" or "This is an image." Good thing! Without the ability to specify more information within a tag, web pages would look just like the boring example you saw in Figure 1.3.

Tags are modified by adding an **attribute**. This is done by including extra instructions right within the angle brackets of the first tag in a pair. For instance, the following code block tells the browser that the text within the paragraph has been assigned a class named `mission`:

```
<p class="mission">Our mission is to provide you with excellent  
→ service. </p>
```

In this case, the `class` attribute allows the page's designer to include styling information using a style that is written to describe how the text should look. More on styling in a minute.

Some attributes are optional, as in the example above, and some are mandatory. The `<a>` tag that defines a link, for instance, isn't much good without the attribute that tells the browser the address of the link. Similarly, an image tag needs the location where the image can be found in order for the browser to get the image and display it in the page. Notice how the attributes are written in the following examples:

```
<a href="http://adobe.com">Visit Adobe!</a>  

```

As you'll learn, attributes are essential to how web pages are designed and styled. And very often they are used in combination with CSS to define all sorts of qualities about the contents of the tag.

## LET'S GET STYLISH

### ★ ACA Objective 2.4

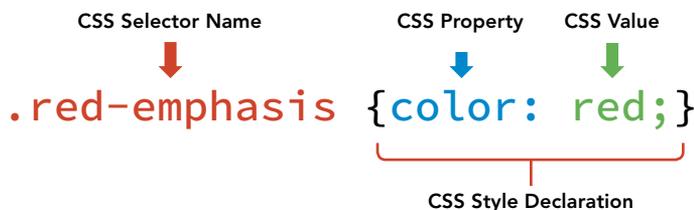
The final elements of modern web design are the styling rules that are written into the HTML for the browser to interpret. Styles are designed using CSS, and once again, there is a particular way that the rules must be written. Remember! Computers are dumb, and they require precise instructions in order to do their jobs. There is a syntax to CSS styling rules just as there is for writing HTML.

Let's look at the example from the simple web page that you saw in Figure 1.3. To style the text in the (only) paragraph on that page, a class attribute might be added that assigns the class named "red-emphasis".

```
<p class="red-emphasis"> How are you doing?</p>
```

To apply a style to the paragraph, a CSS rule is written that alters the appearance of the text. A CSS rule contains a selector and a declaration block, as you see in Figure 1.4.

**Figure 1.4** CSS styling rules follow a standard convention for the name, property, and value used to define the style.



In this example, text with the class `red-emphasis` (the selector) will be colored red and will be bold (the styling declaration).

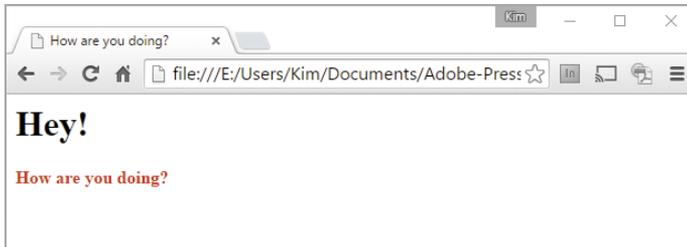
```
.red-emphasis {  
  color: red;  
  font-weight: bold;  
}
```

When you look at what's been added to the HTML, you see that the styling rule has been added to the head, or hidden area of the page.

```
<html>  
<head>  
<title>How are you doing?</title>  
<style type="text/css">  
  .red-emphasis {
```

```
    color: red;
    font-weight: bold;
}
</style>
</head>
<body>
<h1>Hey!</h1>
<p class="red-emphasis">How are you doing?</p>
</body>
</html>
```

Dreamweaver allows you to see both the code and the how the page looks when it's displayed in a browser. As you see in **Figure 1.5**, the addition of the styling rule results in text that is indeed red and bold. Hooray for styles! And hooray for a visual editor like Dreamweaver that lets you work in both worlds at once—the alien world of HTML and the visual world that humans are comfortable in.



**Figure 1.5** CSS styling rules allow the designer to modify the appearance of tags found in the document.

## Once More. With Feeling.

That's a lot of information to take in at one time. But think of the power that you have in your hands. A web designer can write in a language that allows your ideas—expressed in words and colors and images and things that move and make sounds—to be transmitted instantly around the world. And you're well on the way to doing amazing things now that you have these basics under your belt:

- The Internet is the global collection of computers that are joined together using standard methods of communicating known as protocols.
- The World Wide Web is that part of the Internet that allows web pages to be viewed using the hypertext transfer protocol that allows remote computers (servers) to talk with other computers on the network (clients).

- Web browsers are used to connect to the World Wide Web and get the information found on servers to our computers, smartphones, and tablets.
- Web pages are written in a language that browsers understand, known as HTML.
- Browsers are able to display information efficiently because there are standard ways for web pages to be built (the DOM).
- Web pages contain tags—the keywords that wrap around the content of a web page so that the browser knows how and where to display the information.
- Tags can be assigned additional qualities by adding descriptors known as attributes.
- CSS rules are used to provide styling information to the browser so that the information enclosed in tags can be displayed in the way that the designer wishes.
- Coding for the web is powerful and awesome!

## Working in the Web Design World

All this talk of code and different languages can be a bit scary at the start. Really? Web pages are created by typing all these complicated text commands in a whole new language?

Yet that is exactly what the early pioneers of the World Wide Web were doing—creating web pages and entire websites with little more than a text editor, a web browser, and a program that let them copy files from their computer onto a web server somewhere. How very geeky.

But a funny thing happened on the way to providing access to all this information stored on those web servers. As the average, less geeky people of the world began discovering the web, the expectations of what they could *do with* web pages changed. The web became as much about the *experience* as it was about the data, and a whole new industry grew up around the work of designing websites that were attractive, informative, and highly interactive.

## The Rise of the Machines

With all this new work to be done, it wasn't long before some *extremely* geeky people decided that some people had lots of great ideas about how they wanted their

web pages to look and act, but they didn't have a lot of experience in writing code. What they needed was a *visual* editor that would let them create web pages without having to dig too deep into the code behind their designs. One of the earliest web design software programs to really get this right was Adobe Dreamweaver. Released in March 1998 by a company named Macromedia (the company was purchased by Adobe in 2005), Dreamweaver has continued to this day to be one of the most popular What You See Is What You Get (WYSIWYG) web editors. Dreamweaver is used to design and maintain websites around the world by web designers working in environments as varied as one-person web design companies to some of the largest government and corporate sites.

With Dreamweaver, it's possible to view the results of the code that you write into a web page *and* write the HTML and CSS when you use its built-in visual editing tools. Dreamweaver also lets you manage the files and folders that make up a website, transfer those files back and forth between your computer and a web server, and automate the common work that web designers need to do. Dreamweaver provides a much easier design environment for creating, testing, and ultimately publishing a website from the designer's computer to a web server so it can be viewed online.

You've seen a preview of how the code inside a web page determines how things on that document *look*, but (surprise!) there's still a little more you need to know about how websites are put together. Dreamweaver helps you keep all this straight by providing file management tools that work the way that web servers expect, and make it possible for you to keep everything in proper working order.

## THE STRUCTURE OF WEBSITES

Earlier you saw a very simple page in action with nothing but the HTML tags and a few CSS styling rules that modified the way the text on the page was displayed. But a web page that consists of *nothing* but text is a rare thing to find these days. Even a very simple web page is made up of more than an HTML file.

Modern web pages consist of HTML, Cascading Style Sheets stored in separate files, images, JavaScript files that create interaction, and even embedded content that is stored on a web service such as YouTube. A web page is hammered together by the browser by drawing from the HTML file and all the other resources that make up the page. This sort of website is considered a **static website** because it primarily draws from copies of files stored on a web server.

A second kind of website is one that draws from a **database** of information that can be updated on the fly, allowing for live updates. When you post a picture to

★ ACA Objective 2.3

▶ **Video 1.4**  
Understanding  
the Structure of  
Websites

Facebook or Instagram and a friend of yours “likes” it, information is written to a database and then republished to the viewer. This kind of site is considered to be a **dynamic website**.

Dreamweaver is capable of operating in both worlds, and in fact many websites feature a mixture of dynamic and static elements. For instance, the data displayed in a table might be drawn from a database while the styling rules that control its appearance are in a static file stored on the web server. Dreamweaver allows web designers to write the styling rules and view how they will appear even when the source of the data is on a remote file server.

All of this functions the way we expect because the system behind the World Wide Web is predictable to the computers, devices, and browsers we use to access web pages. You’ve already seen how web pages are delivered. Now let’s look at how *websites* are organized in predictable ways. And—you guessed it—there are some rules that web developers need to follow so the required patterns are maintained in your work.

## HOW FILES AND FOLDERS DEFINE A WEBSITE

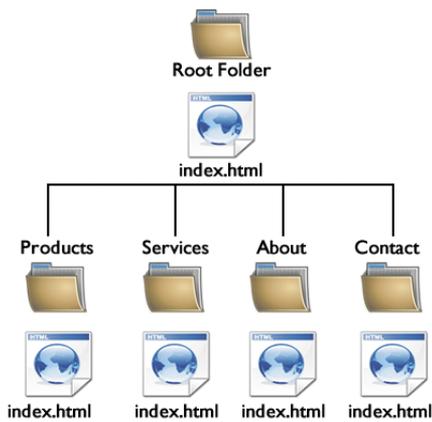
In this book, you’ll primarily be working with a static site—a common development environment where the designer works on the files for a website on their own computer and then publishes them by copying the files to a remote server.

When you visit any web page, you are literally looking at a document that contains a specific filename. Throughout the web, when you land on the home page of a website you are looking at a file named either `index` or `default`. Even though the **file extension** might be different so that some home pages are named `index.htm` while others are `index.html` or `index.asp`, the key predictable pattern is that the name of the web page at the top level of any given domain will be named either `index` or `default`, with `index` being far more common.

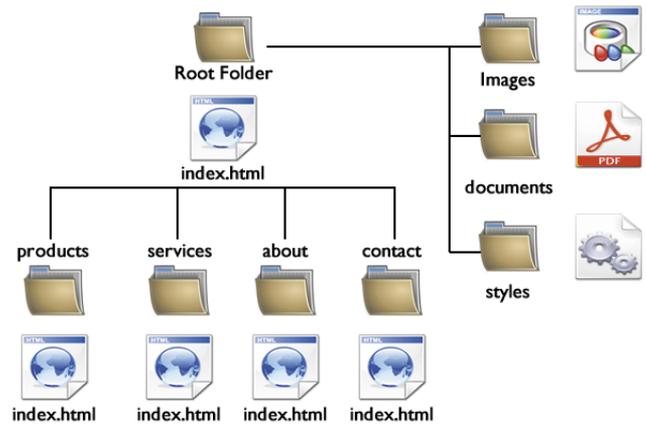
The folder where all the files, subfolders, and resources for the entire site are located is known as the **root folder** for the site. Web servers and Dreamweaver both operate with the understanding that for a given website *all* the files and subfolders for the site will be inside the root folder.

**Figure 1.6** shows how a typical website is structured and how the HTML files within the site are usually named.

Notice how the root folder has a home page and each subfolder also has a home page. There can be many more HTML files within each folder, but this predictable



**Figure 1.6** Websites are organized into folders to allow for easy navigation and the organization of resources.



**Figure 1.7** Special folders within a website are used to store the assets that are used in the documents on the site.

pattern means that when you visit a site that has content in subfolders—such as [www.apple.com/education](http://www.apple.com/education)—what you see is the index page of the subfolder. In this case the name of the folder is `education`.

In addition to segmenting a site into the subjects that the owner of the site wants visitors to see, there are usually additional folders where assets for the site are stored. These files and resources are used to feed content onto the pages within the site and are not viewed by the public. Typical folders for a website might be named `images`, `styles`, `videos`, `scripts`, or other names that define the content stored within them. **Figure 1.7** shows how these folders might be added as a way to organize a site and work more efficiently.

The folder structure of a website also determines another important predictable pattern: the **path** that is taken by the web browser to a given file within the site. Browsers need to know both the *name* of the file and its *location* in order to retrieve and display the document. For example, if you create your résumé and name it `my-resume.pdf` and place it in a folder named `documents`, the path to the file will be written as follows:

`root-folder-name/documents/my-resume.pdf`

You'll find that all of this makes more sense as you dig in and start looking at the code in the web pages you create, as you'll begin doing in **Project 1.1**. But first, here's one more set of rules to follow.

## THE GROUND RULES FOR NAMING FILES AND FOLDERS

You need to be aware of some hard and fast rules before you begin creating files and folders and writing content to your new website. There's nothing too earth-shattering here. Just some simple standards that all web servers and web designers need to follow:

- File and folder names cannot contain spaces.
- In place of spaces, use the two allowed symbols: a dash (-) or underscore (\_). The dash (hyphen) symbol is preferred.
- No symbols other than a dash or underscore may be used in a file or folder name.
- Never start a file or folder name with a symbol.
- Use all lowercase letters when naming files or folders. Some webservers might consider a file named `AboutMe.htm` to be different from one named `aboutme.htm`. It's easier to remove all possibility of confusing the browser by simply sticking to all lowercase.
- Keep your folder and filenames short but understandable by humans.
- Browsers are able to display information efficiently because there are standard ways for web pages to be built (the DOM).

## Let's Review!

We've covered a lot of ground so far in this chapter, and you've learned lots of new terms and been introduced to some new concepts. If your head is spinning just a little bit, not to worry. As you begin working with your own web pages things will become clearer.

Your key takeaway at this point should be that computers and computer networks like the World Wide Web depend on patterns, predictability, and processes that are well-established and agreed upon by all the parties involved. From the methods that allow your desktop computer, tablet, or phone to connect to the web right down to the nitty-gritty of how the code in a web page is named and styled, the work of creating great websites depends on your understanding of what is expected by web browsers and the technologies that make this amazing network possible.

## PROJECT 1.1

# Create Your First Dreamweaver Website

No matter how good the visual editor in Dreamweaver may be, to accomplish the mission of learning to *design* for the web, you still have to know how to *code* for the web.

And just like there's no better way to learn a foreign language than to visit another county, there's no better way to get started with coding than to dive in and start making web pages. Let's do that now with your first hands-on project.

## Get Your Files Here

To jump-start your work in Dreamweaver, a sample website has been created for you so you can examine the HTML and CSS in the page and see for yourself how a simple web page is constructed using tags, attributes, and styling rules, and how the site is organized.

### BRUSH UP ON YOUR SKILLS

This project and others that you'll encounter as you go forward all assume you know how to work with files and folders on your Windows or Macintosh computer. If you need to brush up on these skills, the two video links below cover the basics of working with files and folders on your operating system.

- Windows 8 tutorial: "Organizing Folders and Files" (Lynda.com): [youtu.be/q2ka3149-z4](https://youtu.be/q2ka3149-z4) (Works fine for understanding Windows 10!)
- "How to Use the Finder on a Macintosh" (Lynda.com): [youtu.be/OAwbxOMPD7I](https://youtu.be/OAwbxOMPD7I)

You can watch the two videos below to review how zipped project files can be unzipped.

- "Zipping & Unzipping Files in Windows 8" (Total Training): [youtu.be/HzBF377jwto](https://youtu.be/HzBF377jwto)
- "How to Zip and Unzip Files on a Mac, Quick and Easy" (Digital Byte Computing): [youtu.be/fJygaDO-pa0](https://youtu.be/fJygaDO-pa0)

Download the project files for this lesson, named `chapter-01.zip`, from the Lesson & Update Files tab on your Account page at [www.peachpit.com](http://www.peachpit.com), and store them on your computer in a convenient location.

★ ACA Objective 4.1

▶ **Video 1.5**  
Defining a  
Dreamweaver  
Website

## Define a New Dreamweaver Site

You'll remember from the introduction of this chapter that a **website** is defined as a collection of all the files and folders that are found at a particular domain. You won't actually be publishing any of the work that you do here to a live website that the world can see, but Dreamweaver assumes that you'll *eventually* publish your work, so right from the start your site will be structured just as it would if it were located on a web server. Having a site properly defined is also essential for previewing your work in Dreamweaver's **Live view** environment and in web browsers installed on your computer. So the first step is to create the root folder for your site. This one operation has to be done outside Dreamweaver, using either Windows Explorer or, if you are using a Macintosh, the Finder.

### COPYING PROJECT FILES

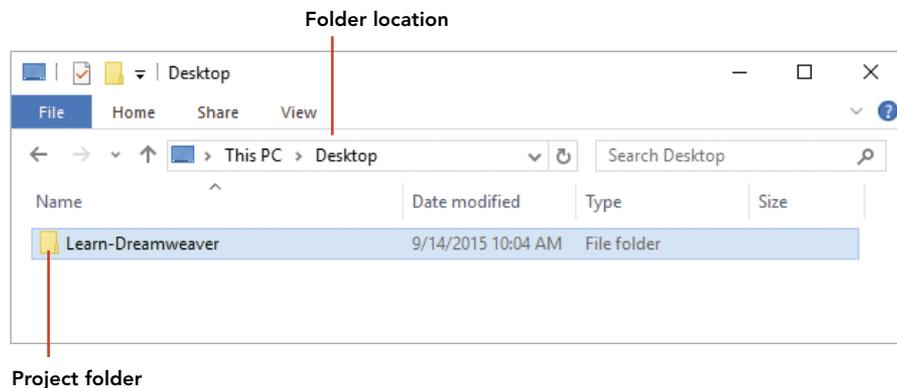
For this book and all the exercises you'll do in upcoming chapters, you'll want to create a master project folder where all your projects will be located. On my computer, I am using the Desktop as the location for a folder named `Learn-Dreamweaver`, as you see in **Figure 1.8**.

- 1 Create a folder on your computer named **Learn-Dreamweaver**.
- 2 Unzip the `chapter-01.zip` files that you downloaded.

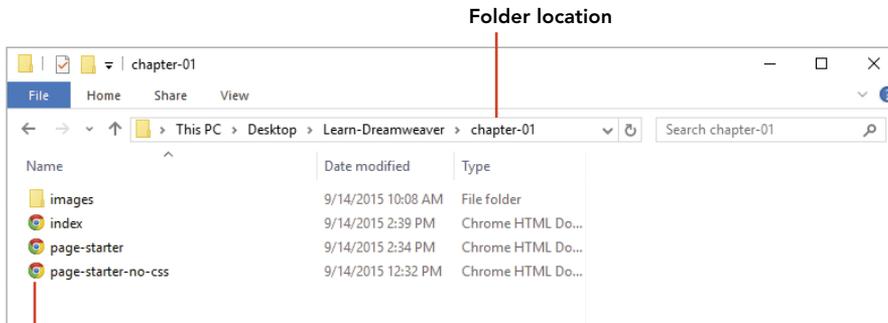
#### TIP

*If you're working in a classroom setting, your instructor may have a specific location where your project files are to be stored.*

**Figure 1.8** Create a project folder on your computer where all your websites for this book will be located.



- 3 Copy the unzipped Chapter 1 project files—folder and all—and paste them into the Learn-Dreamweaver folder. **Figure 1.9** shows how your folder should look once you've copied the resources into place.



**Figure 1.9** Once your project files are in place you're ready to begin working on your first website.

Site files and folder

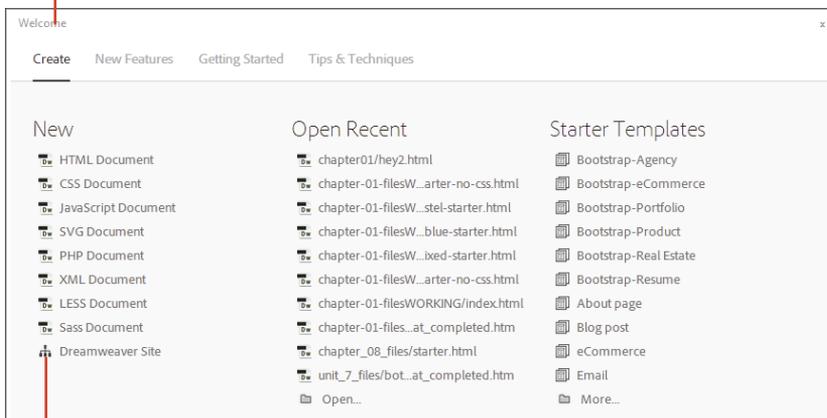
## CREATING A DREAMWEAVER SITE DEFINITION

Now let's get started with Dreamweaver!

When you open Dreamweaver for the first time, you'll encounter the Welcome screen. There are all sorts of goodies tucked away on this screen, but for now you'll get right to work creating the Dreamweaver site.

- 1 Click the Create tab on the Welcome screen (**Figure 1.10**).

Choose the Create tab



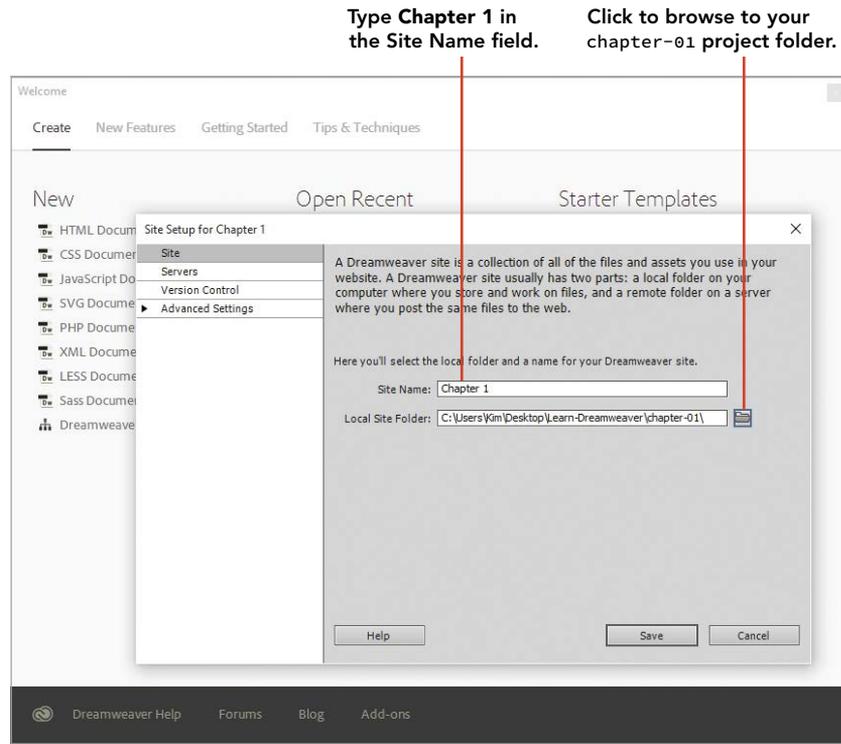
**Figure 1.10** Select the Create tab to start the process of defining a new website.

Select Dreamweaver Site

- 2 Select Dreamweaver Site in the list of options.

When you make this selection, Dreamweaver will open the Site Setup dialog window. The steps you'll need to take to define the site have been labeled for you in **Figure 1.11**.

**Figure 1.11** Defining a simple Dreamweaver web-site only requires that you name the site and point to the location of the site (root) folder.



- 3 Name your site by entering **Chapter 1** in the Site Name field.

The site name is used so you can move between one site and another in Dreamweaver and doesn't need to follow file naming rules. Site names are used only when you are working in Dreamweaver and are never posted to the web.

- 4 Click the folder icon next to the location of the site folder, browse to the **Learn-Dreamweaver** folder, and select the **chapter-01** folder.
- 5 Click the Select Folder button (Choose on the Macintosh).
- 6 Click Save to finish up, and return to Dreamweaver.

## Set Up Your Work Environment

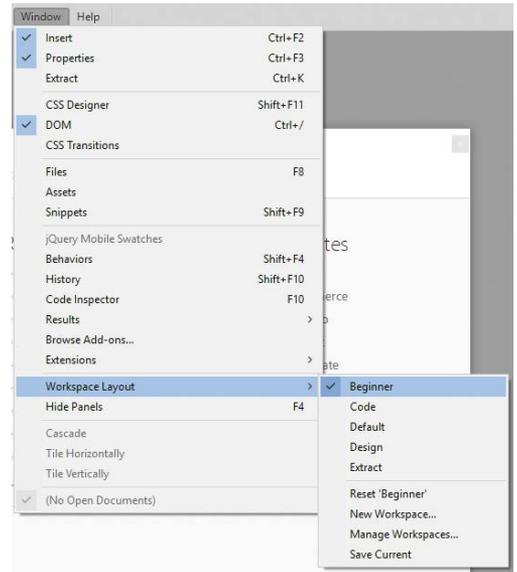
Congratulations! You've defined your first website, and you can now start working with Dreamweaver. So we are all on the same page for this first exercise, let's set the **Workspace Layout** in Dreamweaver so that your view matches what you'll see here.

The Beginner layout is a logical way to start, so change your workspace by selecting Window > Workspace Layout > Beginner, as you see in **Figure 1.12**.

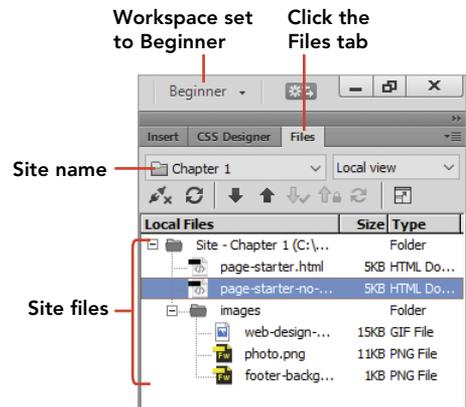
## Let's Review Project 1.1

You'll be learning lots more about how to navigate the Dreamweaver workspace as you move through this book, but for now take a look at the Files panel, located on the right side of the screen as you see in **Figure 1.13**. If the Files panel doesn't display for you, just click the tab labeled **Files** to see the panel. With the Files panel open, you'll see the name of the site you just created and a listing of the files and folders it includes.

Now you're ready to get to work.



**Figure 1.12** The Workspace Layout in Dreamweaver determines how the various panels and inspectors display.



**Figure 1.13** After defining your website, the Files panel appears as you see here.

## PROJECT 1.2

# Under the Hood—Examining Code in a Web Page

★ ACA Objective 2.3

★ ACA Objective 4.2

Congratulations on your new website. Now let's get started on understanding how a web page is created using HTML tags and CSS styling rules.

The goal of this project is for you to look under the hood at how a simple web page is constructed of HTML tags, the styling “hooks” that are created by assigning classes and IDs, and the styling rules that govern how the contents of the page are displayed.

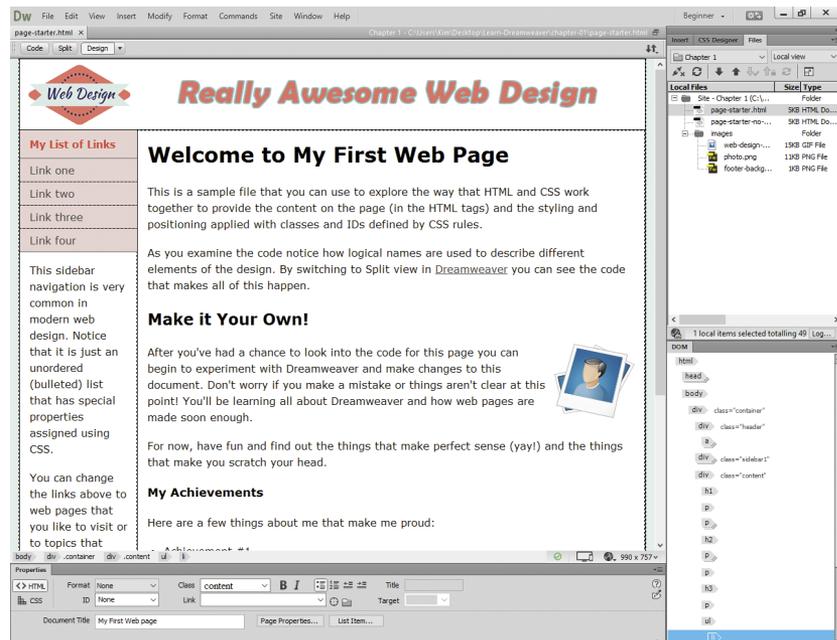
▶ **Video 1.6**  
*Under the Hood—Examining Code in a Web Page*

The downloaded files for this project include two starter pages. We will be looking at one that uses CSS to style the pages.

- 1 Click the Files tab to view the files and folders for this site, then double-click the filename `page-starter.html` to open the file.

Dreamweaver will open your beautiful page in Document view, as you see in **Figure 1.14**.

**Figure 1.14** The `page-starter.html` file opened in Dreamweaver



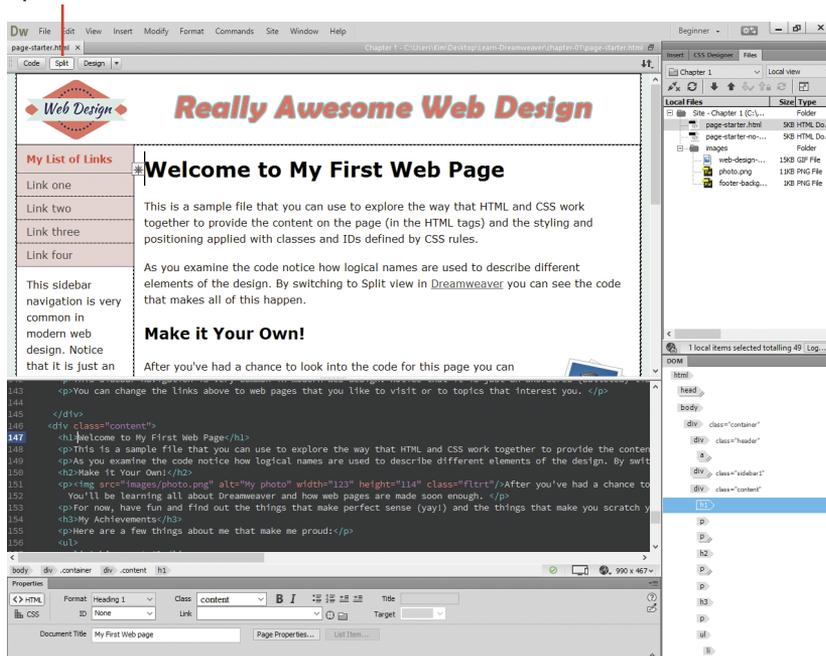
To view the code that creates this page you'll need to change your view of the Dreamweaver workspace so you can see both the code and how that code will be interpreted by a web browser at the same time. Dreamweaver calls this view **Split view mode**, and you can easily switch to that view.

- 2 Click the Split button in the upper-left corner of the Document window.

If you click your mouse in front of the first word in the center of the page (Welcome), you'll see that a cursor appears in both Design view (above) and Code view (below).

It's possible to have this window split in different ways by clicking the View menu at the top of the screen and switching between a horizontal and vertical layout of the Document window. For purposes of this project you should leave your view set to the default, with Design view on top and Code view on the bottom, as you see in **Figure 1.15**.

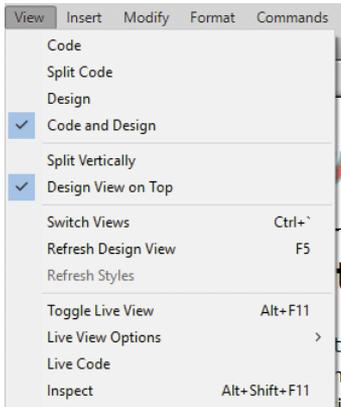
### Split view selected



### TIP

If you wish to compare this document to a version where the CSS has been disabled, open `page-starter-no-css.html`. The CSS rules in this document have been disabled by enclosing the styling rules in a comment so that Dreamweaver will ignore their properties.

**Figure 1.15** The Dreamweaver workspace in Split view



**Figure 1.16** The View menu shows how your workspace is configured. You should see checkmarks next to Code and Design and Design View on Top.

If your view of the workspace doesn't match the one you see in Figure 1.15, you can change the view by selecting View from the menu bar and looking for checkmarks next to the options you see in Figure 1.16.

## Examining the Properties of a Web Page

You'll recall from earlier in this chapter that there are certain patterns to how web pages are constructed. Let's look at each of those principles and see how they are applied in a real web page and how Dreamweaver allows you to access and modify the properties of the page.

- Web pages are written in a language that browsers understand, known as HTML. A file type is declared at the very top of the document in the very first line of code. When you scroll up in the Code window, the first line in the document declares it to be a web page.

```
<!doctype html>
```

- Information that is hidden from the visitors' view of the page but is still critical to how the page displays in the browser is contained within the `<head>` tag. In this document, you see that the `<head>` tag includes the title of the page (My First Web Page) enclosed with the `<title>` tag and the styling rules that begin immediately following the `<style>` tag.

```
<head>
<meta charset="utf-8">
<title>My First Web page</title>
<style type="text/css">
```

- The visible elements of the page are contained within the `<body>` tags.
- Browsers are able to display information efficiently because there are standard ways for web pages to be built using the DOM.

Dreamweaver allows you to view how the content in the page is organized in the DOM panel, located right under the panel group that contains the Files panel. You can navigate within the content of the page by clicking a tag that is listed in the DOM panel, and Dreamweaver will jump to that part of the page in both Design and Code views. In Figure 1.17, you see how the DOM panel was used to select the `<h1>` tag at the top of the **document tree** and how Dreamweaver highlights that tag in both views of the document.

## Tag selected in Design view

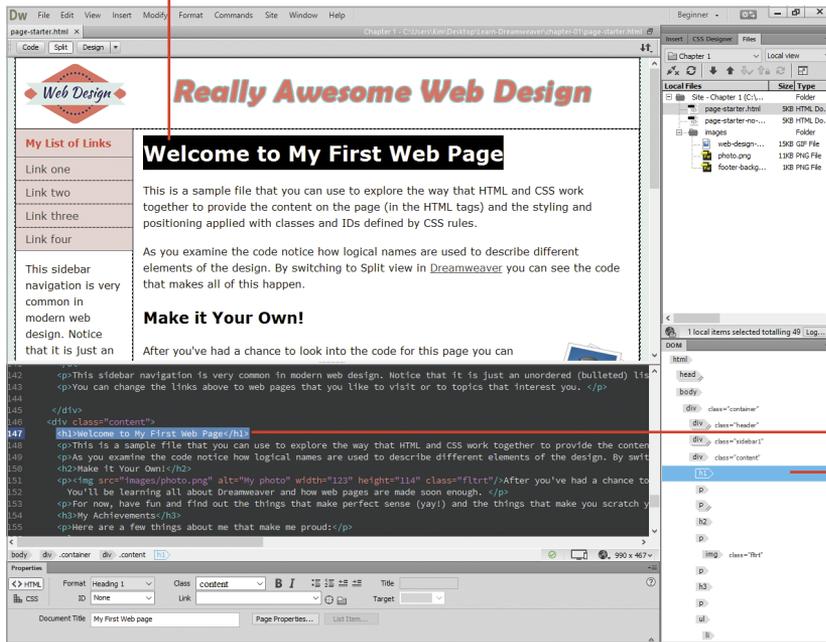


Figure 1.17 The DOM panel displays the structure of the page and allows you to select tags within the document.

- Web pages contain *tags*—the keywords that wrap around the content of a web page so that the browser knows how and where to display the information. In this document you can see instances of the `<div>` tag (used to divide the page into different containers), heading tags, and paragraph tags, as well as unordered lists, image tags, and link tags. These tags contain the content of the page.

Tags can be assigned additional qualities by adding descriptors known as *attributes*. When you look at the opening `<body>` tag, you'll see several tags that have had attributes assigned to them by providing a class. Classes can be assigned multiple times within a document and are the most flexible styling hook that can be applied to a tag. Here you see how classes are applied to the tags that come right after the opening `<body>` tag. You'll also see attributes that have been assigned to the image tag (`<img>`) and hyperlink tag (`<a>`), which assign particular attributes to provide additional information. For the `<img>` tag, those attributes are the width and height of the image measured in

pixels. For the <a> tag, a pound sign (#) has been used to assign a temporary link for that tag.

```
<div class="container">
<div class="header">
<a href="#"></a> </div>
<div class="sidebar1">
```

- CSS rules are used to provide styling information to the browser so that the information enclosed in tags can be displayed the way the designer wishes. Even in a simple web page such as this one, the CSS styling rules can become quite extensive. In this document all the styling information is contained right within the web page itself. This kind of style sheet is called an **internal style sheet**.
- To make the styles in this document easier for you to read, the styles have been organized into four different sections based on how they are used. First, you see the **element styles**, which apply to every instance of a particular tag found within the page. For instance, in this code block you can see how all of the heading tags have been grouped together so that every time the browser encounters tags from <h1> to <h6> the text will be displayed with the top margin set to 0 pixels, and with a bit of padding to the right and left of the content enclosed in the tag. The <p> tag is also included in this group of selectors so that content within paragraphs are given that styling information.

```
h1, h2, h3, h4, h5, h6, p {
margin-top: 0;
padding-right: 15px;
padding-left: 15px;
}
```

- The styling rules that are applied to tags that contain a class attribute are defined in the next block of CSS. Here you see how the container division (<div class="container">) has been styled so that it includes a width, a bit of padding, and a border that extends down the left side of the main content section of the page. **Class selectors** are the most common method for styling content in a web page.

```
.content {
  padding: 10px 0;
  width: 778px;
  float: left;
  border-left: 1px solid #000000;
}
```

- Styling rules can be combined in different ways to make your coding more efficient and specific. For instance, in this document the navigation buttons on the left side of the page are contained in an unordered list using the `<ul>` tag. This rule tells the browser that when it encounters an element with the class `nav` assigned to it *and* that element is enclosed in the `<ul>` tag, the normal bullet should be removed (`list-style: none;`) and border and margin properties assigned as indicated. Notice how the tag name (`<ul>`) is followed by the dot symbol and the name of the class, with no space between the two. This kind of rule combines an element selector with a class selector to provide more specific information.

```
ul.nav {
  list-style: none;
  border-top: 1px solid #666;
  margin-bottom: 15px;
  margin-top: -21px;
}
```

- Styling rules can also feature what are known as **descendent selectors**. With these styling rules, the browser can be directed to treat the content within a tag in a particular way with even more specificity. In this case the browser is being told that where an unordered list with the class of name of `nav` is encountered (`<ul class="nav">`), all the individual items that make up the list (`<li>`) should be styled with a solid border.

```
ul.nav li {
  border-bottom: 1px solid #666;
}
```

- The final set of styling rules in this document is used for floating elements within the page. The image that you see on the right side of the page has a class of `.fltrt` assigned to it, which allows the image to be placed on the

right side of its container so that the text flows around it to the left with a little bit of padding to hold the text away from the image.

```
.fltrt {  
    float: right;  
    margin-left: 8px;  
}
```

- Dreamweaver features code coloring that allows you to see the different kinds of code that exist in your document. Not only does code coloring allow you to clearly see the difference between the different kinds of code in your document, it also helps you see when you've made a mistake. Notice in **Figure 1.18** that the different kinds of code within the page have their own colors assigned.

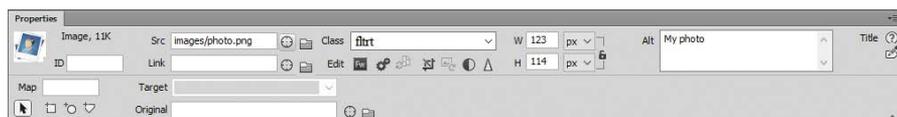
**Figure 1.18**

Dreamweaver uses code coloring to make it easier for designers to view the different kinds of code within a page.

```
113  
114 .fltrt {  
115     float: right;  
116     margin-left: 8px;  
117 }  
118 .fltlft {  
119     float: left;  
120     margin-right: 8px;  
121 }  
122 .clearfloat {  
123     clear: both;  
124     height: 0;  
125     font-size: 1px;  
126     line-height: 0px;  
127 }  
128 </style>  
129 </head>  
130  
131 <body>  
132 <div class="container">  
133 <div class="header"><a href="#">  
135 <p class="linklist">My List of Links</p>  
136 </div class="main">
```

- Dreamweaver gives the designer several ways to modify the properties of elements within the page. The **Property inspector**, located at the bottom of the workspace, provides quick access to the attributes assigned to a selected tag. For instance, when you click the photo embedded in the page, you can see the path to the source of the (src) image along with its filename, the class that has been assigned to the tag, and the **alternate** attribute (alt) of the image. You'll be getting lots of practice in using the Property inspector as you move through the projects in this book, but for now you should know where the Property inspector is located, as shown in **Figure 1.19**.

**Figure 1.19** The Property inspector displays the properties assigned to a selected tag in the page.



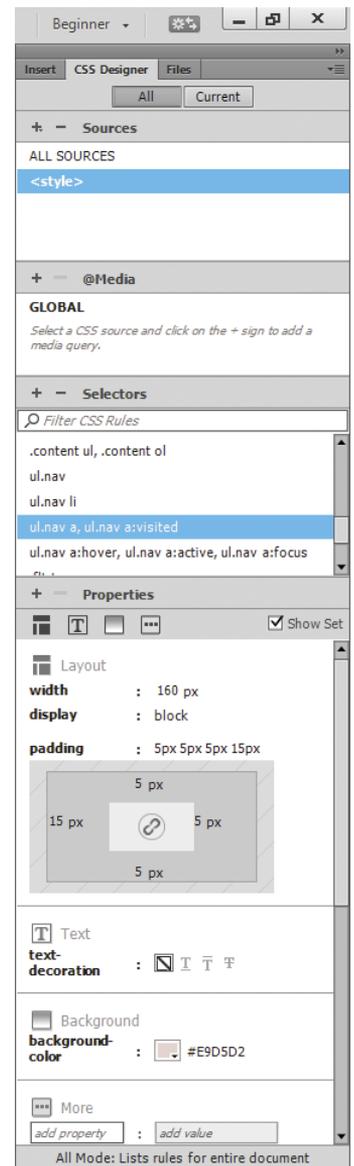
- The **CSS Designer panel** visually displays the styling rules applied to a specific selector. For example, you can see all the rules that define the links in the navigation section of the page by selecting the `ul.nav a`, `ul.nav a:visited` styling rule and then viewing the properties that have been set by selecting the **Show Set** option in the panel and then viewing the styling rules that have been applied for that rule, as you see in **Figure 1.20**.
- You can preview how your page will look in a web browser by switching Dreamweaver to its **Live view** mode. To do this, click the drop-down arrow next to the Live or **Design view** options at the top of the Document window, as you see in **Figure 1.21**.
- You can also preview your design in the default web browser on your computer by pressing the F12 key (Option+F12). The page will open in your browser, and you'll be able to see your design in all its glory.



**Figure 1.21** You can switch to the Live view option in Dreamweaver to see how interactive elements within your design will look when viewed in a web browser.

## Project 1.2 Wrap Up

That's a fair amount of information for you to understand all at once, but remember that the goal here is not for you to understand *everything* that Dreamweaver does, but rather for you to dip your toes into how web pages are made up of tags and CSS styling rules, and how Dreamweaver's workspace displays this information to the web designer. In the Challenge portion of this chapter, you'll have a chance to experiment with Dreamweaver as you take this starter page and make it your own creation.



**Figure 1.20** The CSS Designer panel allows you to see the styling rules applied to an element.

# Challenge! Customize the Starter Page and Make It Your Own

- ★ ACA Objective 2.3
- ★ ACA Objective 4.2

Now that you've worked in Dreamweaver just a little bit, you can take things to the next level by using your own content and making the starter page your very own. Here are the objectives for this challenge and some tips on getting things done.

**Video 1.7**  
Challenge!  
Customize the  
Starter Page and  
Make It Your Own

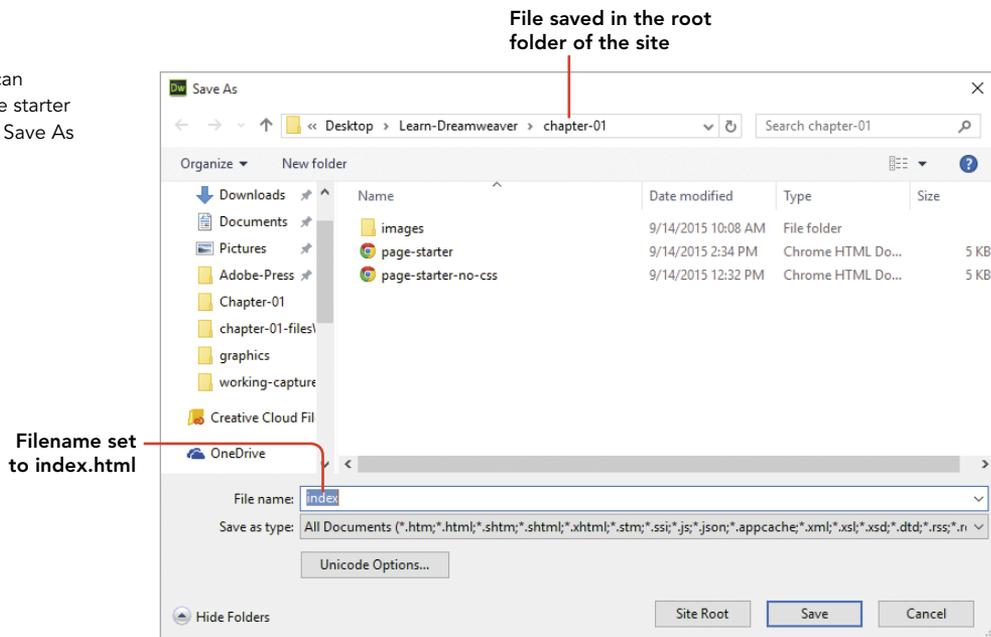
## GOAL 1

### Change the Name of the File and Create a Home Page for Your Site

Before you can start making your own page, you'll want to give the starter page a new filename and make sure it is located within the website you just defined. In these first few steps you'll do just that.

- 1 Make a copy of the `page-stater.html` file by opening that page (if it's not already open) and choosing **File > Save As**.
- 2 In the Save As dialog box, name the file **index.html**, as you see in **Figure 1.22**, and be certain it's saved within the root folder for this site: `chapter-01`.

**Figure 1.22** You can make a copy of the starter page by using the Save As dialog.



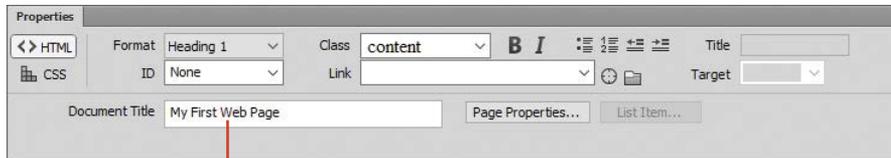
- 3 Click the Save button to save the file with this new name.

You now have a home page for your site!

## GOAL 2

# Change the Title of the Page

Use the Property inspector to change the name of the page so that it uses your name along with some awesome new name for the page. For example, you might name the page **Jonathan's Amazing First Web Page**. (Assuming your name is Jonathan!) **Figure 1.23** shows the location on the Property inspector where you can change the document title.



Change the document title here.

**Figure 1.23** You can change the title of a document directly in the Property inspector.

## GOAL 3

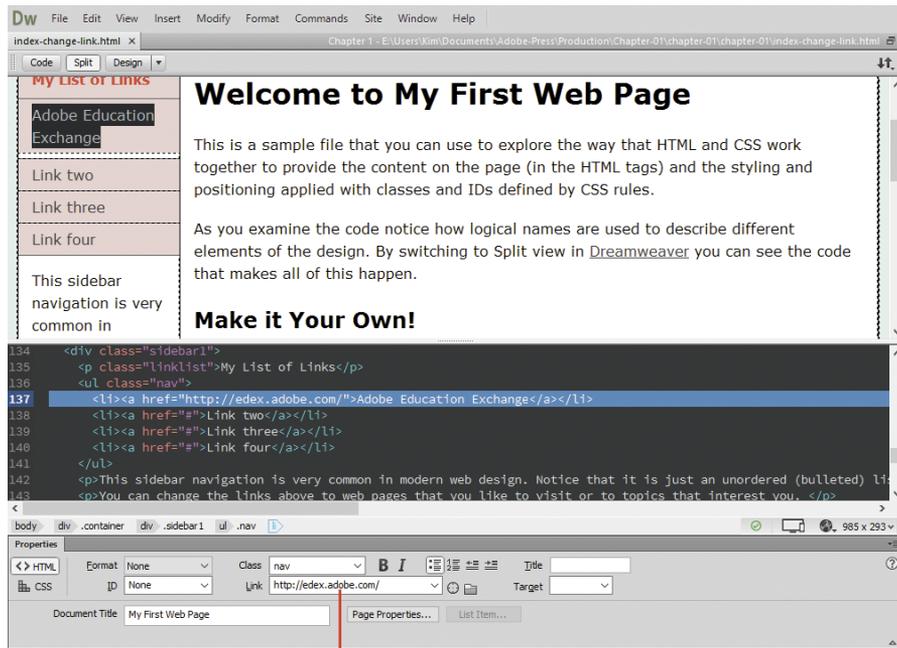
# Change the Navigation Links

The navigation area on the left side of the page contains temporary dummy links that let you see how the design looks when you place your mouse over the links. Use the following steps to change these links.

- 1 Click inside each button, and change the name of the link to the name of a website you enjoy visiting.
- 2 Visit the page, copy the URL from the address bar of your browser, and paste the link into the Link field of the Property inspector, as you see in **Figure 1.24**.

In this example, the name of the link has been changed to **Adobe Education Exchange** and the URL has been inserted into the Property inspector. Note that the full URL—including the protocol—must be included when you link to a web page. For this example the entire correct link is **http://edex.adobe.com/**. But of course you can link to a page of your own choosing.

**Figure 1.24** Select the link you wish to change, and paste the web address (URL) into the Link field of the Property inspector.



Set the link value here.

## GOAL 4

# Write Your Own Story

The body of this page has text that won't have much meaning after you start working on the document. But now it's time to tell your own story. Change the text to tell the world (or at least anyone who sees this page on your computer) a little more about yourself.

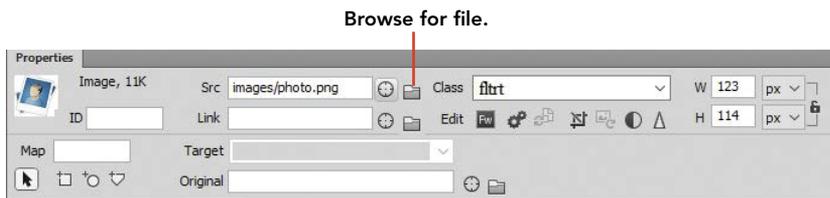
You can use the My Achievements section to list some of the things you've done that make you proud. Then scroll down a bit and modify the list of goals that you've set for yourself.

## ADVANCED GOAL

# Replace the Images on the Page

If you have images of your own that you'd like to use in this design you can certainly do so, but you'll need to do a few things first so that the new images will work with this design.

- 1 Check the size of the images in this design so that the new images you create are the same size as the ones you see in the starter pages. If you want to replace the header image, it will need to be exactly **960** pixels wide by **110** pixels high. The placeholder photo that is embedded in the page is currently set to be 123 pixels wide by 114 pixels high. In the case of this image, you can use a photo that is a little bigger if you'd like, but make sure it's not too big or the page might blow up.
- 2 Copy any image files you want to use in your design into the root folder for this site (**chapter-01**), and place them inside the **images** subfolder.
- 3 Once you have the files in place, click the Browse button next to the image source (SRC) field in the Property inspector to locate your image, and replace the image that's been included with the starter page. **Figure 1.25** shows where to locate the Browse for File button on the Property inspector.



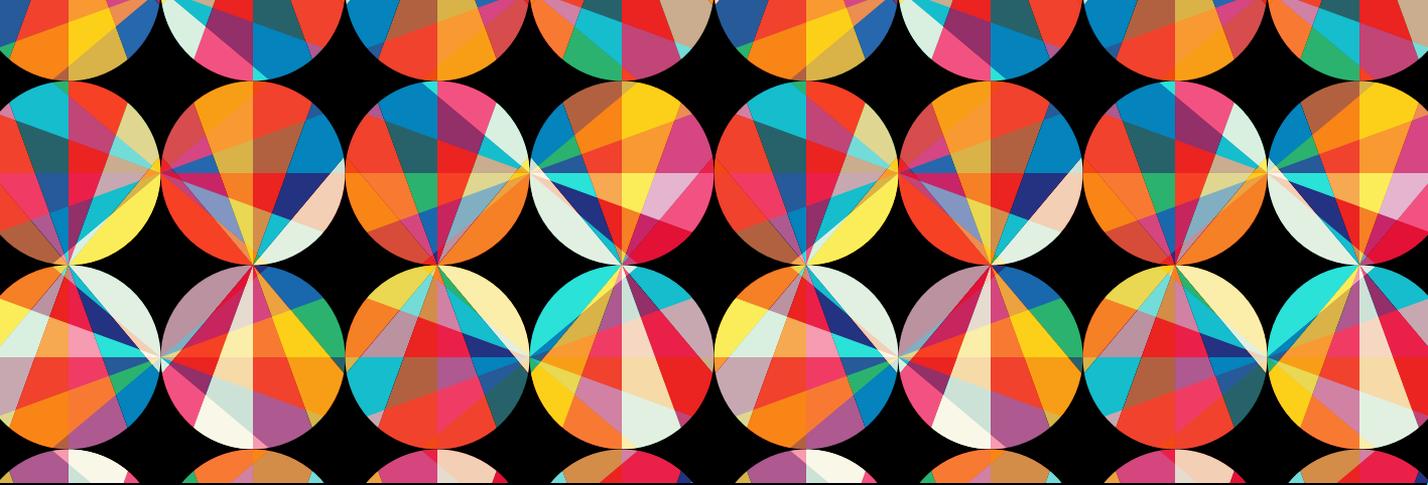
**Figure 1.25** Use the Browse for File button to locate the image you saved into your project folder.

## Conclusion

In this chapter, you've learned about the standard language of the Internet and how the communication system known as the World Wide Web functions using a system of connected computers to deliver information to people all over our planet. You've seen that this system depends on the predictable patterns found in the coding languages of HTML and CSS that are used to create web pages.

You've also taken a short journey into the inner workings of Dreamweaver by defining your first website and examining how the software displays the code that underlies a web page while simultaneously showing you how the page will look when it is rendered by a browser.

Along the way you've been introduced to the terms and technologies that web designers need to understand so they can be successful in their work. There's still much to learn, but as you wrap up this introduction, you should feel confident that you'll soon make your mark on the web by using Dreamweaver to design and code attractive, functional, and interactive web pages of your very own.



# CHAPTER OBJECTIVES

## Chapter Learning Objectives

- Identify the primary elements of the Dreamweaver CC user interface and understand how they are used.
- Learn the different ways that you can display web pages in the Document window.
- Define the functions of the panels contained in Dreamweaver CC and how they are used.
- Understand the functions of the Property inspector for working with content in a web page.
- Learn how the Files and Favorites panels are used to display files and resources located within a website.
- Employ different methods for displaying styling information using the CSS Designer panel.
- Learn the location and purpose of Dreamweaver interface elements in preparation for the ACA exam.

## Chapter ACA Objectives

### **DOMAIN 2.0**

#### UNDERSTANDING WEB AUTHORING

- 2.4** Demonstrate knowledge of Cascading Style Sheets (CSS) and the behavior of basic CSS tags.

### **DOMAIN 3.0**

#### UNDERSTANDING ADOBE DREAMWEAVER CC

- 3.1** Identify elements of the Adobe Dreamweaver CC interface and demonstrate knowledge of their functions.
- 3.2** Identify and define the functions of commonly used panels, including the Insert panel, Property Inspector, Assets panel, and Files panel.
- 3.3** Navigate, organize, and customize the workspace.

### **DOMAIN 4.0**

#### WEB AUTHORING USING ADOBE DREAMWEAVER CC

- 4.1** Define a new Dreamweaver site.

### **DOMAIN 5.0**

#### TESTING, PUBLISHING, AND MAINTAINING WEB CONTENT USING ADOBE DREAMWEAVER CC

- 5.2** Manage assets, links, and files for a site.

## CHAPTER 2

# Dreamweaver CC Essentials

Imagine that your word processing program—say, Microsoft Word—had to display a document in all sorts of different sizes. Of course there's the standard 8.5" x 11" format, which is used most often, but what if Word also had to generate a view of the page so that the document could be printed on 3" x 5" and 5" x 7" index cards? Your work in that program would probably be way more complicated than it is now and would look a whole lot different.

But that is what a web design program like Adobe Dreamweaver has to be able to do, especially now that everyone wants to view content on the web on their smartphones, tablets, and desktop computers. The complexity of web pages and web design is the reason that Dreamweaver doesn't look like other computer programs you've used. Although there are common design elements in Dreamweaver that you'll see in other Adobe programs, such as Photoshop, Dreamweaver can be a bit intimidating when you first open it and see how the user interface is arranged.

Not to worry! Dreamweaver is a full-featured program chock full of all the tools that a web designer needs to create and publish to the web.

In this chapter, you'll learn about the essential elements of the Dreamweaver working environment. The goal is for you to get past that uncomfortable feeling you get when you open a computer program for the first time—especially one as different as Dreamweaver.

# How Does Dreamweaver Help?

Adobe created Dreamweaver with the ability to allow a designer to work the way that is most efficient for *that* designer. It is highly customizable. Among its unique features, Dreamweaver does the following:

- Allows you, as a web designer, to work in a completely visual environment where you work directly on the page much as you would in a word processing program
- Lets you simultaneously see the code behind the page and work on both the code and the page at once
- Gives you a live design environment that shows you how your page will appear in a modern web browser
- Provides tools that let you display your page at different sizes so you can see how the design will appear on iPhones and iPads, Android phones, and tablets, as well as on full-size monitors on a laptop or desktop computer
- Provides visualization tools for writing, organizing, and managing the CSS styling rules that determine how your design looks
- Allows code warriors to work entirely in the HTML code that makes up their web pages and includes all sorts of tools to assist in visualizing HTML and CSS, as well as tools that help you write code more quickly and efficiently
- Provides a resource management panel so you can track and organize the assets within your site, including images, links, colors, and more
- Includes a file management program that lets you publish to a live web server and make sure your files are correctly synchronized

Accomplishing all that is a pretty tall order, but Dreamweaver has been doing all this and more for a long time. Along the way, Adobe updated and tweaked Dreamweaver as new technologies and methods became prominent across the web and web designers needed tools that let them accomplish the very cool work of designing for an interactive environment.

## Get Your Files Here

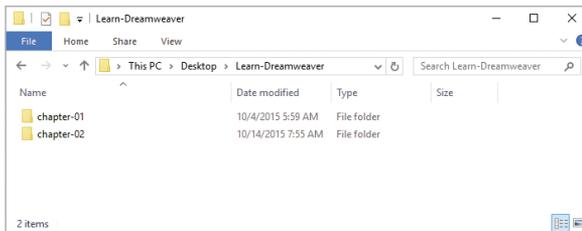
To follow along with your tour of the Dreamweaver workspace, I've created a sample website for you so you can see how different files and resources are displayed in

Dreamweaver. Your first task is to copy the folder that contains these files to your computer and define a website in Dreamweaver.

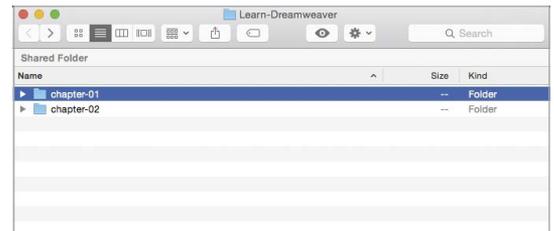
- 1 Download the project files for this lesson, named **chapter-02.zip**, from the Lesson & Update Files tab on your Account page at [www.peachpit.com](http://www.peachpit.com), and store them on your computer in a convenient location.
- 2 Unzip the files, and copy the **chapter-02** folder to the **Learn-Dreamweaver** folder you created in Chapter 1.

The **Learn-Dreamweaver** folder where you have saved your files should appear as you see in **Figure 2.1**.

**Figure 2.1** Copy the **chapter-02** folder to the **Learn-Dreamweaver** project folder you created in Chapter 1.



**Windows**



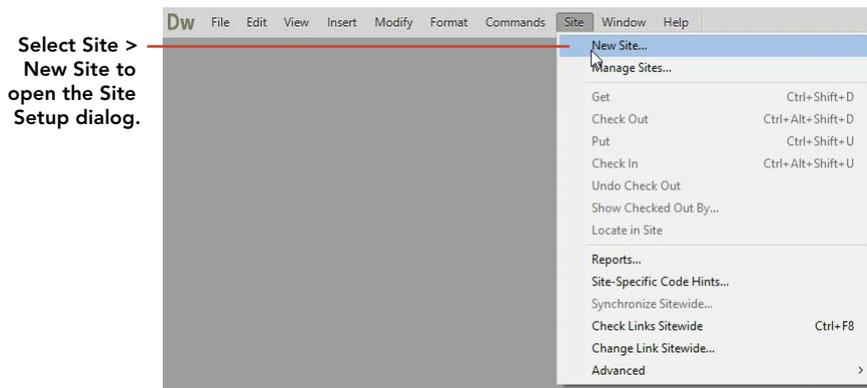
**Mac**

## Define the Chapter 2 Website

You need a website to experiment with. Use the following steps to define the Chapter 2 website.

★ *ACA Objective 4.1*

- 1 From the application bar at the top of Dreamweaver, select **Site > New Site** as you see in **Figure 2.2**.



**Figure 2.2** Use the **Site** menu in the application bar to open the **Site Setup** dialog window.

### **NOTE**

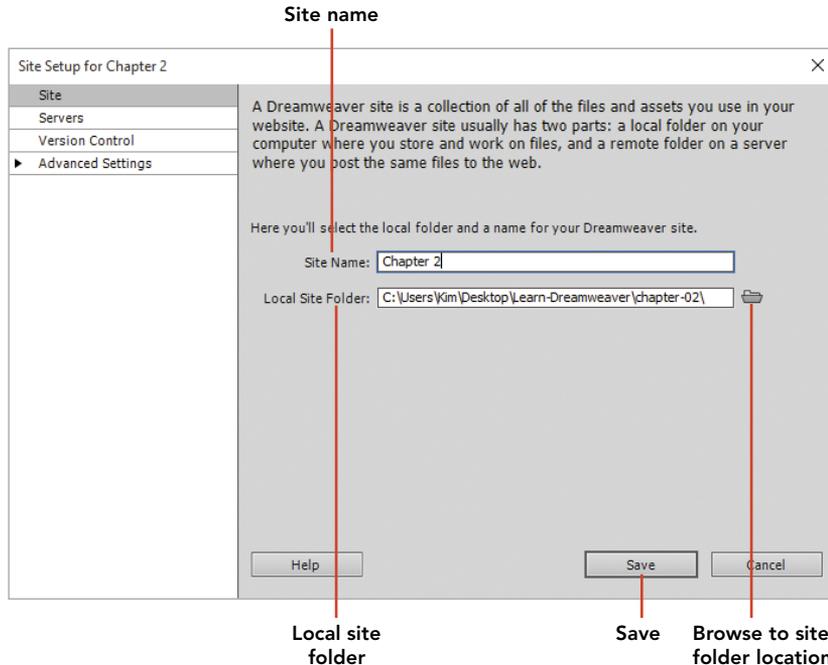
Refer to *Video 1.5, "Defining a Dreamweaver Website,"* if you want a quick demo of how the site definition process works.

- 2 In the Name field in the Site Setup dialog, name the site **Chapter 2**.
- 3 Use the folder icon to the right of the Local Site Folder field to browse to the **chapter-02** folder in your project folder.

Your Site Setup window should appear as you see in **Figure 2.3**. Of course your local root folder is on your *own* computer, so the path will be different than the one shown here.

- 4 Click the Save button.

**Figure 2.3** The Chapter 2 site is ready to go when your settings match the ones you see here.



## Dreamweaver Workspace Overview

★ ACA Objective 3.1

★ ACA Objective 3.2

▶ **Video 2.1**  
A Quick Tour of  
the Dreamweaver  
Workspace

Writing code for the web and working on websites is quite a bit different from creating an image or editing a photograph in Photoshop or designing an illustration in Illustrator. As a result, the interface in Dreamweaver has all sorts of different panels, inspectors, and visualization tools that help you get the work of coding for the web done.

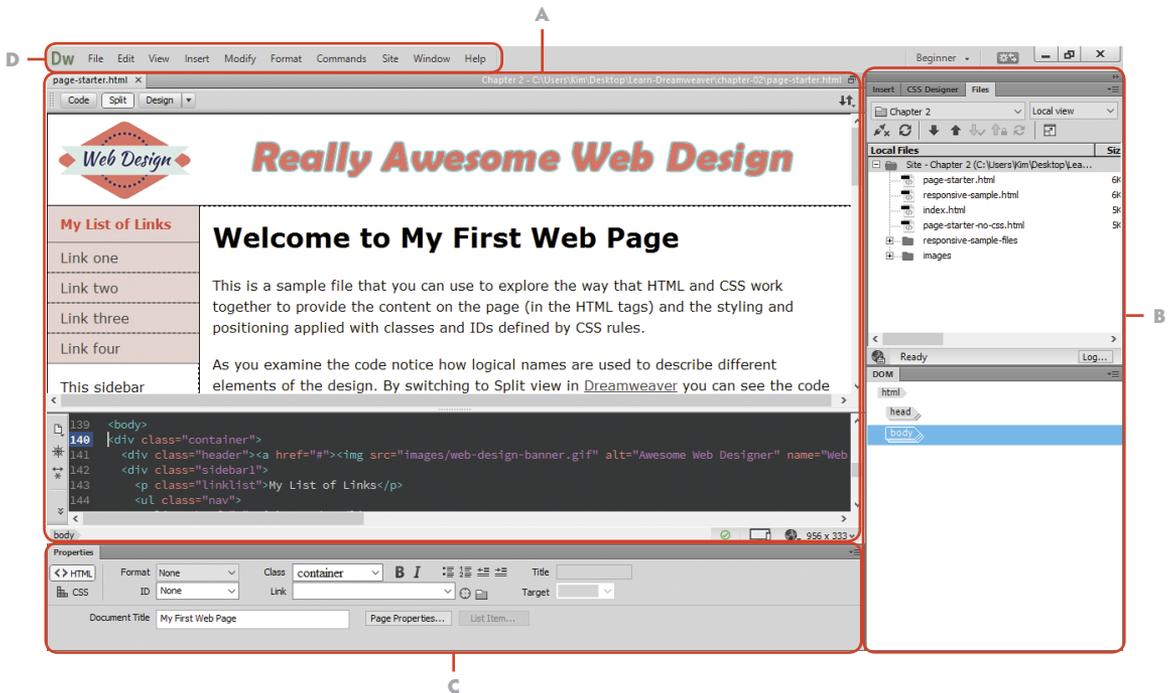
So let's start out with a quick guided tour of the entire Dreamweaver workspace before we look at each of the individual elements in more detail.

Open the files named `page-starter.html` and `responsive-sample.html` from the Files panel by double-clicking their filenames.

The Dreamweaver interface (**Figure 2.4**) has four primary elements.

- A** **The Document window** is where all the action takes place in your design. It's the place where your web page visually comes to life, as well as the location where you can simultaneously view the code of your page and the web page itself while in Split view, or work entirely in the code for the page by switching to Code view. You can choose to view your page in Design view, where you'll have complete freedom to enter text, format text, and drop objects onto the page. Or you can switch to Live view and see interactive elements, such as rollovers and videos, in a view that more closely mimics how a web browser will display a design.

The Document window also contains tools that allow you to change the layout of the page and a button that lets you preview pages in a web browser. You can even select tags directly on the page and in your code using the Tag selector.



**Figure 2.4** The four primary elements of the Dreamweaver workspace.

- B The panel groups** are located on the right side of the Dreamweaver workspace. This area displays panels that have specific functions that allow you to take all sorts of actions with your site and the contents of the files in your site. The most commonly used panels include the Files panel, CSS Designer panel, DOM panel, and Assets panel.
- C The Property inspector**, at the bottom of the Dreamweaver workspace, is the command center for each page. Using the Property inspector, you can (of course) inspect items within the page to see what properties have been applied to them, but you can also make significant changes to the code in your page in this area. This includes inserting hyperlinks, styling text, creating bulleted and numbered lists, as well as assigning styling properties when you switch over to CSS mode.
- D The application bar** is the area of the Dreamweaver workspace that runs across the top of the window, as is common for most computer applications. If you look at the top of the screen, you'll see the familiar File | Edit | View command layout. Dreamweaver does have a few little quirks where a certain command is located only in the application bar. For the most part, you use these commands as you do in most computer programs.

## The Document Window

★ ACA Objective 3.1

★ ACA Objective 3.3

▶ **Video 2.2** *The Dreamweaver Document Window*

The Document window is where your design comes to life. You can view the code in the page along with the visual design while in Split view, or you can work only in code in Code view.

Wrapped around these views of your design are a number of additional tools and inspectors that allow you to work on your design. Let's take these one by one as we review all the features in the Document window. Refer to **Figure 2.5** and follow along as each element of the Document window is described.

- A Document tabs** appear at the top of the Document window for every file that is open in Dreamweaver. You can use these tabs to switch between open documents, and you can close an open file by clicking the little x that appears next to the filename. One important visual aid on the document tabs is the asterisk that appears when you've made a change to the file but you haven't saved it yet. Keep an eye out for that little star that appears next to the filename, and save often!



**Figure 2.5**  
The Dreamweaver Document window

**B View switcher** contains three buttons that allow you to switch between the different ways you can examine and edit your page. You can see only the code that describes the page for a browser when you are in Code view, you can work in the code and in the visual design while in Split view, or you can see only the visual elements when you are in Design or Live view.

## WHICH VIEW SHOULD I USE?

As you begin working in your documents, you'll quickly learn to switch between the two visual displays of your design. Which one is right?

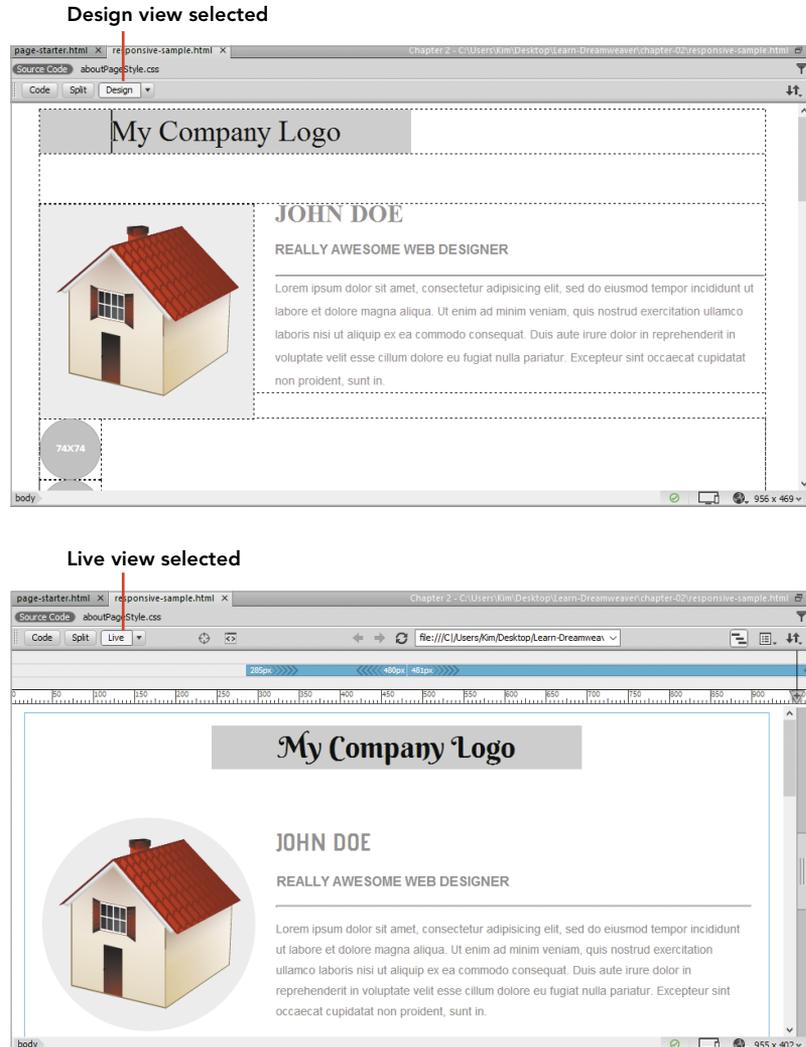
Design view is the easier setting when you need to work with lots of text, insert images, and work with the content that's in your page. Design view lets you insert all sorts of objects into the page and edit their properties.

Live view, on the other hand, shows a much more accurate view of the page as it would be viewed when someone visits your page using a web browser. You can work with most of the content in your design, but lots of people find that it takes a little more effort to get that done.

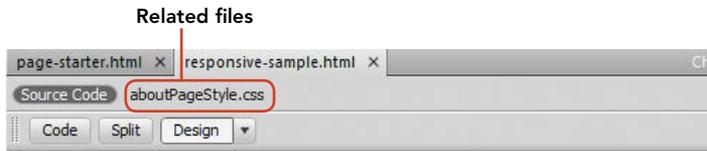
In the end, the choice is up to you to determine your favorite method of working in Dreamweaver. At the very least you'll find yourself switching between these two visual versions of your page to enter and edit content in Design view, then check to see how it would *really* look by switching over to Live view.

As you see in **Figure 2.6**, Live view shows a **font** (letter style) that is actually stored on Adobe's Edge Font service. Dreamweaver literally connects to the service, pulls down the right font just like a web browser would do, and then displays this live element right in the Document window. If you switch to Design view, you see the same text, but it's displayed in a font that is on *your* computer. You also see how the CSS positioning rules that are used in the design place content in the right location.

**Figure 2.6** Live view versus Design view



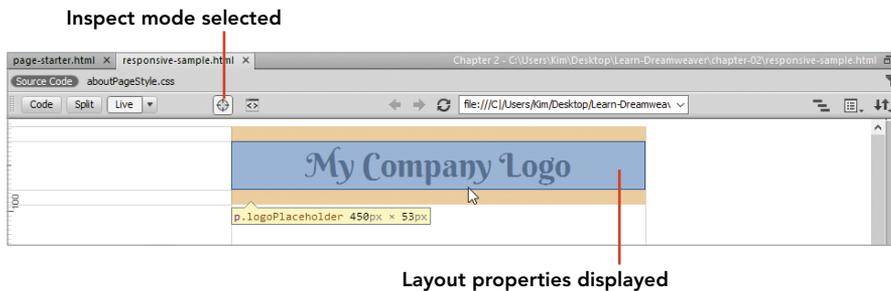
One final element in this area of the Document window is the **related files** that appear when there is a link from the page you're working on to another file that the browser is instructed to use. In the case of `responsive-sample.html`, an **external CSS file** has been linked to the page and all the styling rules that the browser uses are actually in that file. You'll learn soon enough that this is the preferred method for maintaining the styles for a website because it allows you to keep a master styling document that has to be changed only one time to update all the web pages that use the CSS document. In **Figure 2.7** you can see how Dreamweaver shows these related files in the area right underneath the tabs that show the filename; the file named `aboutPageStyle.css` has been linked to the page.



**Figure 2.7** Dreamweaver shows you all the files related to your web page file.

The next icons you see at the top of the Document window are located in the **Document toolbar**. If you switch to Live view the following items show:

- C Inspect mode** does pretty much what you think: It allows you to inspect elements on your page. When you click the Inspect mode button and float your mouse over the page, Dreamweaver shows you how CSS properties are set to position elements on the page. Inspect mode allows you to see how the **CSS box model** has been used to place different content boxes on your page relative to each other, and it uses color coding to show how margin and padding properties are applied to the element. **Figure 2.8** shows Inspect mode in action; the mouse is floated over the header text at the top of the page.



**Figure 2.8** Use Inspect mode to visualize the content containers on a page.

- D **Live view** shows both the content within the page and any code that is being pulled from an outside source. Many web pages contain content that is written directly within the HTML document itself, but they might also use JavaScript to pull content from a remote server or database.
- E **Back/Forward** buttons allow you to click a link within a page and go to another document in your site in Live view. You can use the Back and Forward buttons to move back and forth between pages just like you would in a live web browser.
- F **Refresh** reloads the page and updates the visual display so that your code displays the page properly. If you make a change to the code in a page, you will need to update the display of your changes in the Design area of the page. You'll also see a Refresh button appear in the Property inspector whenever you make a change to the code.
- G **File path** is a field that shows you exactly where the open file is located on your computer.
- H **Media Queries** is a button that displays the Media Queries bar below the Document toolbar, as you see in **Figure 2.9**. CSS styles that are written so that the page responds to a change in the size of the window, or viewport, are known as **media queries**. You can click directly on one of the colored blocks to show how the page would appear when it is viewed at a particular size. This is a really valuable tool to have at your disposal when you are designing pages that respond to a change in the viewer's screen size, as they do in `responsive-sample.html`.

Figure 2.9 The Media Queries bar



- I **Live view options** are available for how Live view will display the contents of your page. When you click this button, a fly-out menu appears that lets you turn on or off certain features, such as the display of JavaScript and other elements.
- J **File management** displays options that are available to you when you interact with a live web server. For example, you can use this button to copy files from your computer to a web server (**Put**) or download a copy of a file from the server to your computer (**Get**).

That concludes our look at the items that are located on the top of the Document window. The following elements in the Dreamweaver interface are located at the bottom of the window, as shown in Figure 2.5.

- K **Window size** is in the lower-right corner of the Document window and allows you to switch the size of the display to view your page as it would appear in different devices, such as the iPad, different iPhone models, and popular Android devices.
- L **Preview in Browser** pops up a list of web browsers installed on your computer; choose one to open the current page and see it as it would appear if it were opened on the web. These previews give you the most accurate view of a web page and are an important part of testing your page designs. Windows computers automatically display the option to view the page in Internet Explorer, whereas Mac computers will automatically have their Safari browser listed. You can add other browsers, such as Google Chrome or Mozilla Firefox, from this area as well.
- M **Device Preview** is very cool new tool (**Figure 2.10**) that has been added to Dreamweaver CC. When this button is selected, Dreamweaver pops up a window with a QR code that you can snap a picture of using your tablet or phone. As long as the page is open, you'll see a real-time display of the page you're working on right on your device. Slick!



**Figure 2.10** The Device Preview

- N Error checker**, as the name implies, gives you instant feedback on whether there's a problem in the code on the page. A green checkmark indicates that your code is good to go, whereas a yellow warning sign indicates there is a recommended change that you should be aware of. A red X means that there is a serious problem you need to fix or your page might not display correctly.
- O Tag selector** shows you the actual tags within the page and the structure of the page in a left-to-right arrangement. You can use the Tag selector to quickly select a tag within a document so that the tag can be removed or changed.

Whew! The Document window sure has a lot of tools and inspectors. But since this window is by far the most important element of the Dreamweaver design environment, it's important that you understand the ways these tools may be used to visualize the code beyond your designs. Now let's turn our attention to the Property inspector.

## The Property Inspector

★ ACA Objective 3.1

★ ACA Objective 3.2

▶ **Video 2.3** *The Property Inspector*

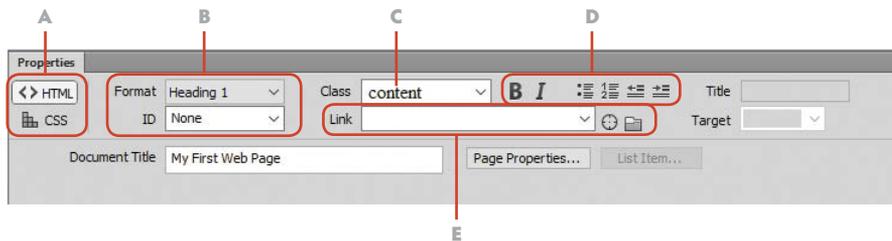
The Property inspector is the command center for working with your web pages and is probably the second most important element of the Dreamweaver work environment.

The Property inspector not only displays information about an item you have selected on the page, but it is also the place where you go when you need to perform common operations such as inserting links, assigning attributes, and making other changes to elements of your designs.

## The Property Inspector for Text in HTML Mode

With `page-starter.html` open, you can examine the different ways that the Property inspector appears based on the kind of object that you have selected in the Document window. Click in front of the first line of text in the page (John Doe) and have a look at **Figure 2.11**, where the various areas of the Property inspector are labeled.

- A Switch views of the inspector**, depending on the properties that you wish to view or edit. When you select HTML mode, as you see in Figure 2.11, the Property inspector displays attributes that have been applied to the selected tag. When you switch to CSS mode, you'll be able to see the styling properties that have been applied to the selection and make some changes to the CSS.



**Figure 2.11** The Property inspector as it appears when text is selected on the page while in HTML view

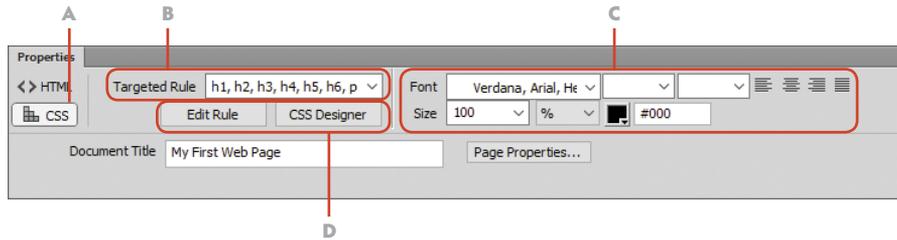
- B Format and ID settings** give you additional control. You can change the kind of tag format that is applied to text, including setting a line of text as a paragraph or heading tag. The ID refers to the identification, or ID, attribute that is applied to a selected tag. You can use the ID attribute to assign styling properties to an item using CSS, and this is where you can see what ID has been applied or add a new ID to a selected tag.
- C CSS Class rule setting** allows you to add a new class to a selected tag or object on the page.
- D Bold, Italic, List, and Indent settings** allow you to assign and remove various tags for text you have selected: the `<strong>` tag to make text bold, the `<em>` tag to italicize text, the `<ul>` tag for an unordered list, the `<ol>` tag for an ordered list, and the `<blockquote>` tag to indent. As you assign these properties, the appropriate tag will be written into the code for your page in the Property inspector.
- E The Link field and linking tools** allow you to assign or edit a link that is attached to text or an image. You can use the Link field to select an image. You use the small icon to the right of the Link field to drag a link to a file in the Files or Assets panel, or use the folder icon to open your system viewer and browse to find the file.

## The Property Inspector for Text in CSS Mode

When you change the view of the Property inspector from HTML to CSS using the buttons on the upper left, a different set of options and buttons appear. Refer to **Figure 2.12** for this review of the function of the options in CSS view.

- A CSS view** of the Property inspector provides different tools and options than the HTML version.
- B Targeted Rule** is the CSS rule that is applied to the currently selected text or tag on the page. You can use the drop-down arrow to the right to apply a different rule or to create a new rule using an inline style.

**Figure 2.12** The Property inspector as it appears in CSS view



- C CSS styling properties** allow you to choose a font for the selected text or tag; assign a font style, such as italic or oblique; set the **font weight**; align the text; set a font size; and set a font color. If you make changes to settings in this area, the CSS that defines the style will automatically be updated no matter where those rules are located.
- D Edit CSS properties in the CSS editor of CSS Designer** allows you to open two separate locations in Dreamweaver where you can edit a styling rule in one of two ways. If you select the Edit Rule button, Dreamweaver's CSS Rule Definition dialog pops up. If you select the CSS Designer button, that panel will open in the panel group area on the right side of the screen and the currently selected rule will automatically be selected.

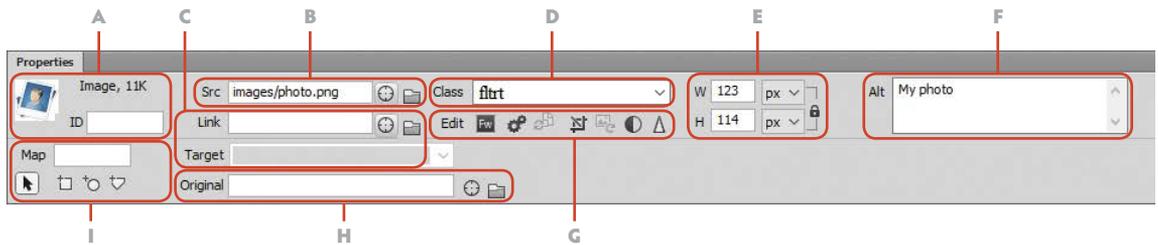
### CAUTION! HERE BE MONSTERS!

Well, maybe not monsters, but you can write some monstrously sloppy code by using the Property inspector. Sure, this is probably the *easiest* way to select a block of text or a tag and make all sorts of changes to how it looks, just like you would in a word processor, by creating an inline style.

But don't go there! Inline styles should be avoided whenever possible because they are much harder to maintain when you have more than one web page. It's much better to use CSS classes that are consistent across all your web pages by creating style sheets in the head of the page or, even better, in an external style sheet.

## THE PROPERTY INSPECTOR FOR IMAGES

The second most common elements that you'll likely work with using the Property inspector are images. Select the image on the right side of `page-starter.html`, and refer to **Figure 2.13** for this review of the options you see when an image is



**Figure 2.13** The Property inspector as it appears when images are selected in Design view

selected. The view you see here is shown when the Document window is in Design view mode.

- A Image thumbnail and ID** are displayed here. You can assign the image an ID for use in styling rules.
- B Path to image** lists the location of the file (its path) and the name of the file. The buttons to the right of this field let you drag a pointer to an image listed in the Files or Assets panel, or you can use the folder icon to browse to the image using a Windows Explorer or Mac Finder window.
- C Link applied to image** is displayed here. You can set the properties for how the link should open using the Target drop-down area below the Link field.
- D CSS class style** is the name of the currently assigned CSS class style. You can use the drop-down area to the right to change to different styling rules that exist in your document.
- E Width and height** for the image, measured in either pixels or percentage, can be changed here. With the lock icon active to the right of these settings, your image keeps the proper proportions if you change either of these values.
- F Alt text** must be provided to meet international standards regarding accessibility of web pages for disabled persons. **Alternative text** provides a short description of the image in case the image doesn't load properly in the browser. Alternative text is also used by those with vision problems who might use a software program that reads the text on the page to them.
- G Image editing tools** provide quick access to either Adobe Photoshop or Fireworks, based on whichever program is your default image editor. Clicking the program icon opens the image in the selected program, where you can make and save changes. This allows you to **round-trip** your edits using the source, and updating the image on the page when it is saved.

You'll also find buttons that allow you to adjust the optimization settings for the image, edit the brightness and contrast of the image, and even sharpen the image. Although these tools are fine for making very tiny corrections to an image you're using, you should keep in mind that any changes you make become permanent as soon as they are applied.

- H **Original image** is the path and name of the source image that you have used to create the image on the page. This is commonly done if you compose your image in Photoshop or Fireworks and then export the image to a format that can be used on the web.
- I **Image map tools** allow you to visually draw a hotspot in different shapes and assign its link properties. **Image maps** are clickable hot spots that can be placed above an image. This is achieved when Dreamweaver writes the code that maps out the location of the clickable region relative to the image.

## ADDITIONAL PROPERTY INSPECTOR VIEWS

As you continue to work in Dreamweaver you'll come across a number of ways that the options on the Property inspector are contextually presented. Tables and the tags that make up tables—such as `<th>`, `<tr>`, and `<td>`—all have specific options that are available for your use in viewing and modifying your code. You'll see those tools in action when you complete the hands-on exercises that are coming up.

The Property inspector also displays specific properties for multimedia objects, such as movie files, Flash movies, and others. You'll be turning to the Property inspector often as you view and edit your work.

## Panels and Panel Groups

★ ACA Objective 3.1

★ ACA Objective 3.2

★ ACA Objective 3.3

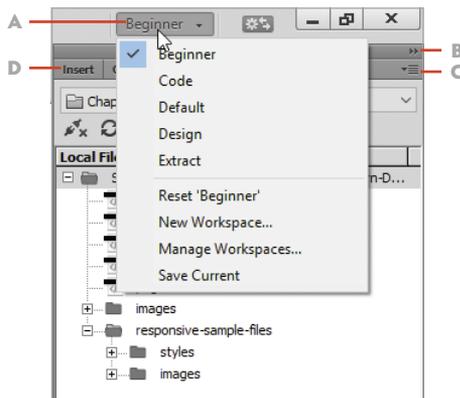
The third and final prominent elements of the Dreamweaver user interface are the panels and panel groups that you'll find on the right side of the window. Each panel in this area is designed to give you access to common actions you'll need to take when working on single documents or an entire website. The panels and the panel groups are arranged based on their function.

## Preparing the Workspace

You can customize the Dreamweaver workspace to fit the style of the kind of work you are doing. And no wonder! Dreamweaver is used by hundreds of thousands of people around the world, and they use many different kinds of technologies to create content for the web. Dreamweaver needs to be flexible to meet the requirements of its many customers and how they prefer to work.

For this overview of the behavior of the Dreamweaver work environment for panels and panel groups, open the `responsive-sample.html` file you've been working with from the Chapter 2 site in the Files panel. Then refer to **Figure 2.14** for an overview of the ways you can customize the appearance of the workspace.

- A Workspace Layout settings** allow you to quickly customize how panels and other areas of the user interface are displayed by choosing one of the **workspace layouts** that Dreamweaver provides.
- B Collapse or Expand Panel Groups** is a tiny button at the upper right of the Panel Groups area that allows you to collapse or expand the entire column of panels.
- C Panel options** are available for every individual panel through a button in the upper-right corner. At the very least, you have the option to close the panel. Some panels, such as the Files panel, provide several settings and operations.
- D Panel tab** displays the name of the panel. You can click the top of the Panel tab and drag it to pull the panel away from the column, or you can drag it to place the panel anywhere inside the Panel Groups area. If you double-click the top of a panel tab, the panel will collapse. A single click on a panel tab will open the panel again.



**Figure 2.14** Customization options in the Panel Groups area

**Video 2.4**  
Panel Groups in  
Dreamweaver

## Using the CSS Designer Panel

★ ACA Objective 2.4

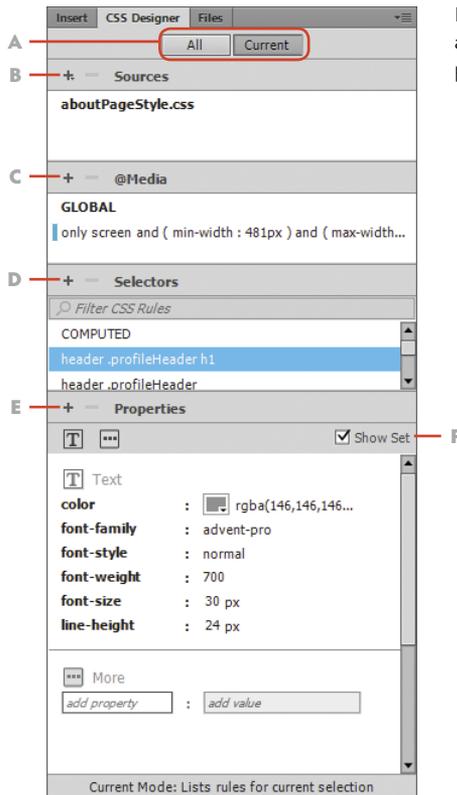
★ ACA Objective 3.2

The **CSS Designer panel** provides a location where designers can create, investigate, and modify the styling properties assigned to a page. Taming the many styling rules that make up an average web page is a tall order. But once you begin working with your own designs, you'll find that the layout and functions of the CSS Designer panel make sense.

▶ **Video 2.5** *The CSS Designer Panel*

For this look at the CSS Designer panel, open both `responsive-sample.html` and `page-starter.html` from the Files panel. You'll find the major functional areas of the panel labeled for you in **Figure 2.15**.

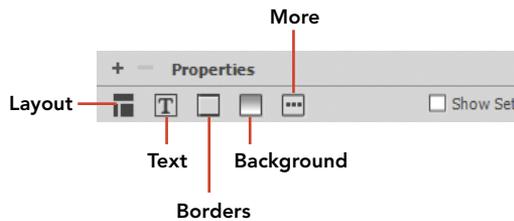
- A** Switch between **All** and **Current** view to change the appearance of the items shown in the CSS Designer panel. When **All** is selected, you see all the available sources, media queries, selectors, and properties.



**Figure 2.15** The functional areas of the CSS Designer panel

If you select **Current**, only the properties for an object selected in the Document window appear, along with the name of the selector, its source, and any applied media queries.

- B CSS Sources** are the locations of the CSS styles for the open document. If the style is located in the head of the document, you'll see the `<style>` tag. If an external style sheet is in use, as you see in `responsive-sample.html`, the name of the CSS file will be listed. You can add and remove style sheets by clicking the + and – buttons, respectively, in the upper-left corner.
- C Media queries** are those special CSS position and styling rules that transform a page based on the size of the screen. You'll see the kind of media that is modified in this area and the screen size that has been defined.
- D Selectors** show all the tags, classes, and ID styling rules that exist in this document or in any style sheet attached to the page. The list of the selectors allows you to choose a selector by its name, then view and edit the properties of the selector in the Properties area of the panel.
- E Properties** contains the many different CSS styling values that can be applied to a selector. This visual editor allows you to view only the properties and values that are in use for the current selector when the Show Set checkbox (F) is selected, or lets you see every possible value when that checkbox is deselected. The available values that you can set in the Properties section are organized using five buttons that appear at the top of the panel. These buttons appear as you see in **Figure 2.16**. You can use these buttons to filter the values that appear in Layout, Text, Border, Background, and More so that only the values in each category are shown.
- F Show Set checkbox** either shows the properties applied to a selected object or shows all the available properties.



**Figure 2.16** Use the categories in the Properties section to filter the available options.

## Using the Files Panel

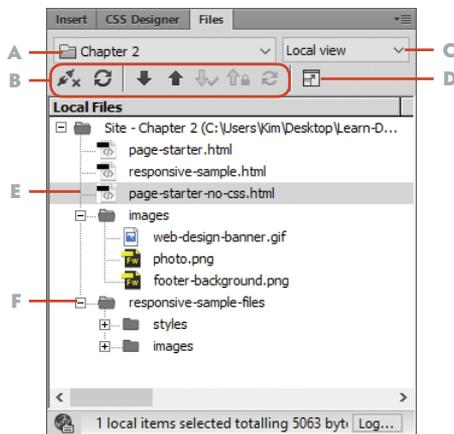
★ ACA Objective 3.2

★ ACA Objective 5.2

▶ **Video 2.6** The Files Panel

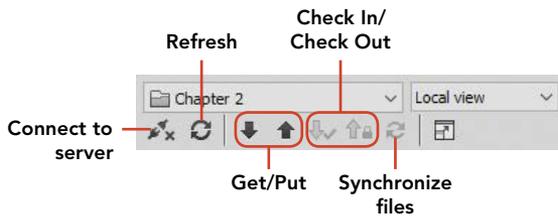
Dreamweaver provides numerous ways for web developers or even teams of developers to manage all the files and resources for a site. The **Files panel** makes all this possible with tools that let you manage, edit, and duplicate, and it provides a built-in **File Transfer Protocol (FTP)** program that allows you to work with files on a live web server.

You have already used the Files panel to define two websites so far, and you know how to open a file within the Files panel. Let's take a deeper dive now as you refer to **Figure 2.17**. Open both `page-starter.html` and `responsive-sample.html` for this review.



**Figure 2.17** The functional areas of the Files panel

- A Site name** displays name of the site you are currently working on. You can switch to other sites you've defined by clicking the drop-down arrow on the right.
- B Server operations** is a group of buttons you can use to open a connection to a file server on the web or on your network, to refresh the view of the files in your site, to copy files from a server (Get), or to publish files to a server (Put). In **Figure 2.18**, you can see these buttons and others that are typically used when multiple people are collaborating on a website. The **Check In** and **Check Out** options allow different people to work on a single website without worrying about one person overwriting someone else's file.



**Figure 2.18** Server options in the Files panel

- C View local or server files** allows you to switch the view of which files display in the Files panel by choosing the options in a drop-down menu. Local files are those that are on your own computer, but you can switch the view to see files published to a remote server or a testing server, or see files that are being managed in a version control system.
- D Expand to full-size Files panel** allows you to open a full-size version of the Files panel in a separate window so that you can compare the structure of a site side by side.
- E Right-click** any selected file in the Files panel to display a menu with numerous options for managing your files. You can make a duplicate copy of a file, copy or paste a file into a new location, or use other operations for working with the files in your site. Accessing menus by right-clicking will save you lots of time as your site becomes more complex.
- F Expand/collapse folder views** uses a “tree structure” to visually represent the files and folders within your site. The dotted lines show how your site is organized. Clicking the + icon (or the twirly arrow on a Mac) expands or collapses the contents of the folder.

## Using the Assets Panel

The Assets panel gives Dreamweaver users a central location where they have access to the various images, colors, links, movies, and other resources found within the site.

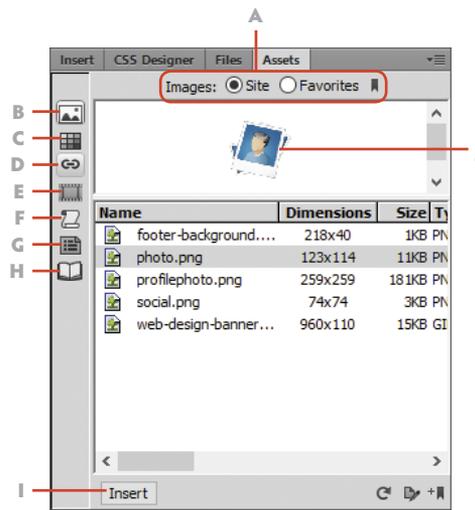
The Assets panel does not appear in the Beginner workspace layout that you’ve been using, but you can open the Assets panel by selecting **Window > Assets**. The Assets panel will appear beside the Files panel in that panel group.

Continuing on with your tour of the Dreamweaver workspace, refer to **Figure 2.19** and note the location of the primary tools in the Assets panel.

★ *ACA Objective 3.1*

★ *ACA Objective 3.2*

▶ **Video 2.7** *The Assets Panel*



**Figure 2.19** The functional areas of the Assets panel

- A** **Site or Favorites view** allows you to toggle to see either all the assets for a site (Site) or those assets you've set as a favorite so they can be accessed more easily (Favorites).
- B** **Images**, when selected, allows you to see a listing of all the images within a site, no matter what folder they are stored in. A thumbnail version of the image appears in the preview area (J), making it easy to check that you have the right image before you insert it into the page.
- C** **Colors** shows all the color values that have been specified in any styling rules. You'll see the color values written in both **hexadecimal** and **RGB** code, along with a preview of the color in the preview area.
- D** **URLs**, which are the links to external websites, display in this area.
- E** **Media** shows any media files, such as movies or interactive Adobe Flash content, that are used in the site.
- F** **Scripts** displays the JavaScript files and VBScript files used in a site.
- G** **Templates** shows the template files used in the site. Templates are complete web pages that are used for automating the production of an entire site.
- H** **Library items** shows all the Library items used in the site. A Library item is reusable code that can be used throughout a website. When the code in a Library item is changed, all the pages that use the item can be easily updated.

- I **Insert** is a button at the bottom of the Assets panel that allows you to insert a selected asset into a page. Typically this button is used for inserting images and media files.
- J **Asset preview** displays image thumbnails, colors, and other information about assets that are cataloged in a site.

## Coding Tools in Dreamweaver CC

Dreamweaver has a whole set of features that help you visualize and work with code more efficiently. You can view the code that makes up a web page in the Document window by using Split or Code view. Or you can open a separate code window from the Window menu on the application bar. No matter how you access the code in your page, Dreamweaver will continuously update the visual display of your work.

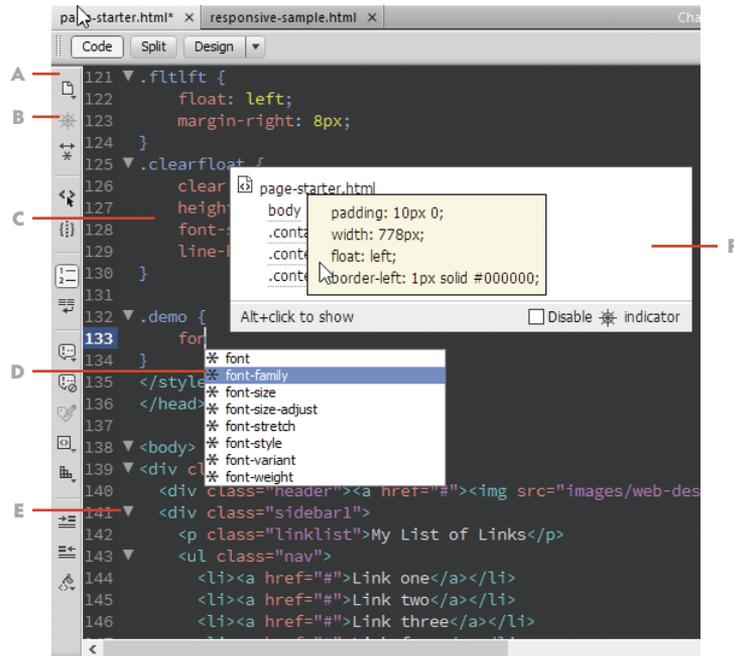
When you're working in code, Dreamweaver provides color coding so you can clearly see the distinction between tags, attributes, and values. You can also take advantage of the code hinting feature, which lets you write and modify your code more quickly. Refer to **Figure 2.20** for a run-through of the coding tools in Dreamweaver.

- A **Coding toolbar** contains buttons that let you work with your code by collapsing and expanding code blocks, inserting comments, and performing other functions that help modify and clean up code in a document.
- B **Code Navigator** is a button that opens the Code Navigator (F) for a selected tag. This “floating” icon also appears when you are in Design view and select a new area of the page.
- C **Syntax coloring** is a Dreamweaver feature that color-codes the tags and CSS rules in your code and then displays the attributes, values, and content within the tag in different colors. This helps you easily see that the syntax is correct and helps in hand-coding your content.
- D **Code hints** appear as you type within any code window, displaying and filtering the available tags, CSS selectors, and properties. To accept a suggested value from the list, just highlight the value and press Enter/Return on the keyboard.

★ *ACA Objective 3.1*

▶ **Video 2.8**  
*Coding Tools in  
Dreamweaver CC*

**Figure 2.20** Dreamweaver coding tools

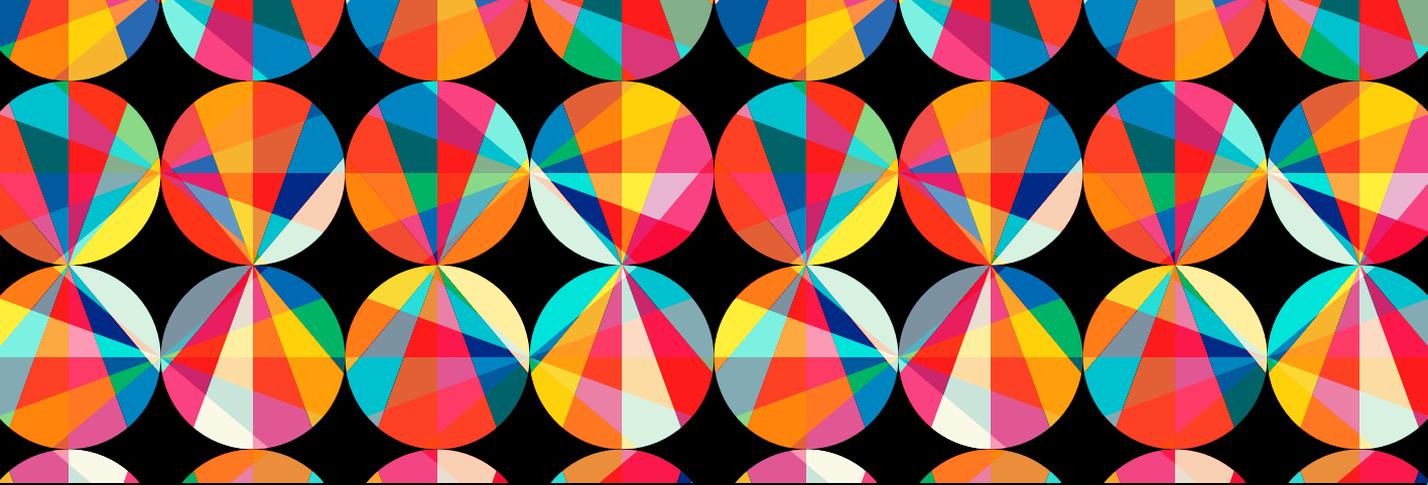


- E** **Code collapse** allows you to hide or display a block of code. Small triangles appear next to the line numbers in a block of code as you hover your mouse in this area. The same button allows you to expand the selection when you're ready to see more of your code.
- F** **Code Navigator window** is a thumbnail version of all the CSS settings that are applied to a tag. As you hover your mouse over the list of selectors, the properties that have been applied to the selector are displayed. If you click a selector name, you will jump to the CSS settings in your code.

# Conclusion

Wow. That is a huge load of windows, panels, inspectors, fly-outs, buttons, and options that are packed into Dreamweaver. But remember that web design is pretty complicated. Not because writing good, valid code is so difficult, but because the *medium* of the web—with its unique ability to display documents at all sorts of sizes and on many different devices—requires a lot from the web designer. Dreamweaver helps you to meet those requirements by giving you tons of ways to visualize, modify, and manage the pages and documents that make up a website.

As you move onward and upward in learning how to use Dreamweaver—and to study for the Adobe Certified Associate exam—return to this chapter to review the user interface features in Dreamweaver and the methods used to create awesome content for the web.



# CHAPTER OBJECTIVES

## Chapter Learning Objectives

- Learn project planning techniques to determine the structure of a website and create the folders and files that support the client's requirements.
- Add content into web pages using both the Document and Code windows.
- Modify the basic properties of a web page.
- Modify text elements of a design, including formatting and links.
- Examine wireframe documents and translate the specifications into page design.
- Understand how content containers are created using <div> tags and segment a web page into separate containers.
- Complete the layout of a page by assigning CSS styling and positioning properties to page containers.
- Employ various methods for visualizing, editing, managing, and creating styling rules.

## Chapter ACA Objectives

For full descriptions of objectives, see the table on pages 240–246.

**DOMAIN 1.0**  
SETTING PROJECT REQUIREMENTS  
1.1, 1.2, 1.3

**DOMAIN 2.0**  
UNDERSTANDING WEB AUTHORING  
2.2, 2.3, 2.4

**DOMAIN 3.0**  
UNDERSTANDING ADOBE DREAMWEAVER CC  
3.2

**DOMAIN 4.0**  
WEB AUTHORING USING ADOBE  
DREAMWEAVER CC  
4.1, 4.2, 4.3

**DOMAIN 5.0**  
TESTING, PUBLISHING, AND MAINTAINING WEB  
CONTENT USING ADOBE DREAMWEAVER CC  
5.1, 5.2

## CHAPTER 3

# Building Your First Website

If you've worked through all the lessons in the first two chapters, you should have a solid understanding of how web pages are created using coded instructions and of how Adobe Dreamweaver allows you to visualize, modify, and manage all the web pages you'll create for a typical website. Well done!

Now that the basics are under your belt it's time to dig in and make some real, live web pages. In this chapter, you will learn about the process of working with a customer by creating an entire website from scratch. From there, you'll begin making your first web pages come to life with text, styles, and images. Let's get started on your next mission by meeting your new client.

## Your First Client— Chris the Cartoonist



**Figure 3.1** Meet Chris, your first customer.

Over the course of the next three chapters, you'll be working with Chris, a cartoonist friend who is launching a new comic strip and needs your help. Chris's comics (**Figure 3.1**) will be based on a world he's creating in which a group of werewolves, zombies, vampires, and other monstrous creatures operate a company that provides security services. Kind of goofy, but that's Chris for you.

Chris is hard at work on the storyline and artwork for his new comic strip, but he's asked if you can get a website together for him, just like a company in the real world might have. And of course, knowing what you know now about writing code for the web and working in Dreamweaver, you're ready to dive right in.

You have only a little bit of information that Chris has provided in an instructions file that you'll find in the site files for Chapter 3. And of course Chris is working on a deadline, so you'll be getting information from Chris as you move along. Just like in the real world of web design.

### MEET THE ARTIST



**Figure 3.2** Chris Flick

Chris the cartoonist is the alter ego of Chris Flick (**Figure 3.2**), a real live person (or so he claims) who works by day as a professional user experience designer in the Washington, D.C., area. Chris also has a passion for cartooning, and he operates the "Capes & Babes" online comic at [www.capesnbabes.com](http://www.capesnbabes.com). You can find Chris at comic conventions on many weekends all over the U.S. and especially in the Northeast.

Chris has been kind enough to provide the cartoon artwork for this book, and you have his explicit permission to use the images that come with this chapter's project files as you wish.

If you're interested in cartooning, you should also check out the *Webcomic Alliance* podcast ([webcomicalliance.com](http://webcomicalliance.com)), where Chris hangs out with his comic artist buddies as they discuss what it takes to be a working creative artist.

# Get Your Files Here

To get started on your mission to build a website for Chris, you'll need to get the initial files that he's provided. As you did in Chapters 1 and 2, copy the folder that contains these files to your computer and define a website in Dreamweaver.

- 1 Download the project files for this lesson, named `chapter-03.zip`, from the Lesson & Update Files tab on your Account page at [www.peachpit.com](http://www.peachpit.com), and store them on your computer in a convenient location.
- 2 Unzip the files and copy the `chapter-03` folder to the `Learn-Dreamweaver` folder you created in Chapter 1.

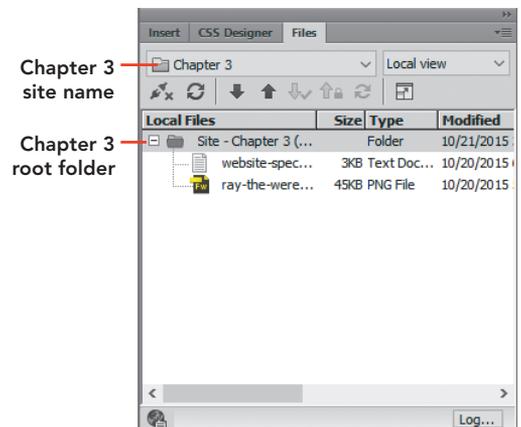
## Define the Chapter 3 Website

★ ACA Objective 4.1

This is now the third time that you've defined a website in Dreamweaver, and you should be getting pretty good at it. Remember the process. You first get your files into the proper location on your computer, and then you set up the site in Dreamweaver by giving it a name and pointing to the project folder for this chapter. Ready? Let's go!

- 1 From the application bar at the top of Dreamweaver, select `Site > New Site`.
- 2 In the Site Setup dialog Site Name field, name the site **Chapter 3**.
- 3 Use the folder icon to the right of the Local Site Folder field to browse to the `chapter-03` folder in your `Learn-Dreamweaver` project folder.
- 4 Click the Save button to save these settings.
- 5 Check the site name at the top of the Files panel. With your site properly defined, it should appear as you see in **Figure 3.3**.

**NOTE** Refer to Video 1.5, "Defining a Dreamweaver Website," if you want a quick demo of how the site definition process works.



**Figure 3.3** The Chapter 3 website as it appears in the Files panel

## PROJECT 3.1

# Translating Client Project Requirements

★ ACA Objective 1.1

★ ACA Objective 1.3

★ ACA Objective 4.1

★ ACA Objective 4.2

★ ACA Objective 5.2

▶ **Video 3.1**  
Translating  
Requirements into  
Structure

Chapter 9 of this book has a comprehensive guide to working with clients and outlines the steps that a typical web design project follows. But guess what? In the real world of working with clients, things don't always follow the optimal path to get to where your client wants to go. You'll often be faced with a task like the one you'll tackle here: Take some sketchy information that a client has provided and then translate it into a functional website. Even with a lot of information missing, you can still do some of the first steps needed in the creation of a new website.

For this project you'll do the following:

- Identify the purpose and the audience for the website.
- Work with the client and create a plan for the site.
- Structure the site following the standard methods used for creating websites.

You'll do that now by reviewing the list of requirements that Chris has provided, and then you'll build out the structure of his new site in Dreamweaver.

## Think Like a Boss

Starting a new design project is such an exciting time. It doesn't matter whether the idea is one of your own or you've been asked (or hired) to design a new website. Getting to see something new and original come to life before your eyes is pretty cool.

But before you get to all that coolness, you need a plan. If you follow the planning process outlined in Chapter 9, you'll see that the first step in the planning process is to learn what the *problem* is.

The problem for most design projects is to figure out how you can help your clients share their goals with the people they want to reach—their target audience. The biggest challenge as a designer is really understanding the client's goals and message and identifying the target audience. Often, even the client won't clearly know what to communicate.

To begin understanding where your client is and what he wants, you need to view the project specification file.

Open the file named `website-specs.txt` from the `chapter-03` project folder. Dreamweaver will open this file in Code view if you double-click the file in the Files panel, but it might be easier to read in a word processing program such as Microsoft Word or a text editor of your choice.

As you read the file, consider some common questions that are typically asked during a client interview process. It's important that you learn as much as possible about the goals your client has at this point.

- What is the purpose for the website?
- What are your goals for this website?
- What key information do you want visitors to see?
- Who is the audience for this website?
- What key adjectives (goofy, modern, dark, light, serious, friendly, and so on) would you use to describe the look and feel of the site?
- What is the title of the site?
- What topics are covered in your site?
- What is the domain name?
- Are there particular colors or fonts that you like?
- Do you have existing artwork, such as a logo, pictures, or other media, you want on your website?
- Do you have examples of other websites that you like?

In the case of this project, where you can't really interview your client, you have to rely on the instructions in the specifications file. From the specifications, you can determine the following about the work that Chris wants:

- **Site title:** Werewolves UnLimited Valuables Security Services, Inc. (WULVS).
- **Domain name:** *wulvs.com*.
- **Key information:** The website supports a fictional company run by cartoon werewolves, zombies, and other creatures.
- **Look and feel:** Kind of goofy, but looking like a "real" company website.
- **Topics and pages:** Home, About, Services, Rates, Contact.
- **Colors and fonts:** The client doesn't really know.

That might not seem like a lot, but in the real world the instructions you've been provided are better than most. In fact, with just that bit you can move to the next phase of making this site—creating the file and folder structure that will support the client's requirements.

Finally, the client also provided some sample artwork that helps to determine the look and feel of this site. If you open `ray-the-werewolf.png` in Photoshop or Fireworks, you'll see that the style of the featured cartoons is meant to be fun and a little goofy. This will help establish the look and feel for the site as you begin your design work.

## Creating the Site Structure

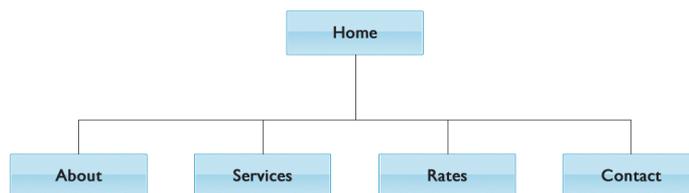
Recall from Chapter 1 that most websites follow a standard folder structure and file naming convention. It's essential that you tend to the **information architecture** of a website right from the start. Having a site that is properly organized will make things far easier to maintain as time goes on. But perhaps more importantly, the structure of the site helps with the **usability** of the site. In other words, a well-structured website should make it easy for visitors to find their way around and determine where they are at any given moment.

The site structure also allows the client to easily tell visitors how to navigate to content within the site. If, for instance, Chris wanted to tell someone how to navigate to his contact page, it's easy to say, "Just go to `wulvs.com/contact`." Compare that address to one you might find in a poorly designed structure where files are assigned without any thought to how they'll be available to visitors. In that case, poor Chris might have to tell someone "Go to `wulvs.com/pages/contact-us.html` to get in touch with us." Try comparing those two by saying them out loud and you can tell the difference right away.

One method for sketching out how a website should be organized is the **sitemap**—a visual representation of how the content of the site is arranged. You learned about how websites are organized using folders in Chapter 1, and having examined the instructions that you've been provided you should already have a good idea that this site will be organized as you see in **Figure 3.4**.

At this point, a completed sitemap is all you need to jump over to Dreamweaver and start creating the files and folders for this site using the Files panel.

**Figure 3.4** The initial sitemap for `wulvs.com`



## Creating the Site Structure in Dreamweaver

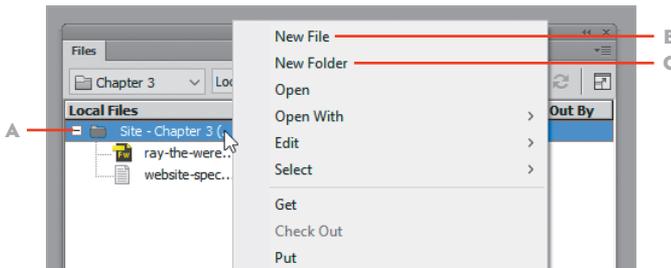
The process of creating the files and folders for this site is pretty straightforward. Use the Files panel in Dreamweaver and the following steps to build out the site. Refer to **Figure 3.5** to complete these steps.

- 1 Right-click the site (root) folder at the top of the Files panel.
- 2 From the context menu, select New File.
- 3 Change the name of the file from `untitled.html` to **`index.html`**.

You now have a home page for the site.

- 4 Right-click the root folder, and choose New Folder.
- 5 Name the new folder **`images`**.

You now have a location to store the images that will be used in the site.



**Figure 3.5** When you right-click a folder in the Files panel, a context menu appears.

- A Right-click the root folder.
- B Choose New File to create a file in the selected folder.
- C Choose New Folder to create a folder in the selected folder.

- 6 In the Files panel, drag the `ray-the-werewolf.png` file into the `images` folder you just created.

You now have a location in which to store the images that will be used in the site.

- 7 Create four new folders in the root folder. Name these folders **`about`**, **`contact`**, **`rates`**, and **`services`**.

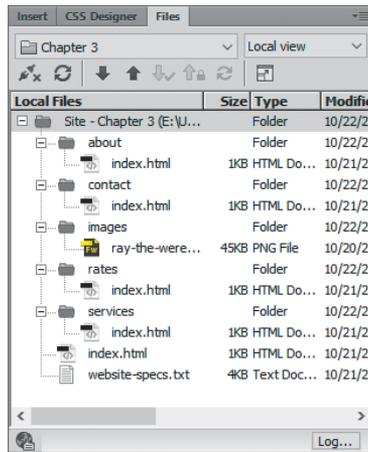
You now have locations in which to store the home pages for each area of the site.

- 8 Right-click each of the folders you just created, and select New File from the menu. Name each new file **`index.html`**.

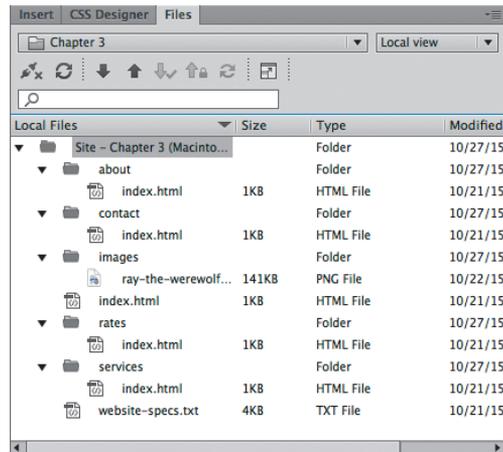
You now have a home page for each topic area of the site.

With all those actions complete, your Files panel should appear as you see in **Figure 3.6**.

**Figure 3.6** The initial site structure for Project 3.1 as seen in the Files panel on Windows and Mac



Windows



Mac

You'll notice one minor difference between a Windows computer and a Mac when it comes to the Files panel. On a Windows computer, folders are listed in alphabetical order first, followed by files. On a Mac, files and folders are listed in strict alpha order without regard to whether it's a file or a folder.

You're now ready to move along to the next project.

## LEVEL UP: GRAPHIC DESIGN

The next step in the design process for a project like this would typically be to start getting ideas onto paper and, ultimately, create sketches in your favorite graphic design program. Of course, Adobe makes it easy to work between Dreamweaver and Adobe's two image editing programs. Adobe Photoshop may be your design weapon of choice, or you might want to use Adobe Fireworks CS6 instead.

Whichever program you use, creating mockups of how the pages in the site should look usually begins right about now. With the information this client has provided, you can draw how you think the site should look. Image editors don't have all the technical challenges that web pages have, where code has to be written to define how things look, so you can let your imagination have free rein. Now is an excellent time to work on your own unique design for *wulvs.com*, or think about how you would design a site of your own with a structure similar to the one you see coming together here.

## PROJECT 3.2

# Modifying Page Properties

Now that the structure of the site has been prepared, it's time to turn your attention to the initial work that can be done in formatting and styling the pages. Even though the client hasn't provided all the information that's needed for this site to come together, there's still a lot that you do know based on the specifications you've been provided.

For this project, you'll complete the following tasks:

- Review the requirements of the site based on the client's instructions.
- Think about the look and feel that the client wants to achieve with the site.
- Determine how the content of the page should be organized.
- Add content onto the home page so you have material to work with as you continue to build out the design.
- Use the Page Properties dialog to assign values to the working document in the site.
- Copy and paste critical information from the client's specifications onto the working web page.

★ *ACA Objective 1.2*

★ *ACA Objective 2.4*

★ *ACA Objective 4.2*

 **Video 3.2**  
*Modifying Page Properties*

## Taking the Next Step in the Design Process

In Project 3.1, you started reviewing the client's requirements for the new website, and you determined some critical information about the structure of the site.

Now it's time to move along to the next logical step in the design process—to learn more about what the client hopes to achieve, and thinking about how you as the designer can meet those needs.

Many times this involves a second interview or an exchange of emails and phone calls with the client to get a feel for exactly what they're looking for. It's also common at this stage to be thinking about the look and feel for the customer's website. There's still plenty of work to do before the real visual design can begin, but having an idea of what the client wants in general terms will help get things started out right.

Although the visual design process has not yet begun, this is a good time to find out how your client envisions the visuals for the new page. You might want to ask your client to send you links to other sites that he likes and to describe what it is he



**Figure 3.7** Ray the hipster werewolf

likes about those sites. Your client might also have an existing logo or artwork that will help you pick up color schemes and that all-important “feel” that the client is after. Your cartoonist client, Chris, has provided you with a sample of his work, Ray the hipster (**Figure 3.7**), and judging from the appearance of the werewolf character, you can tell that this is a site that is meant to be fun and a bit goofy. Imagine if Ray weren’t this happy, hipster werewolf and was instead a scary, snapping, dangerous-looking creature. The site would have an entirely different feel, and the design process would be heading off in a whole different direction.

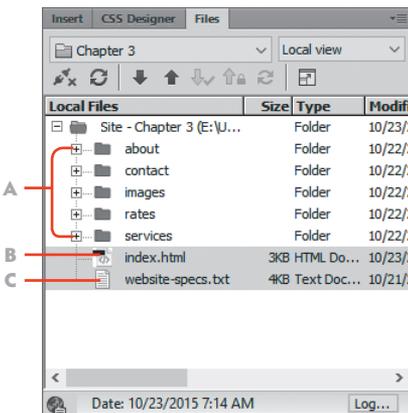
## Examining Client Resources

As you read through the client’s specifications, you’ll see that there is a fair amount of information provided. You’ll use this information right within Dreamweaver as content is added to the home page of the site. The home page will become your working document, and you’ll be adding content and styling to this page as the design comes together. From what the client has already said that he wants, you can determine the following critical information:

- **Site name:** Werewolves UnLimited Security Services Inc.
- **Header Text:** Werewolves UnLimited Security Services Inc.
- **Tagline:** “Let us protect your stuff with a snarling pack of crazed creatures!” (Maybe)
- **Navigation:** Links to the pages within the site
- **Content:** The text beginning with “Welcome to Werewolves UnLimited Valuables Security Services Inc.!”

Open the two files that were initially provided to you by double-clicking `index.html` and `website-specs.txt`. Double-clicking opens the files in Code view, making it easy to copy and paste between files.

Be sure that you open the correct index file. You can collapse the subfolders in the site so you don’t confuse those home pages in the Files panel with the primary home page of the site, as you see in **Figure 3.8**.

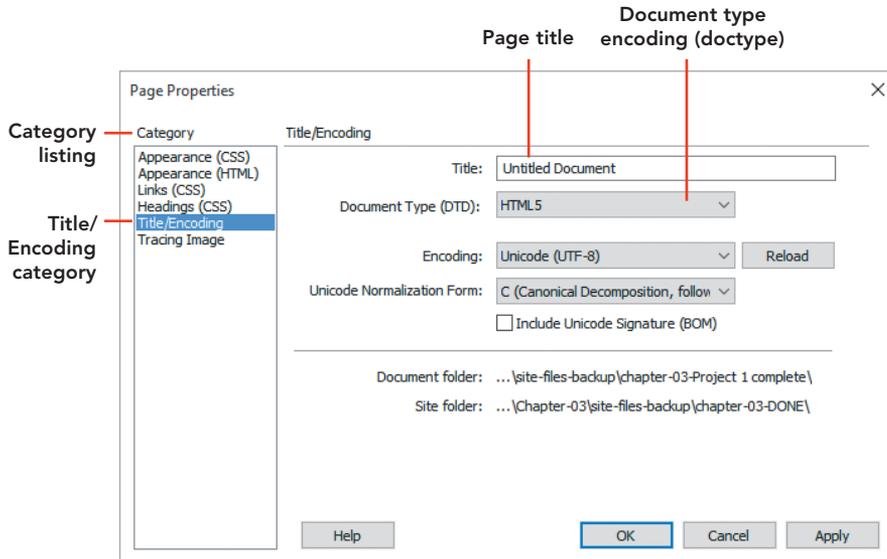


**Figure 3.8** Open the documents for this project from the Files panel

- A** Collapse subfolders to hide their contents
- B** Open `index.html`
- C** Open `website-specs.txt`

# Adding Content to a Web Page and Setting Basic Properties

With both the `index.html` and `website-spec.txt` documents open, you'll be able to switch from one file to another by clicking the file tab at the top of the Document window. This will be helpful as you follow along with these steps and put content onto the page. Refer to **Figure 3.9** to complete the first part of this project.



**Figure 3.9** The Page Properties dialog provides a location where properties for an entire document can be set based on different categories.

The first thing you need to tend to is the **page title**. Titles are the names that humans read when they see a page name listed in the title bar of their browser or when they save the page as a Favorite or Bookmark. Page titles also appear in a browser's history listing and appear when the page is found by a search engine. This differs from the *filename* (`index.html` in this case), which is what computers need when a request to open the page is sent to a web server.

- 1 Open `website-specs.txt`, and highlight the block of text that contains the name of the company: Werewolves UnLimited Valuables Security Services. Copy (CTRL+C/Command+C) the highlighted text.
- 2 Click the tab `index.html` at the top of the Document window to switch to that file. In the application bar at the top of the screen, choose **Modify > Page Properties**. This opens the Page Properties dialog you see in Figure 3.9.

- 3 Select the Title/Encoding category on the left side of the window.
- 4 Click in the Title field, delete the existing text, and replace it by pasting in the company name you copied in step 1. You can use the standard keyboard shortcut of CTRL+V/Command+V to paste in the text, or right-click and choose Paste.
- 5 Type the word **Home** after the company name. Your completed page title now reads Werewolves UnLimited Valuables Security Services Home.

For new documents you create for entirely new websites, you should stay with the most current HTML specification: HTML5. You would only change this setting if you were working on an existing website where older pages are in use and you need to match the existing encoding.

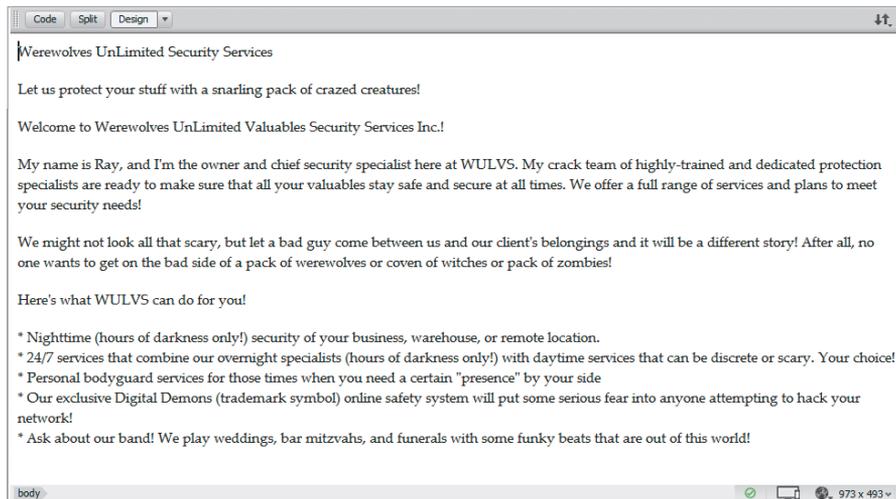
- 6 Leave the Document Type field set to HTML5, and click OK. Your page now has a title.

To get the rest of the content onto the page, you'll return to the text file and copy and paste the essential elements that the client has provided.

- 7 From `website-specs.txt`, copy the company name, tagline, and introduction to the company that the client has provided and paste them into `index.html`.

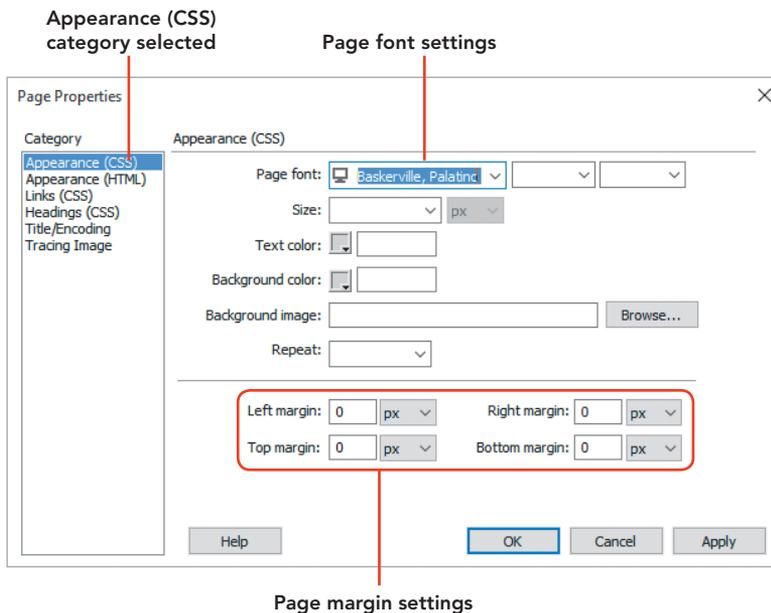
Your completed document (in Design view) should appear as you see in **Figure 3.10**.

**Figure 3.10** Plain text can be copied directly into the Document window to add content to a page.



## ADDING ELEMENTAL CSS PROPERTIES

The final few steps for this project include setting some basic styling properties for the page. One of the little things that drive web designers crazy is how page margins change based on which browser is being used. One browser might add 5 pixels to the left and top of the page window, and another might make that setting 3 pixels. To take control of this little discrepancy between browsers, it's standard practice to “zero out” the page margins by adding some CSS. While you're at it, also set a default font that you might want to use in this design. Both of these actions take place back in the Page Properties dialog. Refer to **Figure 3.11** as you complete the final steps in this project.



**Figure 3.11** The CSS category of the Page Properties dialog

- 1 In `index.html`, choose `Modify > Page Properties`. Click the `Appearance (CSS)` category on the left.
- 2 In the margins area at the bottom of the window, enter `0` for each of the four margin properties.
- 3 At the top of the window, use the drop-down menu to set the default font for this page from the available fonts. This example uses `Baskerville, Palatino`. Click `OK/Apply`.

- 4 Review the changes that have been made to the head of the document. You'll see these two styling declarations listed because Dreamweaver (and you) wrote the code to set the margin properties of the page and the font to be used in the document.

Take note of one little thing in the CSS. Notice how the first rule combines three tag selectors so that the font used within the page (<body>) is the same font used in table cells (<td>) and table headers (<th>). This takes care of the places where most text might appear, and using the kind of selector that styles more than one tag with a single set of properties and values is a very efficient way to write your code.

These kinds of style rules, which define how a tag should appear at a very elemental level, are known as **element selectors**. With these rules in place, every time the tag is encountered the browser will apply these styling rules.

```
<style type="text/css">
body,td,th {
    font-family: Baskerville, "Palatino Linotype", Palatino,
    → "Century Schoolbook L", "Times New Roman", serif;
}
body {
    margin-left: 0px;
    margin-top: 0px;
    margin-right: 0px;
    margin-bottom: 0px;
}
</style>
```

And that takes care of adding content to the page and setting some basic styling properties. Now, be sure to save your work, and get ready for the next project.

## PROJECT 3.3

# Setting Font Formats and Links

Now that the structure of the site has been prepared and some basic content is in place, it's time to turn your attention to the initial work that you can do in formatting and styling the pages. In this project, you'll add some additional text to the page and prepare the content on the page for further production by creating links to files that are internal to the client's website.

Here's what you'll get done in this project:

- Type directly into a document.
- Learn about the difference between paragraph (`<p>`) tags and break tags (`<br>`).
- Use the Property inspector to convert the format of text.
- Add links to your document using three common methods that Dreamweaver provides: directly entering the URL in the Property inspector, dropping a link onto the page, and browsing to a file location on your computer.

★ ACA Objective 2.2

★ ACA Objective 2.3

★ ACA Objective 4.3

 **Video 3.3** *Setting Font Formats and Links*

## Working with Text and Text Formats

Text formatting in a web page is a lot like the styling rules that you'll find in a word processing program—word processors can do way more than just setting the size and color of text on the page. During the evolution of this book, for instance, very particular styles are being used so that the many people who work on the document can track what's going on and convert the Word document to the layout you see on this page.

When text is formatted in an HTML document, very fundamental changes happen to how browsers and even search engines interpret what's inside tags. For example, if text is formatted as an `<h1>` tag, a search engine considers that element to be the most important one on the page. In the same way that you might outline a topic for a school report, heading tags assist the search engine (and we're thinking Google here) to find content on the page based on how important its meaning is. Creating **semantic** structure in a web page is one of the many ways you can help your clients achieve their goals—because everyone wants to be Googled!

**Text format** is also used to define lists on a page. In modern web pages, the unordered list (`<ul>`) in particular gets used a lot. The `<ul>` tag and the items that are

listed inside the unordered list using the `<li>` tag are very flexible containers for styling elements on the page, and are probably used more for page navigation than they are for making lists. Dreamweaver has numerous ways for you to format text in a document, with most of the action taking place in the Property inspector.

## HOW YOU LOOK ISN'T WHO YOU ARE

How you're *formatting* text is very different from how you're *styling* text. The format of the text determines its purpose within the document and doesn't really have much to do with how it looks. Sure, heading tags have some default sizes, but for the most part, the format of text is used to give it meaning, not to change its appearance.

Styling text is an entirely different thing. Just like you can dress up based on what you're doing—going to school, going to work, heading out to the movies, or playing a sport—you can dress up text in a web page just about any way you want. The tag (purpose) of the text doesn't change when you apply a style any more than you change into a different person when you put on different clothes or change your hairstyle.

Let's return to the home page for your site, where you're doing the initial development work using the specifications that the client has provided. Your first task is to create the text block that will be used for page navigation as the site is built out.

- 1 Open `website-specs.txt` and `index.html`, if they are not already open. Set the view of the page to Split Code and Design.
- 2 From `website-specs.txt`, copy the block of text that describes the pages that the client wants to have in his site. You'll find this block of text on lines 22 to 28.
- 3 Switch to `index.html`, and place your cursor right after the exclamation point at the end of the tagline text. Press Enter/Return.
- 4 Right-click, and select Paste to paste the text into the document. The text in the top of the page should appear as follows:

```
Werewolves UnLimited Security Services
```

```
Let us protect your stuff with a snarling pack of crazed  
→ creatures!
```

```
Pages:
```

```
* Home page (With Ray's introduction)
```

- \* About Our Team (With pictures and "bios" of the company  
→ employees)
  - \* Services (With a list of the work the company will do)
  - \* Rates (I'll need a table with prices)
  - \* Contact (A way to send the fake company fake e-mail)
- 5 Delete the asterisk at the beginning of the lines and the parentheses and all the words inside the parentheses.
  - 6 Change the word Pages to **Navigation**.

You now have the basic site navigation content created. Now on to setting the format for this text. In the code for this text block, all the names are contained in a single paragraph tag (<p>) with a break tag (<br>) following each name.

```
<p> Home page<br>
  About Our Team <br>
  Services <br>
  Rates <br>
  Contact <br>
</p>
```

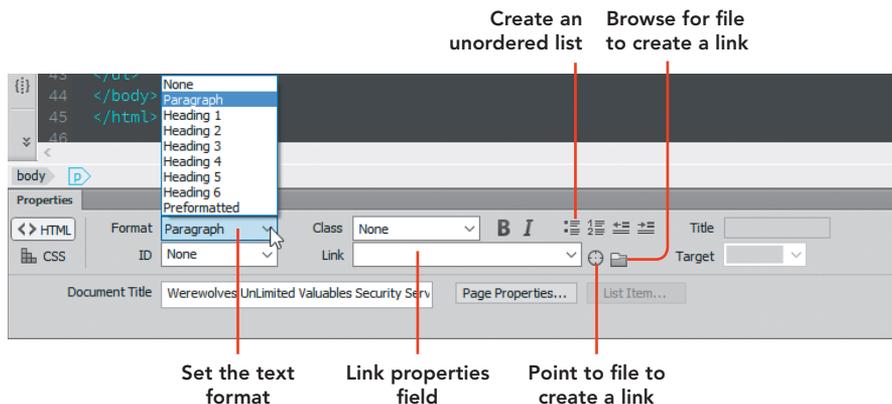
Break tags drop a line of text down a line, but they don't have any other inherent purpose. It's pretty common to get a lot of break tags when you copy and paste from a source other than Dreamweaver. With just a little bit of keyboard work, this can be quickly corrected.

- 7 Place your cursor in front of the letter A in **About Our Team**. Press the Backspace/Delete key to remove the break tag, and then press the Enter/Return key.  
  
Visually, you'll now see the text drop down two lines. But most importantly, the first word (Home) in the list is now enclosed in a <p> tag.
- 8 To complete the conversion of this code block, place your cursor in front of each line, press Backspace/Delete to delete the break, and then press Enter/Return to generate the <p> tag. Your code should appear as follows:

```
<p>Home page</p>
<p>About Our Team</p>
<p>Services</p>
<p>Rates </p>
<p>Contact </p>
```

Good job! You've added some needed purpose to this text. Now you need to convert the text into an unordered list that can be used for navigation. You'll use the Property inspector to do all the work for the next part of this project. Refer to **Figure 3.12** to complete these next tasks.

**Figure 3.12** The Property inspector with text selected



- 9 The first element to format on the page is the text that begins with the word *Welcome*. Select this entire line, then use the Format field in the Property inspector to change the formatting of this line of text to Heading 1.
- 10 Highlight the names of the pages in the navigation text block that were just converted to paragraphs. In the Property inspector, click the Unordered List button to convert the entire block to an unordered list, with each word contained in a list item tag (<li>). Your HTML should look like the following:

```
<ul>
  <li>Home page</li>
  <li>About Our Team</li>
  <li>Services</li>
  <li>Rates </li>
  <li>Contact </li>
</ul>
```

Don't worry about the bullets that appear next to the names of these pages. This is the default appearance for list items. Those bullets can be hidden or styled in different ways using CSS. You'll take that step when you start writing styles.

- 11 Now that you know how to convert a block of text into an unordered list, you can turn to the final block of text on the page. Following the procedure in the previous steps, convert the text marked with asterisks at the very bottom of the page into an unordered list.

## Creating Hyperlinks

The last stop in this project is to create the hyperlinks that will allow a visitor to navigate within the site. The text is now ready, and all you need to do is attach the link property to the names that are in the list. Refer to Figure 3.12 and locate the area of the Property inspector where links are listed and created. You'll start by creating the link to the home page of this site. Your view should be set to Split Code and Design.

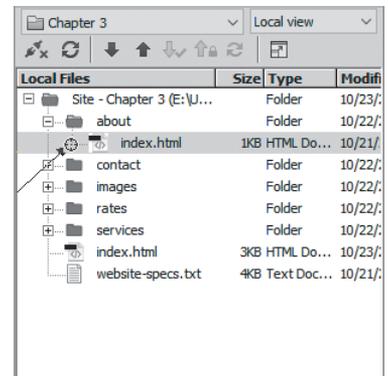
- 1 Highlight the words **Home page**, then turn to the Property inspector. Type **index.html** into the **Link** field, and press Enter/Return.  
Inserting the URL or filename and path in this area by typing or copying and pasting is the first way you can create a link in Dreamweaver.
- 2 Highlight **About Our Team** in the next line.
- 3 In the Files panel, expand the **about** folder so you can view the **index.html** page located in that folder.
- 4 Click the Point to File icon in the Property inspector, drag the marker onto the **index.html** file in the **about** folder Files panel, and release the mouse when the symbol you see in **Figure 3.13** appears.

You've now used the second method for creating a link.

- 5 Highlight **Services**, and click the Browse for File icon in the Property inspector.
- 6 When Windows Explorer or the Mac Finder opens, browse to the **index.html** file that's inside the **services** folder.
- 7 Select the file, and click OK/Open.

You have now used the third method for inserting a link in Dreamweaver.

- 8 Continue linking all the text in the navigation area using one of the three methods to create a link.



**Figure 3.13** The Point to File button in the Property inspector allows you to drag and drop a link onto a file in the Files panel.

9 Save your work.

The code block for this area of the page will appear as follows:

```
<ul>
  <li> <a href="index.html">Home page</a></li>
  <li><a href="about/index.html">About Our Team</a></li>
  <li><a href="services/index.html">Services</a></li>
  <li><a href="rates/index.html">Rates </a></li>
  <li><a href="contact/index.html">Contact </a></li>
</ul>
```

### FOLLOW THAT FILE!

When you look at the links you created in Project 3.3, you'll notice that the href value inside the quotation marks is written differently for the link to the home page versus the links to the pages inside folders.

When a link is made to a file within the same folder as the document, the path to the file is simply written with the filename. Since the home page is in the root folder for this site, a link to a page that's also in that folder will be written as `index.html`.

When a link points to a file inside a folder one level *down*, the path to the folder has to be spelled out, then a slash character, and *then* the filename. And so, a file inside the about folder is written as `about/index.html`. For a link that points *up*, the path is written with two dots and then a slash, like this: `../index.html`.

One of the great things that Dreamweaver does is maintain the correct file path for you, even when you move things around in the Files panel.

In this project, you've learned more about the purpose of text formatting and how text formats are set in Dreamweaver. Remember that text formats have a purpose, and it isn't to determine how they look. You've also used three methods for inserting a hyperlink in Dreamweaver.

## PROJECT 3.4

# Adding Content Containers and Attributes

The first web page in the site is starting to shape up. So far, you've added the text that the client has provided and set the formatting for the text so that the page has some structure. Now you'll continue with another common way that web designers get their web pages ready to be styled and laid out. In this project you're going to be adding the containers for this page and the "hooks" for the CSS styling rules that are yet to come. In essence, you'll be assembling the building blocks of the page design.

To create these building blocks, you'll use named `<div>` tags and tag attributes that are a snap to put into place. And there's good news. Your client has given you a mockup of how he wants the web page to look—or at least the beginnings of a design. From the client's initial ideas, you'll be able to determine how the containers of the page need to be built so that in the final construction process the design of the page can be set into place.

You have these tasks ahead of you to complete this project:

- Understand how **wireframe** diagrams are created based on client requirements.
- Learn how to break down the different content containers that are required for a typical web page.
- Wrap the existing content on the page in `<div>` tag containers.
- Understand the difference between ID and class attributes.
- Examine your code to be certain that the content on the page is properly placed in the correct container.

## Working with Wireframes

Since your client gave you a design composition (or **design comp**, as it's frequently called), you can now examine the design and prepare a wireframe diagram to show the building blocks that you'll need to lay out the page. Sure, this comp isn't all that close to how the final design should look, but at this point in the design process knowing how the page needs to be structured takes you farther down the path

★ ACA Objective 1.3

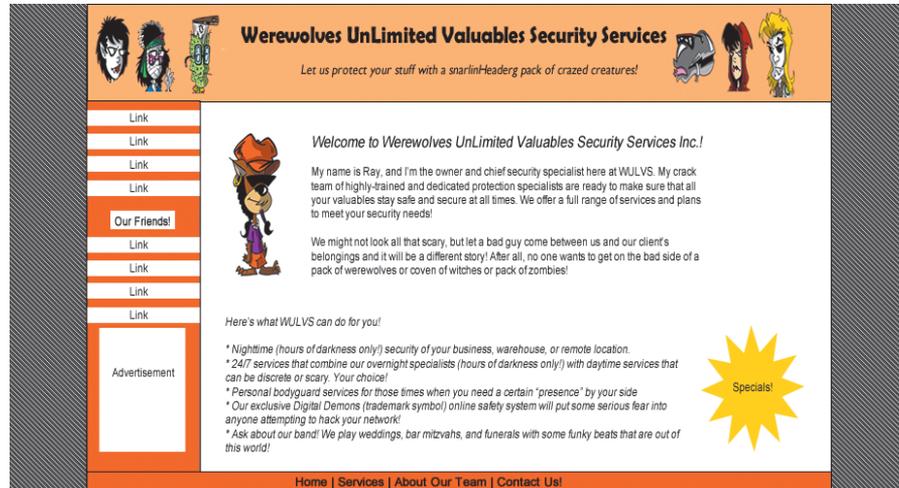
★ ACA Objective 3.2

★ ACA Objective 5.1

▶ **Video 3.4** Adding Content Containers and Attributes

of getting a final design done. You can see the client-supplied design for the site in **Figure 3.14**.

**Figure 3.14** An initial design comp for the cartoonist's website shows how the page might be constructed.



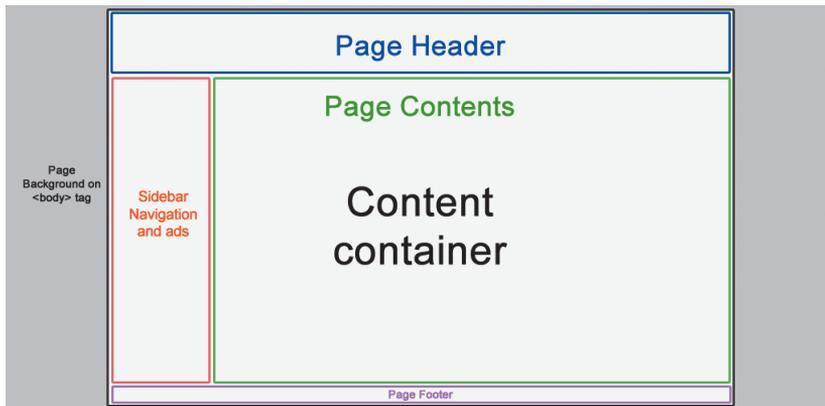
Wireframe diagrams are intended to show the architecture behind a design and are often simply sketched out freehand on a piece of paper. Wireframes usually go through many versions as the designer and the client nail down what's possible, what's in the budget, and how well your ideas match up to the client's expectations.

In the case of this design, if you translate the comp into the different boxes of content that will appear, you can see that the client's requirements can be turned into the wireframe pretty easily. This design requires:

- Content that "floats" over the top of a background
- The entire page centered on the screen, with margins on the left and right
- A header across the top of the page for the page name and some artwork
- A sidebar down the left side for navigation and advertising
- The main content area of the page with some images and the text for each page
- A footer where links and other information can be listed

Once those ideas are translated visually, you end up with a wireframe that might look like the one you see in **Figure 3.15**. You can even label the different containers to get an idea of how things are positioned on the page.

With the content containers identified, the next step for this project is to insert the named `<div>` tags that will be used for the final layout.



**Figure 3.15** Wireframe diagrams break down a design into the essential containers where the page contents will be located.

## Inserting <div> Tags in Dreamweaver

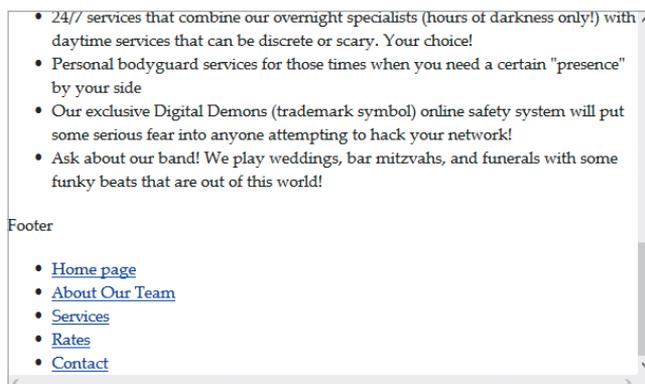
Dreamweaver makes inserting <div> tags a snap. Using the Insert panel, you can easily wrap the content that's already on the page in the building blocks that are needed to construct the page layout.

- 1 Open the `index.html` file that you've been working on in Dreamweaver.

When you compare the wireframe to the content that's already on the page, you'll see that the footer area of the page hasn't been included yet. Client change!

The footer for this design is just a copy of the same links that are over in the Navigation area, so you can easily duplicate the links.

- 2 Copy the list of links from the Navigation area, and paste them under the last item on the page. Type **Footer** as the name above the list of links, as you see in **Figure 3.16**.



**Figure 3.16** Copy and paste the unordered list of links to the bottom of the document to create the Footer area.

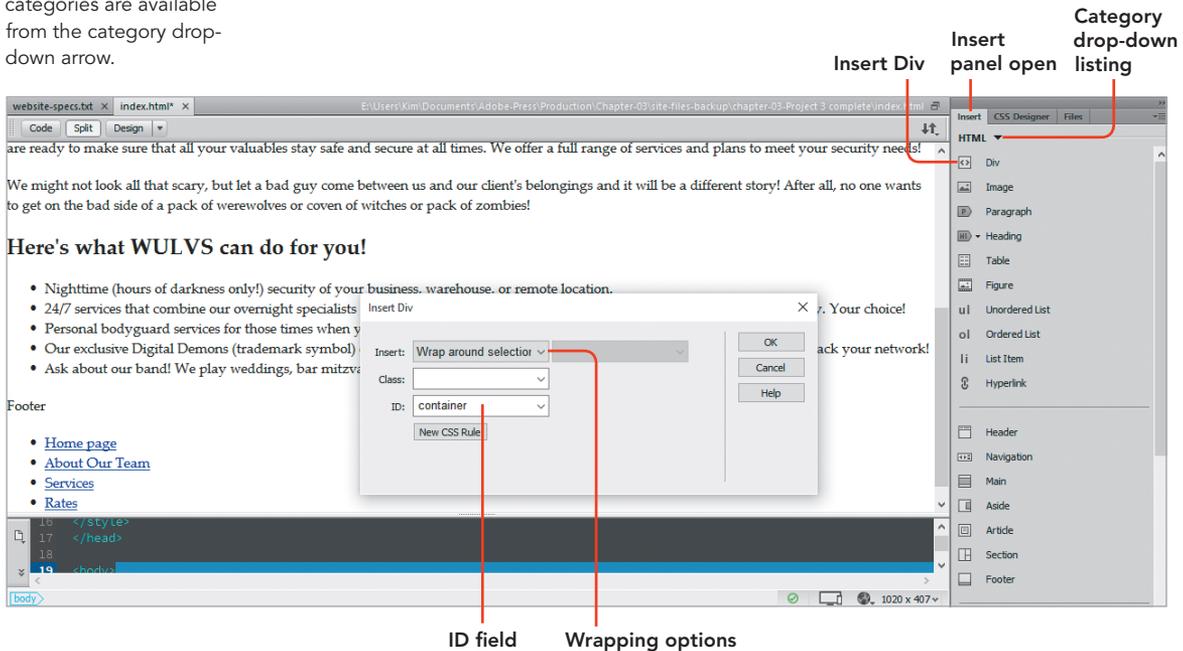
To complete the next steps, you'll use Dreamweaver's Insert panel. This panel is grouped at the top of the right column, next to the Files and CSS Designer panel.

**TIP** *If you don't see the Insert panel in the group as shown in Figure 3.17, you can open the panel by selecting Window > Insert.*

- 3 Click anywhere within the Document window, and press CTRL+A/Command+A. This will select all the content in the visible area of the page.
- 4 Click the **Div** button in the Insert panel (Figure 3.17). The Insert Div dialog will appear.
- 5 Set the Insert field at the top of the Insert Div dialog to Wrap around selection.
- 6 Click in the ID field, and enter **container** as the name of this div. Click OK when your settings match the ones you see in Figure 3.17. The <div> tag will be inserted into your code.

Dreamweaver has all sorts of ways to check your code to make sure the action you took wrote the code correctly. Remember that the goal for this <div> was to create a container that will appear inside the <body> tag.

**Figure 3.17** When the Insert panel is set to the HTML category, the Div button appears at the top of the list of items you can insert. Additional Insert categories are available from the category drop-down arrow.



- 7 Place your pointer next to the line where the `<div>` is located, and click the code collapse arrow (Figure 3.18).

It's easy to see that the container `<div>` is completely within the `<body>` tag. Nesting code like this allows for styling rules to be written that take advantage of the order of the tags in the document. Well done!

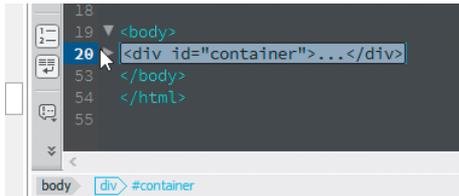


Figure 3.18 Collapsing and expanding code

## CLASSES OR IDS? WHAT'S THE `<DIV>`-ERENCE?

The `<div>` tag may accept an ID attribute, a class attribute, or even both. But what's the difference?

You can use classes as many times as needed to apply styling to an element on the page. For instance, it's common to write a CSS styling rule so that a class can be applied to images, making them float to the left, right, or center. You can use that styling rule over and over within the page by assigning the class to images.

IDs, on the other hand, can occur only *once* within a document. On this page there's only one content `<div>`, and no other `<div>` can be named `<div id="container">` on this page.

Now you're going to repeat this process of selecting text on the page and inserting a `<div>` tag to wrap around each selection. Ready? Real quickly then.

- 8 Select the company name and the tagline at the top of the page. Insert a `<div>` tag that wraps those two lines of text, and set the ID to **header**.
- 9 Select the navigation text and the list of links. Insert a `<div>`, and set the ID to **sidebar**.
- 10 Select the Heading 1 text and all the text below it until you get to the end of the list of services. Insert a `<div>` that contains this area, and set the ID to **content**.

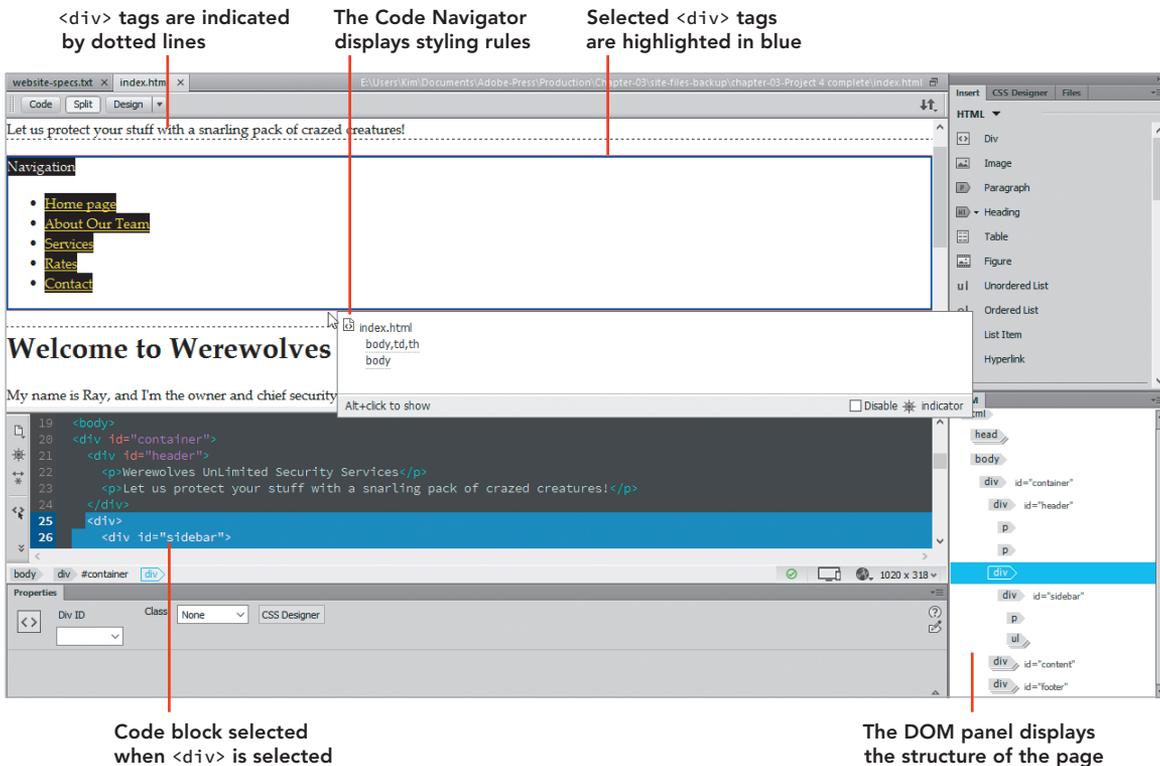
11 Select the footer area. Insert a `<div>`, and set the ID to **footer**.

Before you finish things up, this is a good time to carefully check your code using some of Dreamweaver's visualization tools. The DOM panel, for instance, is a great way to see how the containers you just inserted are nested within each other and to check for problems such as stray paragraphs and other code that doesn't belong. In the Design window, you'll see dotted lines around the `<div>` tags in the document, and if you select one of the dotted lines, Dreamweaver will place a blue border around the selected tag. You'll even be able to use the Code Navigator to have a look at any styling rules that apply to the selection.

12 Review your document, and compare it to the screen capture you see in **Figure 3.19**. Take care of any cleanup that's needed, and then save your file.

**Figure 3.19** Dreamweaver provides multiple ways for you to visualize the contents of a page and check for errors.

Congratulations! You've gotten the building blocks for this page all set. In the next project, all this groundwork will finally come together as your design begins to take shape.



## PROJECT 3.5

# From Blueprint to Assembly: Positioning Page Elements with CSS

The wulvs.com design shop has lots of excitement today. All the blueprints have been completed in the wireframe, Chris the Cartoonist has given you a better idea of how he wants the design to look, and most importantly, all the elements are baked into your code for the page using named content containers. Now it's time to experience the joy and wonder of laying out a modern web page using CSS positioning techniques. In this project, you'll complete the following tasks:

- Understand how the CSS box model is applied to elements within a design.
- Use the CSS Designer panel to add selectors to the style sheet.
- Apply margin and padding properties to content containers to elements on the page.
- Understand how float properties are applied to content containers to position elements.
- Convert an unordered list into a horizontal menu.
- Use Dreamweaver's tools to examine and modify styling rules in your document.

To begin, let's look at the CSS box model and how it affects positioned objects in an HTML document.

- ★ ACA Objective 2.2
- ★ ACA Objective 3.2
- ★ ACA Objective 5.1

 **Video 3.5** From Blueprint to Assembly: Positioning Page Elements with CSS

### DO NOT MISS THIS MOVIE!

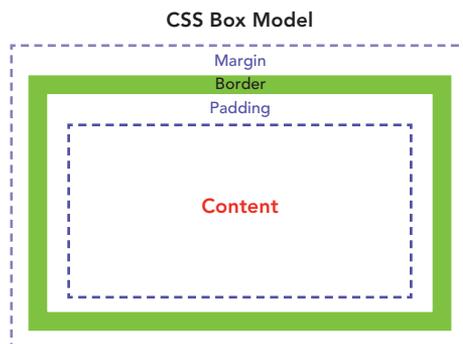
More than any other project in this book, this project really needs to be done after watching Video 3.5, "From Blueprint to Assembly: Positioning Page Elements with CSS."

Lots of the visualization tools that Dreamweaver provides are interactive, and you'll get a better feel for how they work if you see them in full-motion action. The video also gets you right down into the action as the layout of the page moves along.

## The CSS Box Model

Elements within a content container in a web page have certain values applied to them by default. Values for the margin, border, and padding wrap around the content within a container and have width values that you can apply to all four sides of the box. Although it might *appear* that content in a web page should be right up against another element, what you actually see in practice is the box model in action.

You should always keep in mind that every element within a web page lives within a box, and every one of those boxes contains values that might be hidden from plain view. Content inside a container is wrapped by invisible padding and margin properties. In between those two invisible elements lies the border, which may be styled with a color value and even a border style. Visually represented, the box model appears as you see in **Figure 3.20**.



**Figure 3.20** The CSS box model refers to the way that content in a web page is wrapped by padding, borders, and margins.

You'll witness some of the gotchas that the CSS box model can create when you lay out this page. So with that in the back of your mind, press ahead and start putting the building blocks for this design into position.

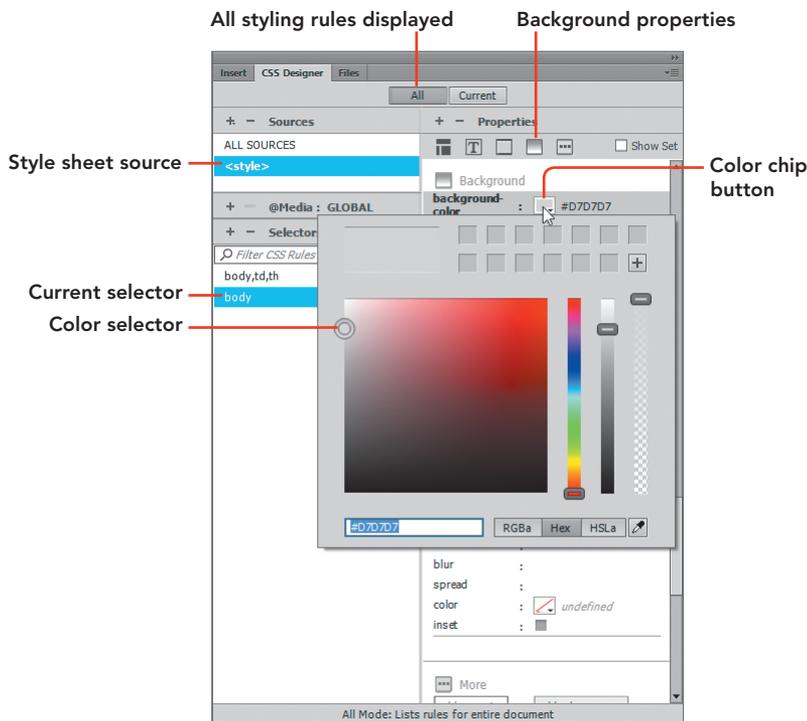
## Positioning Elements with CSS

You'll be using the CSS Designer panel to complete the steps that follow, but before starting you may wish to drag the left edge of the panel to the left so that you see two columns displayed. You'll see this configuration in the screen captures that follow.

## SETTING SOME BACKGROUND COLOR

It's helpful to visualize the layout of the page by assigning temporary background colors to the different containers. This will allow you to see any gaps that might appear and to check the position of the content blocks.

- 1 In the CSS Designer panel, click the **body** selector, as shown in **Figure 3.21**. Note that the view at the top of the panel is set to All and that the `<style>` tag is highlighted in the Sources area. The Show Set checkbox should be unselected.
- 2 Click the Background category at the top of the Properties section, as shown in **Figure 3.21**. Drag the color selector to the position you see indicated, and set the background of the `<body>` tag to a light gray.



**Figure 3.21** Applying a background color to a selector in the CSS Designer panel

## STYLING THE OUTERMOST <DIV>

Now you can begin to style the outermost <div> in this design—the #container div.

- 1 In the Code window, select <div id="container"> in the Code window.
- 2 In the CSS Designer panel, click the + next to the word Selectors to create a selector.

Dreamweaver will identify the selection on the page and automatically name the selector based on its attribute, as you see in **Figure 3.22**. Slick!

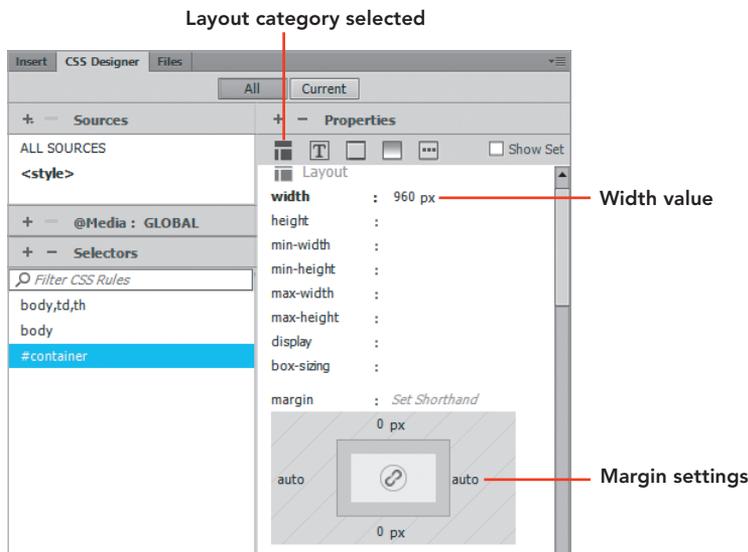
This container needs to “float” on top of the body tag and needs margins applied left and right. One way to achieve this is to set a width for the container and apply an auto value to the margins of the container.

- 3 In the CSS Designer panel, select the Layout category, and set the width value to **960 px**. Then set the left and right margins to **auto** in the margin section. Set the top and bottom margins to **0 px**. Use **Figure 3.23** as a guide.



**Figure 3.22**  
Dreamweaver automatically names the new selector.

**Figure 3.23** Use the CSS Designer panel to apply width and margin properties to the container <div>.



- 4 Switch to the Background category, and set the Color value to white using the same process you followed in step 2 of the previous exercise.
- 5 Preview this page using the Preview in Browser button located in the lower-right corner of the Document window. If you have unsaved changes, you'll be prompted to save the file before you preview.

You'll see that the container <div> is floating above the page and that margins appear to the left and right. Great job!

## STYLING THE HEADER <DIV>

The next <div> to style is the header.

- 1 Select <div id="header"> in the Code window, and then click the + in the Selectors section to create a selector for the header. If your code is still collapsed from your previous exploration, you may need to expand the code to see this tag.

You'll see that in this case Dreamweaver creates a *descendent* selector by putting the #container selector in front of the #header selector. You can make these settings more or less specific by pressing the Up Arrow key on your keyboard. The only selector you need in this case is #header.

- 2 Press the Up Arrow key until only the #header value remains, and press Enter/Return.
- 3 Switch to the Text category in the CSS Designer panel, and set the text-align value to center using the button located next to the name of the property.
- 4 Select <div id="sidebar"> in the Code window.
- 5 Click the + to create a selector. Press the Up Arrow key on your keyboard to change the value of the selector until only the #sidebar name remains, and accept the name.
- 6 In the Layout category, apply the following values to the #sidebar styling rule to set the width of the box and move it to the left side of the screen:

Width: **180 px**

Float: **left**

- 7 Select the line of code containing the <div> named content and create a new selector with the name set to #content.

The goal here is to set the width of this container so that it snugs up against the sidebar. Since you have a 960 px container <div> and the sidebar <div> is 180 px, the content <div> *should* be 780 px, right? But not when a little padding is added to the left side of the container. When that happens, the CSS box model comes into play, so the content actually slides underneath the sidebar. But you can adjust that easily enough by subtracting the 10 px of padding from the formula to get a width value of 770 px.

- 8 Apply the following values to the #content styling rule in the Layout category:

width: **770 px**

float: **left**

padding-left: **10 px**

When you finish creating that styling rule, the content <div> will float up against the sidebar, and you'll be good to go.

## STYLING THE FOOTER <DIV>

The final content container to address is the footer <div>. This container needs to have a special float property applied to it so that the sidebar and content containers stay put and aren't allowed to move next to the footer.

- 1 Select <div id="footer"> in the Code window, and create a selector named **#footer**.
- 2 Apply the following values to the **#footer** styling rule in the Layout category to clear out the previous floats in the page and make the text center aligned:  
text-align: **center**  
clear: **both**

Now is a good time to temporarily assign background color values to the containers on the page so you can see how things lay out and if there are any gaps that need to be addressed.

- 3 Return to the selectors for the **#header**, **#sidebar**, and **#footer**, and set a background color of your choice in the Background category.

## FIXING THE GAPS

If you preview your design in a browser now, you'll see that there are indeed two issues with this page. There's a margin at the top of the page and a margin below the header.

These gaps (Figure 3.24) are caused by the default margin properties that the paragraph tags in the header contain. A great way to see these hidden margins is by switching to Dreamweaver's Live view and turning on Inspect mode. With that little button selected, Dreamweaver will display hidden margin and padding properties as you float your mouse over the different containers on the page. To correct those margins, you'll need to add some new attributes to the paragraphs within the header.

- 1 Select the line of text at the very top of the document that contains the company name. In the Property inspector, set the ID field value of the paragraph to **header-name**.
- 2 Create a selector for **#header-name**, and assign the value **margin-top: 0px**.
- 3 Select the next paragraph in the header area—the tagline—and set the ID of the paragraph to **header-tagline**.
- 4 Create a selector for **#header-tagline**, and assign the value **margin-bottom: 0px**.



**Figure 3.24** Dreamweaver provides multiple ways for you to visualize the contents of a page and check for errors.

Once this action is complete you'll find that the margins that were at the top of the page have disappeared. Great work.

## STYLING THE LINKS IN THE FOOTER

The final bit of work that needs to be done is in the footer area. The list of links in this part of the page need to be displayed side by side, and you certainly don't want those bullets to appear. Let's take care of those things with a few more styling rules.

- 1 Locate and select the `<ul>` tag that's inside the footer container. Add a selector using the descendent selector. Dreamweaver suggests names for selectors based on the element that's selected. For this selector, set the name to **#footer ul**.

- 2 In the Text category, set **list-style-type: none** as the value for the text in this list.

The bullets will magically disappear.

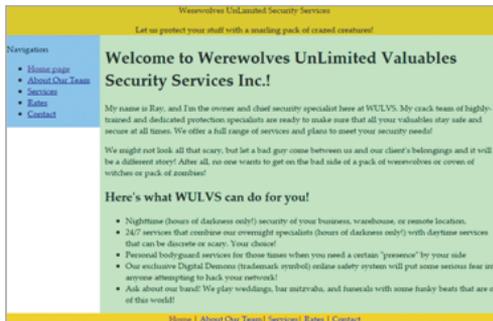
- 3 Click the first `<li>` tag in the list, and click the + in the Selectors area. Accept the suggested new selector name: **#footer ul li**.

- 4 In the Layout category, apply the value **display: inline** to have the text appear side by side.

More magic! The list of links shrinks down to a single line, and the footer area now appears as it was envisioned. Once again, well done.

Hooray! You made it!

- 5 Save your work then preview your design in a browser and make note of any issues that appear. After you compare your design to the one you see in **Figure 3.25**, save your file, and take a well-deserved break.



**Figure 3.25** The final page layout as viewed in a browser window

# Challenge! Create Your Own Page Comps

You've learned a lot about page layouts and the building blocks of HTML in this chapter. Now it's time to think about how you might go about creating your own designs. What you know about web design and how pages are constructed should give you an appreciation of how to go about creating (in your favorite graphics editor) new designs that can be translated into HTML and CSS.

Whether you use Photoshop or Fireworks or even Illustrator for designing your page comps, there are a few things to keep in mind that have to do with the available canvas sizes and other measurements that you'll want to use. As you start getting creative, keep these facts in mind.

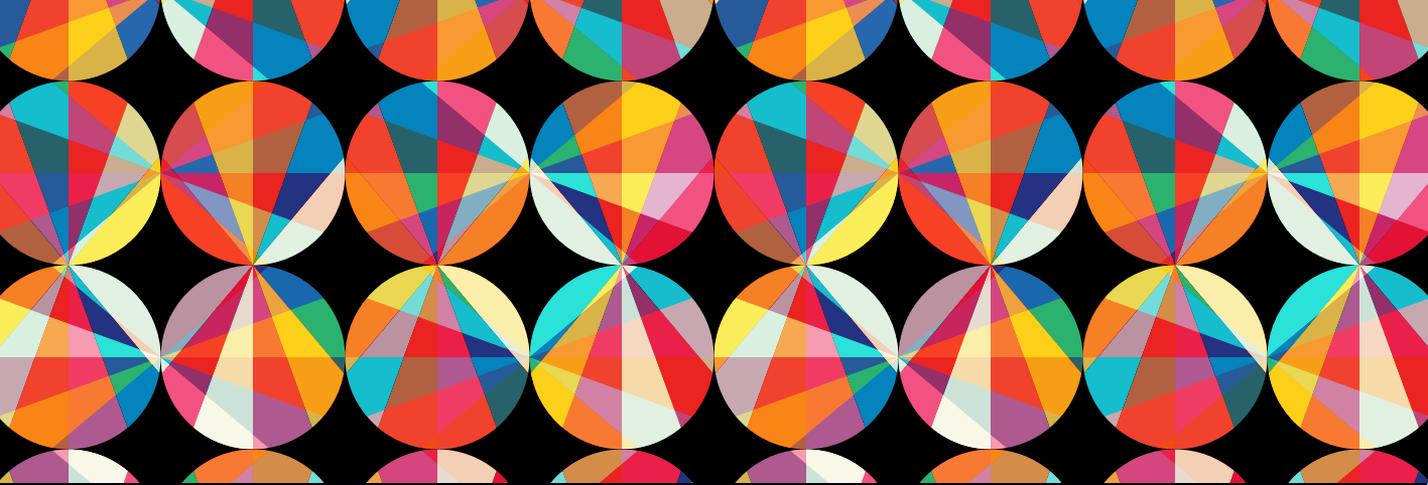
Most web designers design for a default monitor resolution of 1024 x 768 pixels. But that's only half the story. When you subtract for scroll bars along the sides and browser buttons across the top, the actual viewable area is approximately 1000 x 475 pixels. That's all the space you have to work with before the viewer starts needing to scroll to see additional content.

Where you want to have margins on the page, you'll need to reduce the area of the page where the content sits to an even smaller size. The size that was used in the final project here (980 pixels) is a good one to begin with.

Finally, it's awfully common these days for page comps to be designed for multiple screen sizes. You might have one design for the page as it will be seen on a desktop computer, another comp for a tablet view, and a third comp for the design as it would appear on a phone. Online, you can find free blank templates that show the various sizes that are used in design comps, and you can use those templates when you open Photoshop and get to work on your own unique creation.

# Conclusion

When you think back on all you've done in this chapter, you have to be feeling pretty good about what you can now do with Dreamweaver. Along the way to learning about how text is formatted on the page, how links are inserted, and ultimately how a page layout is designed using named content containers with CSS, you've used many of Dreamweaver's tools for visualizing and modifying the code that makes up your new web pages. In one (kind of) short chapter, you've built out a complete website with the folders and files that are needed for the site to function and be easy to use, and you've even done some detective work in figuring out what a client wants and translated those clues into code. You are well on your way to getting a handle on the amazing things you can do when code comes under your command.



# CHAPTER OBJECTIVES

## Chapter Learning Objectives

- Understand how the client-to-design process works for creating page prototypes that meet the customer's requirements.
- Create CSS styling rules that make use of background images.
- Use round-trip editing to insert a source Photoshop or Fireworks image file and edit the file from within Adobe Dreamweaver.
- Insert images on a web page and position them using CSS floats.
- Learn the location and purpose of Dreamweaver's image editing tools and the recommended methods for their use.
- Use Dreamweaver's visualization tools to troubleshoot designs and align elements on the page.

## Chapter ACA Objectives

### DOMAIN 1.0

#### SETTING PROJECT REQUIREMENTS

- 1.2** Summarize how designers make decisions about the type of content to include in a web authoring project, including considerations such as copyright, project fit, permissions, and licensing.

### DOMAIN 2.0

#### UNDERSTANDING WEB AUTHORING

- 2.4** Demonstrate knowledge of Cascading Style Sheets (CSS) and the behavior of basic CSS tags.
- 2.6** Demonstrate knowledge of digital images, text, interactive elements, animation, and video and audio content as they pertain to web authoring.

### DOMAIN 4.0

#### WEB AUTHORING USING ADOBE DREAMWEAVER CC

- 4.1** Define a new Dreamweaver site.
- 4.2** Create a web page and define the page layout using HTML, CSS, and div tags.

## CHAPTER 4

# Getting the Big Picture

Human beings are incredibly visual creatures. Our brains can process more information when we view a picture or illustration than we can by reading a long passage of text. The old saying “A picture is worth a thousand words” is certainly true.

Images and visuals are a huge part of the modern web experience, and in a world where you can snap a photo on your phone and instantly publish it online, visitors to a web page expect to see lots of images. And just like graphics, images can be used to define the layout of web pages and assist in the usability of the site—making it easier for visitors to find their way around and get to the information they’re after.

In this chapter, you’ll take a look at the ways you can use images to further define the look and feel of a design, how to insert and position images, and the tools that Dreamweaver provides to help you get those jobs done.

## Bring On the Art!

Great news! Your client Chris the Cartoonist (**Figure 4.1**) has provided a *ton* of artwork for the *WULVS.com* website, and he's also given you excellent guidance on how he wants the design to proceed.

**Chris Sez:**

*I really like the layout of the page! Those colors are kind of weird, but I know that's just temporary, right? Let me tell you a little more about the storyline and the characters in my made-up world.*

*WULVS is a comic strip built around a security service run by Ray the Werewolf.*

*In Ray's world, he works with a group of vampires, zombies, witches, mummies, aliens, and other characters who work as guards for hire by the "regular" humans of the world.*

*Part of the fun of the comic strip is how Ray runs his business—and his website—just like a regular business. After all, a monster has to make a living too. What I want to do is use the artwork of the characters in the strip to create Ray's company website, *wulvs.com*.*

*I'm going to break down what I want for each page and put it into a separate document. OK? I'm looking forward to seeing your work!*

Lucky for you, he's happy with the layout that was made from the wireframes in the last chapter, and now you can take the big step of designing the pages the way Chris wants. Well, he isn't sure exactly what he wants quite yet, but that just means that you can put on your designer hat and start working on a prototype for this site.

To get started on all the new elements for this design, you'll need to download the files that have been provided and define your Chapter 4 website.

## Get Your Files Here

As you did in previous chapters, copy the folder that contains this chapter's files to your computer.

- 1 Download the project files for this lesson, named `chapter-04.zip`, from the Lesson & Update Files tab on your Account page at [www.peachpit.com](http://www.peachpit.com), and store them on your computer in a convenient location.
- 2 Unzip the files and copy the `chapter-04` folder to the `Learn-Dreamweaver` folder you created in Chapter 1.



**Figure 4.1** Chris the Cartoonist

# Define the Chapter 4 Website

With the files you've been provided for this chapter in the project folder, you're ready to set up the site in Dreamweaver.

★ ACA Objective 4.1

**NOTE** Refer to Video 1.5, "Defining a Dreamweaver Website," if you want a quick demo of how the site definition process works.

- 1 From the application bar, select Site > New Site.
- 2 Name the site **Chapter 4**.
- 3 Browse to the **chapter-04** folder in your **Learn-Dreamweaver** project folder.
- 4 Click Save.

## Image File Formats for the Web

Deciding on the right kind of image to include in a web page goes beyond simply getting the best picture available and sticking it on a page—although making that choice can be a huge chore in itself. You also need to understand which types of files will be displayed properly by the viewer's browser and the type that is most appropriate for the kind of image you are adding.

★ ACA Objective 2.6

At the most basic level, it's pretty simple. Web browsers can display three file types:

- **GIF** (Graphics Interchange Format) images have filenames that end with the .gif file extension. GIFs can display a limited number of colors, which makes them very efficient. GIFs are a great choice for graphics—such as logos, illustrations, navigation bars, and other design elements—that don't require colors with subtle shades. GIFs can have a transparent background, and of course you can animate GIFs to create miniature little movies. How else would we be able to see cats playing pianos?
- **JPEG** (Joint Photographic Experts Group) files use the .jpg or .jpeg extension, and as you can probably guess, this format is best used for photographs and other images with thousands or millions of colors. JPEG images can be compressed by changing quality settings in an image editor to get smaller file sizes. Web designers often compare different quality settings in Photoshop,

for instance, until they achieve a good balance between how good the image looks and its file size. JPEGs cannot have transparent areas and cannot be animated.

- **PNG** (Portable Network Graphics) is the newest image format. This file type was specifically developed for use on the web. The cartoon character images you've been provided all use the PNG format because images in this format can display transparency, sharp edges, and color changes—like a GIF—but still allow for subtle shades of colors like a JPEG. PNG files are by far the most common file type used by modern web designers.

All these file types are in the family of images known as **bitmaps**. In a bitmap, every color in the image is defined as a tiny block of data with a color value and other properties. If you've ever zoomed way in on a digital image, you've seen those individual blocks magnified and seen how the image is composed.

Every image in a web page must be downloaded from a remote server somewhere, which makes image **file weight**—the size of the file as measured by how much data it holds—almost as important as how it looks. Large image files are one of the primary culprits when a page loads slowly—for example, inserting an image that weighs 500 KB versus one that weighs 100 KB.

This process of choosing the right file type and adjusting image properties to get the smallest file size possible is known as **optimization**. Adobe makes two fine software programs for editing and optimizing images: Photoshop and Fireworks. In fact, the best method for working with images is to compose and edit the images in a graphics editor, then export a version that has been optimized for viewing on the web.

Whether or not you're optimizing your images or you're working with a graphic designer who will prepare the images while you take on the coding duties, you do need to know the kind of images that are appropriate for different applications. Smart designers *always* work with a source Photoshop or Fireworks file and *never* work directly on an image file or photograph without having a backup.

## PROJECT 4.1

# Prototyping and Designing with Background Images

Chris has been hard at work and has provided all sorts of new information to work with. In the project files for Chapter 4, Chris has supplied a bunch of images and specific feedback on how things should look going forward. Based on the instructions and the materials he has provided, you can start getting content into place.

★ ACA Objective 1.2

★ ACA Objective 2.4

★ ACA Objective 4.2

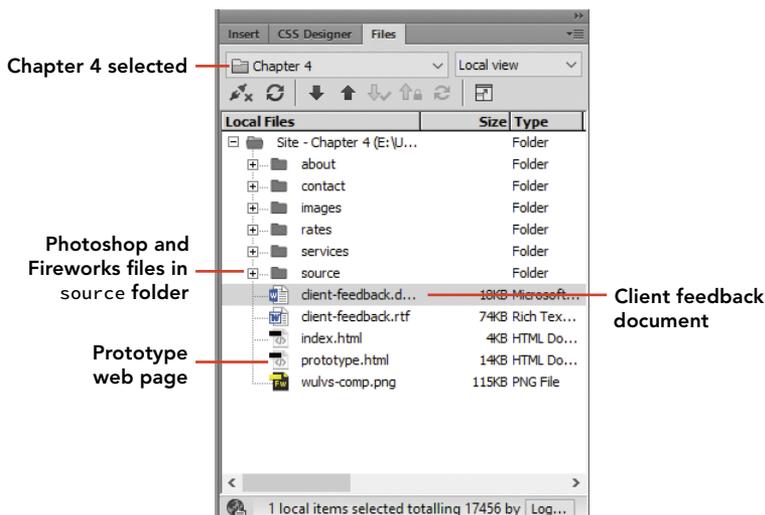
## Examining Client Feedback and Requirements

Take a few moments to familiarize yourself with the new files and folders that Chris provided.

- 1 Open Dreamweaver and review the structure of the site and the new folders and images.

The folders and files you created in Chapter 3 are still in place, but new images and documents have been supplied by the client (**Figure 4.2**). The `images` folder contains lots of new cartoon characters, and the `source` folder contains Photoshop and Fireworks files. Also included are two client feedback files.

▶ **Video 4.1**  
*Prototyping and Designing with Background Images*

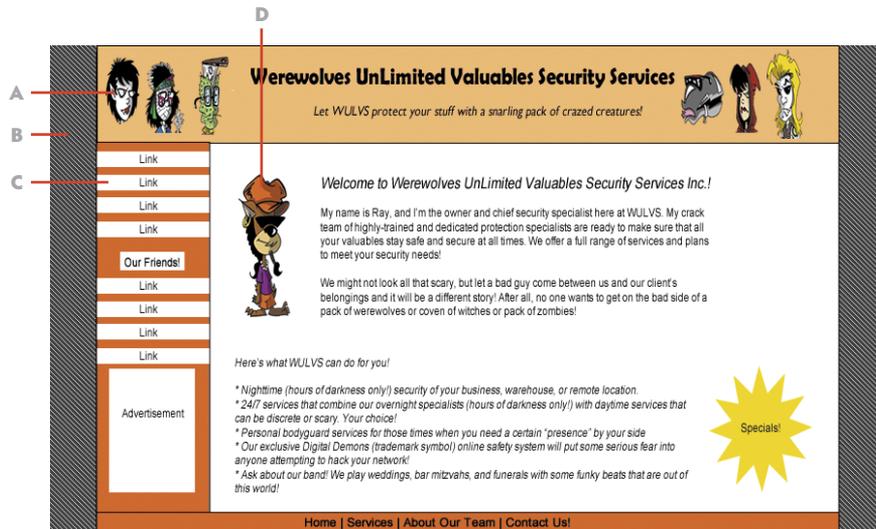


**Figure 4.2** The files and folders for the Chapter 4 website

- 2 Open `client-feedback.docx` by double-clicking the file in Dreamweaver. The file opens in Microsoft Word. If you don't have Word installed on your computer, you can open the `client-feedback.rtf` file in almost any word processor.
- 3 Read through the client feedback from the client. Deciphering information from a client frequently involves a lot of detective work. What specific guidance does the client provide? What things are being left up to you as the designer? And most importantly at this stage, can you determine enough information to get a *feel* for how things should look?
- 4 Open the client's design comp by double-clicking `wulvs-comp.png`. This file opens in the default graphics editor on your computer (Figure 4.3).

**Figure 4.3** Major design elements in the design comp provided by the client

- A** Header with character collage
- B** Page background with textured image
- C** Sidebar and navigation area
- D** Character inserted into the page



The design comp includes the major design elements that the client is requesting. The page is centered in the viewport and has a header, sidebar, footer, and main content area. You'll also see design clues as to how the client wants images to be used. This is another good time to jot down some notes on what you discover in this document.

- 5 Now open `index.html`. This is the same file that you finished at the end of Chapter 3. Compare this file to the client's composition to see the differences.

## 6 Open `prototype.html`.

This is a file that you might have developed based on your detective work of the information your client provided. In the interest of time, the following tasks have been done for you:

- The document includes the content that will appear in several pages of the website. You'll find new Heading 1 tag sections for the Team, Services, Rates, and Contact pages.
- The Team section has character descriptions, the Services and Rates sections have tables of information, and the Contact section has a spot for a form.
- The sidebar area has text for two navigation sections and placeholder text for where advertising might go.
- New CSS styling rules have been created to assign background colors and borders to the header and footer areas of the page.

## 7 Open the CSS Designer panel. Select the Show Set checkbox in the upper-right corner of the panel, and examine the properties that have been applied to the `#header` and `#footer` rules in this document.

Once you've thoroughly reviewed all the changes that have been made and compared those changes to the `index.html` file, you are ready to move on to more styling. You can even close `index.html`. All the work you do from this point forward will be done in `prototype.html`.

# Prototyping in Web Design

Working with a technical prototype is a very common method used by designers as they reach this stage in designing a website. By using a single document like the one you'll work on in this chapter, all the individual images, text, tables, and other elements can be placed and styled into a single page. The CSS styling rules remain in the `<head>` of the page while this work is going on, making it easier to keep track of the rules as you create, troubleshoot, and modify styles. When the styling is complete and the client has given final approval to move forward, the CSS file will be converted into a separate, external file, and the contents of each section can be copied to the pages where they'll ultimately reside.

This kind of workflow also allows you as the designer to take advantage of Dreamweaver's automation features as final designs are converted to templates and library items. These templates and library items make it easier to generate new pages and update an entire site full of web pages when a revision takes place.

## LEVEL UP: SMART WORKFLOW

One of the best habits you can have as a programmer is to incrementally back up your files as you work. Throughout the course of this chapter, you'll be working on a single file: `prototype.html`. You should save a version of the file as you achieve each milestone in the design process. To do this, select `File > Save As` to make a copy of the file as you go along. For instance, at the conclusion of this project you might save this file as `prototype-project-4-1-complete.html`. You'll continue to work on `prototype.html`, but in case some tragic mishap occurs you can open a previous version of the file to recover your work.

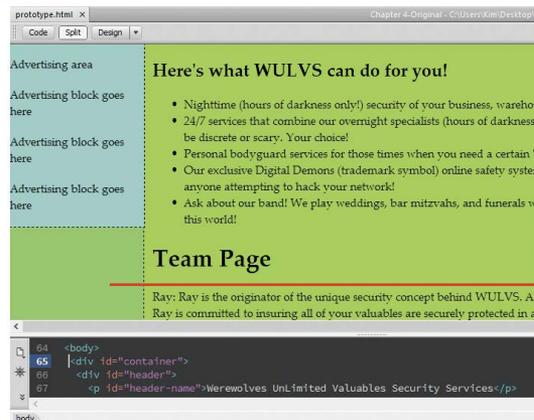
## Using the Faux Column Technique to Style the Sidebar

The client's design calls for a sidebar to extend down the left side of the page. That's a very common arrangement, but it presents a challenge: How does a web designer style this area of the page in such a way that it appears correctly no matter how long the web page is? Let's understand the problem, then see a common solution.

- 1 With `prototype.html` open, place Dreamweaver in Design view. From the application bar, choose `View > Visual Aids > CSS Layout Backgrounds`. You can enable and disable this view as you examine the page.

Dreamweaver will apply color coding to help you visualize the `<div>` elements of the page (Figure 4.4).

**Figure 4.4** The Design window with CSS Layout Backgrounds enabled



Sidebar content does not extend down the page

Notice how the main area content is longer than the sidebar content. You could insert your cursor into the sidebar and press Enter/Return to add a bunch of empty paragraphs until the sidebar is as full as the content area, but that's a very sloppy way to deal with the issue. Instead you will use a time-tested method of styling with a background image. This method is known as the *faux column* technique.

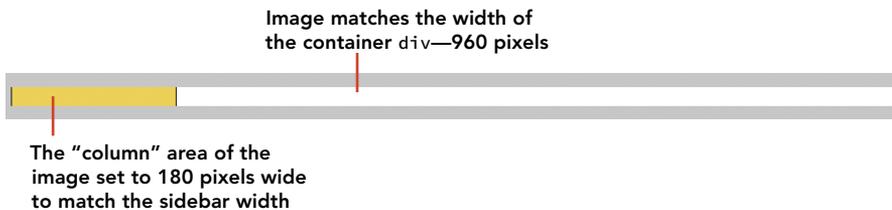
- 2 Double-click the `container-bg-gold.png` file inside the `images` folder to open it.

The faux column technique is perfect for a fixed-width page like the one you're working on. To create the illusion that the left column is filled up, a background image that matches the width of the container is used for the background. In this case, the `#container <div>` is set at 960 pixels wide.

The `#sidebar <div>` is 180 pixels wide. The area of the image that will provide the background color or pattern needs to be set to a matching width.

**Figure 4.5** details how this image is laid out.

Notice that the image is only 20 pixels high. A background image like this one doesn't need to be any larger, because it will repeat vertically inside its container.



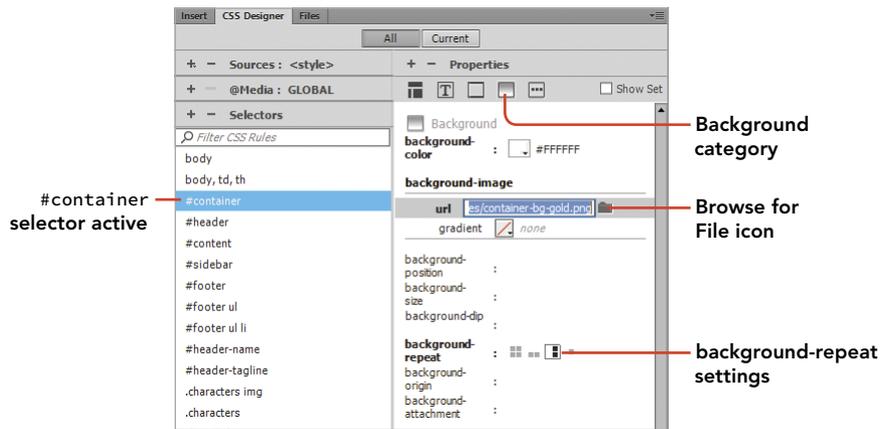
**Figure 4.5** An image for use as a faux column

- 3 Open the CSS Designer panel, and select the `#container` selector.
- 4 In the Background section, locate the background-image property, click inside the field next to url, and then click the Browse for File icon to locate the image.
- 5 Browse to the `chapter-04/images` folder, and select `container-bg-gold.png`. Click OK/Open to set the image as the background.
- 6 To control how the image repeats, or tiles, down the container, set the background-repeat property to `repeat-y`. This will restrict the image to repeating only down the page and not across.

**NOTE**

*You'll actually have to click the folder icon beside the url field twice in many cases. This might be intentional or just one of those little quirks that software programs have sometimes.*

**Figure 4.6** The CSS Designer panel set to the Background category



7 Compare your settings to **Figure 4.6**.

If everything looks good, it's time to preview your work in your favorite web browser.

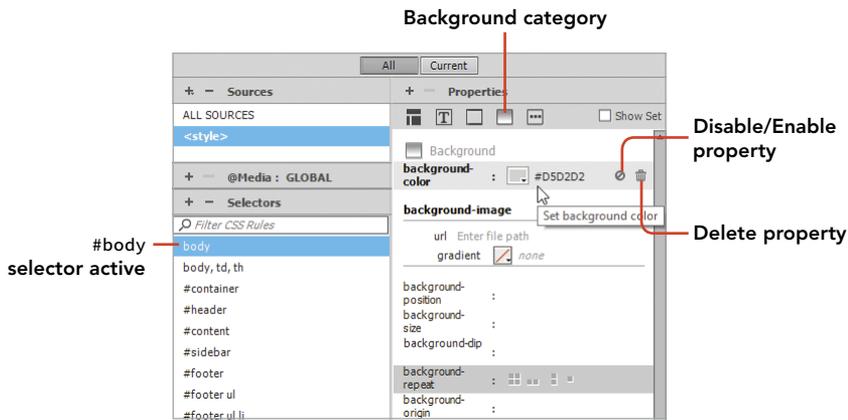
## Using Images in Page Backgrounds

The second client request to address is for an image that displays on either side of the content area of the web page. Chris has provided several different images for you to try. You might not actually use these in the final design, but it's a good opportunity to see the different background-repeat settings that CSS provides and to see how images may be used to style the entire visible window in a web browser display.

This process uses the same tools and techniques you just used on the `#container` selector, except this time you'll work with the `body` selector.

- 1 Open the CSS Designer panel, and select the `body` selector. This selector was created previously to “zero out” the page margins and to set a temporary background color.
- 2 Click the Background category, and float your mouse over the background color setting, as you see in **Figure 4.7**.

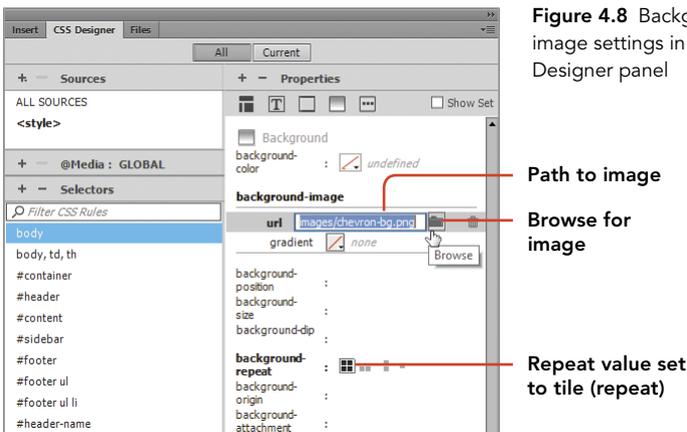
You'll see two buttons appear to the right: a button that allows you to disable a setting temporarily and a trashcan icon for deleting a property.



**Figure 4.7** Background color settings in the CSS Designer panel

- 3 Click the trashcan icon to delete the background color setting.  
Web designers frequently use tiny images set to repeat across and down the page to create a wallpaper effect.
- 4 Click the Browse icon next to the background-image url field.
- 5 Browse to the `chevron_bg.png` file in the `images` folder. Select the file, and click OK/Open.  
The background-image setting will appear (**Figure 4.8**).
- 6 Set the background-repeat option to tile across and down.

**NOTE** *The default behavior for background images is to tile across and down the container where they are set. Although it isn't necessarily required, if you want this default behavior it's good practice to explicitly set that value.*



**Figure 4.8** Background image settings in the CSS Designer panel

- 7 Click the Preview in Browser button (  ) in the lower-right corner of the document window. When prompted, choose to save and view the file in your favorite browser.

You'll see this one small image tile across and down the page.

- 8 Click the Browse icon in the background-image settings area, and switch the file to `diamond-bg.png`. Preview your page in a browser again to see tiled background images in action.

The design in **Figure 4.9** might not make the final cut, but you should have a good idea of how this kind of image can be incorporated into a design.

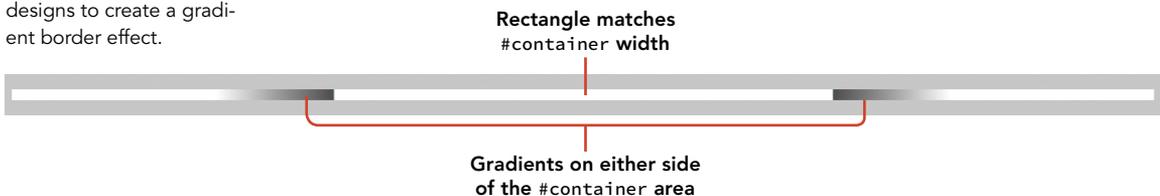


**Figure 4.9** Background images are tiled inside their container when set to repeat.

Another common background image design technique is to use a wide image with gradients or other visual properties set on either side of a blank area of the canvas.

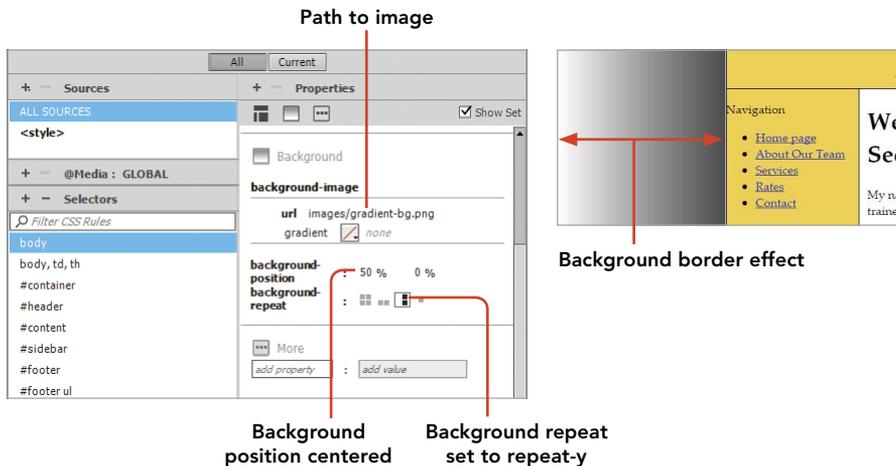
- 9 From the source folder, open either `gradient-bg.psd` (a Photoshop file) or `gradient-bg.fw.png` (a Fireworks file) (**Figure 4.10**). Note that this image is quite wide—2200 pixels—but only 20 pixels high. You'll also see that the center portion of the canvas has been filled with a white rectangle that matches the width of the `#container <div>`.
- 10 Return to Dreamweaver, and use the Browse for File icon in the background-image setting to switch the background image to `gradient-bg.png`.
- 11 Set the background-position setting to **50% left** to place this image in the center of the viewport. You can drag your mouse to apply this setting, or click inside the value field and type the value.

**Figure 4.10** Background images are used in some designs to create a gradient border effect.



12 Click the repeat-y icon in the background-repeat area, and compare your settings to those in **Figure 4.11**.

When you preview the page in a browser, or switch to Live view, you'll see that the area on either side of the page contents has a gradient background that fades to white. This is another way the client's requirements might be met using background images.



**Figure 4.11** The effect of using a background image that repeats vertically down the page

In this project, you've learned how you can use background images as a layout technique where a column on the page is visually defined using a background image, and you also have seen how the entire viewable area around the page contents can be styled by applying a background to the `<body>` tag.

### LEVEL UP: YOUR OWN BACKGROUND IMAGES

Now that you know how to use background image settings, you can modify the properties of the CSS styling rule that is applied to the header area with the ID `#header`.

Select the `#header` rule in the CSS Designer panel, and apply one of the background header images the client has provided.

Looks like Chris is still going with sort of a hipster vibe. The files named `argyle-bg.png` and `diamond-bg.png` are two examples you might use. Or you can work a background image into your own design comp and use your own file.

## PROJECT 4.2

# Inserting and Editing Images

★ ACA Objective 1.2

★ ACA Objective 2.4

★ ACA Objective 2.6

★ ACA Objective 4.2

▶ **Video 4.2** *Inserting and Editing Images*

You accomplished a lot in the first project. You've gotten some important page elements all set using background images and CSS styling rules to style the content areas of the page. Did you take the Level Up challenge? If you did, your header should also have a nice background image applied to it.

Now it's time to turn your attention to inserting images into the body of the page. Luckily for you, Chris has provided all sorts of images for you, and he's even optimized them and set the file types of the images properly so you can jump right into inserting the images. In this project you'll use the tools that Dreamweaver provides for working with images in a website, try the editing options that are available, and follow some recommended practices for modifying images.

From your review of the design comp and the instructions the client provided, you know you have a lot of images to insert into the page.

### LEVEL UP: DREAMWEAVER PREFERENCES

Dreamweaver allows you to set numerous individual preference settings that determine how the software works for you. You can determine which browsers you want to have available for previewing web pages, change the colors of your Code window, and set the font size used in the Code window, to name a few.

You can also set your preference for which image editor will open when you are using the image editing tools while working on a document.

- 1 Open Dreamweaver preferences by choosing Edit (Dreamweaver CC) > Preferences.
- 2 Select the File Types/Editors category in the left column. You'll see the three image file types available for use in web pages (GIF, JPEG, PNG), and you can select which editor to use.
- 3 Choose the editor you wish to use in the right column, and click Close.

Photoshop is by far the most popular editor to use and is the one most likely to appear on your computer if you have the Creative Cloud suite installed. Many people also like to use Adobe Fireworks CS6, which is still available as part of a Creative Cloud license but is no longer under active development by Adobe.

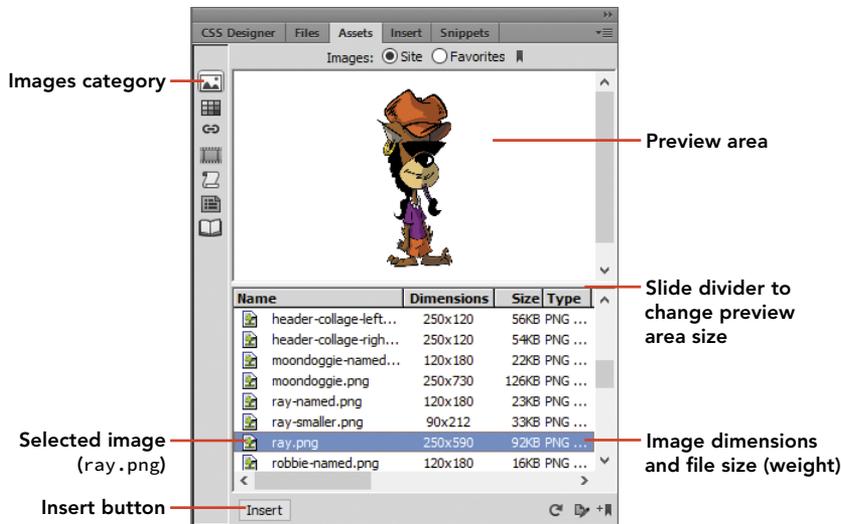
## Insert an Image Using the Assets Panel

The Assets panel is a great tool for situations in which you have many available images and want to identify which ones to use.

- 1 If the Assets panel is not already open, select Window > Assets. By default, the panel appears in the same panel group as the Files and CSS Designer panels.

The Assets panel makes it easy to sort through all the images in your site and displays a preview of the images at the top of the panel. You can slide the divider up or down between the preview area and the alphabetical list of files to see a smaller or larger preview. **Figure 4.12** shows the functional areas of this panel.

- 2 Scroll through the images to locate `ray.png`, and select it.
- 3 Place your cursor in front of Ray's introduction to the site (*My name is Ray...*), and click the Insert button in the Assets panel to place `ray.png` onto the page.



**Figure 4.12** The Assets panel with the Images category selected

## Scale and Size an Image

As soon as the image displays on the page (Figure 4.13), you can tell that you have a big problem. This image is way too large.

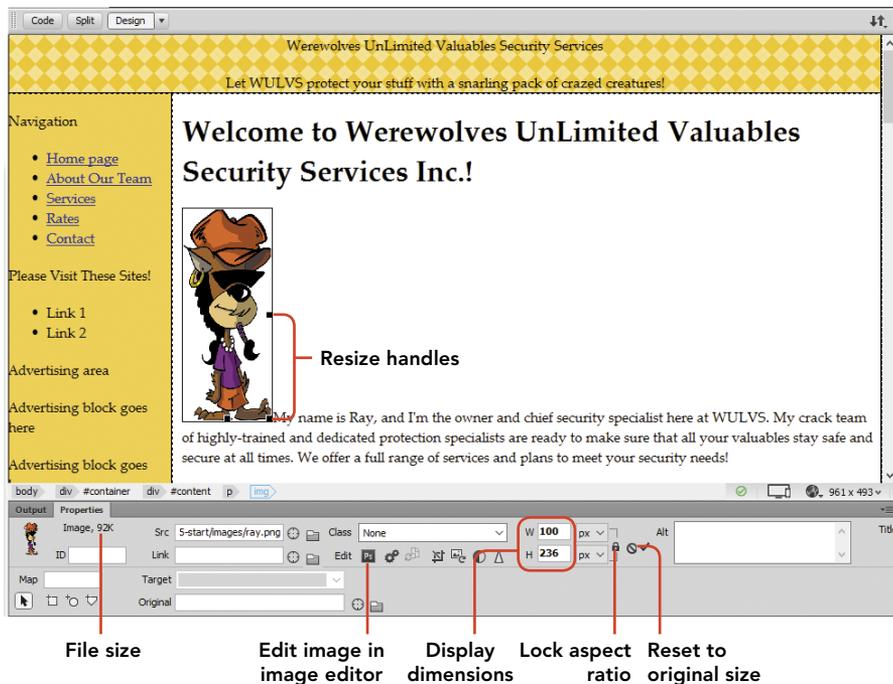
When an image is selected, you'll see a small, square resize handle appear in the image's lower-right corner. You can scale the image by holding down the Shift key while dragging this handle. Or you can click the lock icon in the Property inspector to lock the **aspect ratio** and keep the image set to the proper width and height dimensions as you drag.

- 1 Click the image, and drag the image's resize handle to what you think will be the proper dimensions for this image.

As you resize the image, dimensions will change in the Property inspector. Make note of those dimensions.

When you resize or scale an image in this manner, you are changing the width and height that the browser will use to *display* the image. The image itself is not changed in any way, and most importantly, the file size of the image is the same as always. Placing a large image onto the page and then setting a smaller width and height is a waste of bandwidth. A smaller version of this image works better in this case.

**Figure 4.13** The Property inspector tools for setting image dimensions



- 2 Undo your changes to the image so it returns to its original size, or click the Reset to Original Size button in the Property inspector.

In the next steps, you'll use a better method for editing an image that is too large for the page.

## Duplicate and Edit an Image

Remember that smart designers never (ever!) destroy an original image. You might be tempted to open `ray.png` in Photoshop, change the image dimensions, save the file, and plunk the smaller version onto the page. But that kind of destructive process changes the file forever. You could never make the image larger again, and because you never really know how you might want to use this image in the future, it's always better to keep the original file as it is and work from a duplicate of the file. The Adobe toolset makes this an easy process.

- 1 Select the `ray.png` image, and press the Delete key to remove it from the page. You will not be using this file for this location.
- 2 Click the Files panel, and expand the `images` folder.
- 3 Select `ray.png`. Right-click, and choose Edit > Duplicate from the context menu (Figure 4.14).

Notice the other basic file operations you can perform from the Edit menu.

Anytime you duplicate a file in this way, Dreamweaver automatically appends the name with “- Copy.” In this case, Dreamweaver will create a file named `ray - Copy.png`, which you'll see listed in the Files panel.

- 4 Click twice—but not too fast—on the file named `ray - Copy.png`. The filename is highlighted in blue. Change the name of the file to `ray-smaller.png`.

This is the file you will use in your design, leaving the original image of Ray as it is, safe and sound.

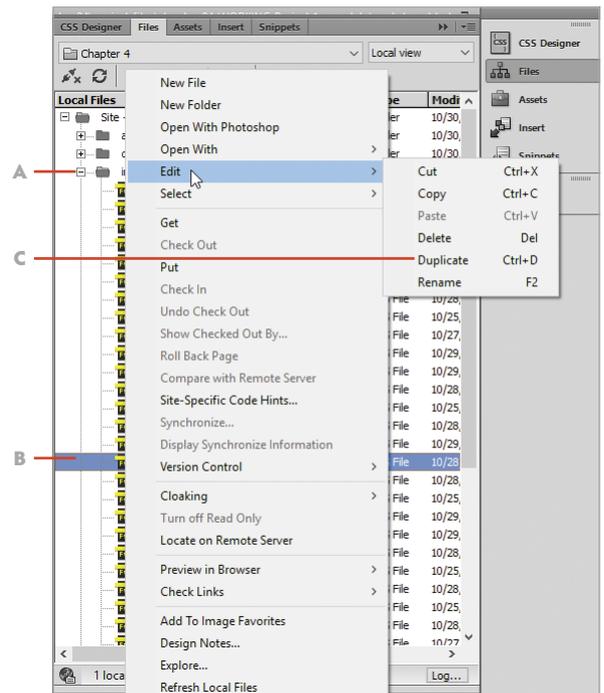
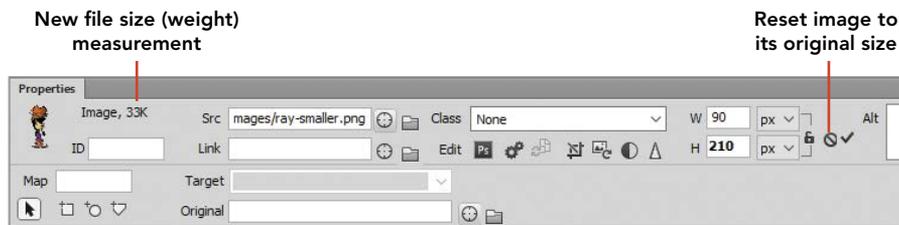


Figure 4.14 The Files panel allows you to duplicate files and perform other basic file operations.

- A Expanded `images` folder
- B `ray.png` selected
- C Duplicate an existing file

- 5 Drag the `ray-smaller.png` file from the Files panel, and drop it onto the page in front of the first sentence (“My name is Ray...”).  
The image is at its original size, but now you can safely edit this image without harming the original.
- 6 With the image still selected in the page, click the Edit button in the Property inspector to open Photoshop (or the default image editor you set in Preferences).
- 7 Resize the image, and save the changes you make in your image editor. In Photoshop, you can resize the image by choosing Image > Image Size. Set the image width to 90 pixels, and save the file.
- 8 Return to Dreamweaver and note the appearance of the image. It’s still too big. The Property inspector still shows the original dimensions. To set the dimensions to the new, correct dimensions, click the Reset to Original Size button in the Property inspector (**Figure 4.15**).

**Figure 4.15** Resizing an image in a graphics editor like Photoshop reduces the file size of the image and saves on bandwidth.



Notice that the image is displayed at the dimensions you set when working in your image editor. Also note the new file size of this image as shown in the Property inspector: only 33 Kb. You saved almost 60 Kb of unnecessary data that doesn’t have to be downloaded by a site visitor. And you saved bandwidth and made a backup of the original client artwork all at the same time. Good job!

## Dreamweaver’s Image Editing Tools

Dreamweaver includes tools in the Property inspector for editing images and performing small adjustments directly in Dreamweaver. But be warned! These tools make permanent changes to the image once you save the Dreamweaver page you’re working on.

Figure 4.16 shows the location of these tools in the Property inspector. You can try these out for yourself, but be sure to choose Edit > Undo after you make each change so the file you are using isn't permanently altered.

- A Crop Image** displays a crop area. Use the crop area handles to set the size of the area to be cropped. Double-click to crop the image.
- B Resample Image** button is used when you have changed the dimensions of an image and you want to improve its picture quality at its new size and shape.
- C Brightness and Contrast** is used to make small changes to how dark or light an image is and how sharply the colors in the image are defined.
- D Sharpen** adjusts the focus of an image by changing the contrast of pixel edges in the image.

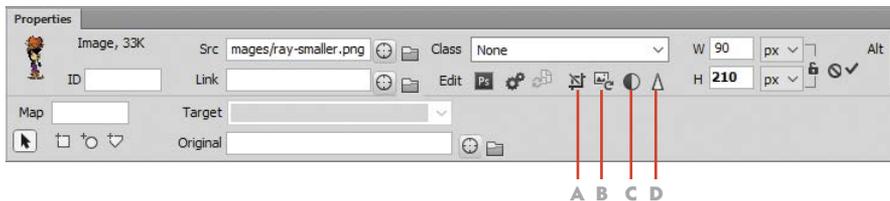


Figure 4.16 Image editing tools in Dreamweaver as seen in Design view

## LEVEL UP: SMART FILE MANAGEMENT

You're probably beginning to get a feel for just how complex a website can become and just how many files and folders there are to deal with. And this is a very small website.

One file management method that most web designers use is to keep a separate folder within their site where they store all their original Photoshop or Fireworks source files. This might also be the location where files used in other creative tools, such as Adobe Flash, might be stored. These source files aren't used in the actual HTML documents you create until a version of the source is published from the editor to a format that web browsers can display. But even though these files aren't part of the final web page product, having them stored inside the main (root) folder for each site you create makes it easier to find them when it's time to make edits and export for the web.

## Round-Trip Editing with Photoshop

Recall that at the beginning of this project you were told that smart designers always work from an original source file from Photoshop or Fireworks and never take actions that will destroy an original. However, web designers in particular never know where their artwork might come from or in what format. You might, for example, have the freedom to work from an original composition that you create in Photoshop, using all your own art and photos to compose something entirely new. Photoshop has awesome tools that let you compose an entire page layout and design in one PSD file and then export the images and CSS values for use in a web design project.

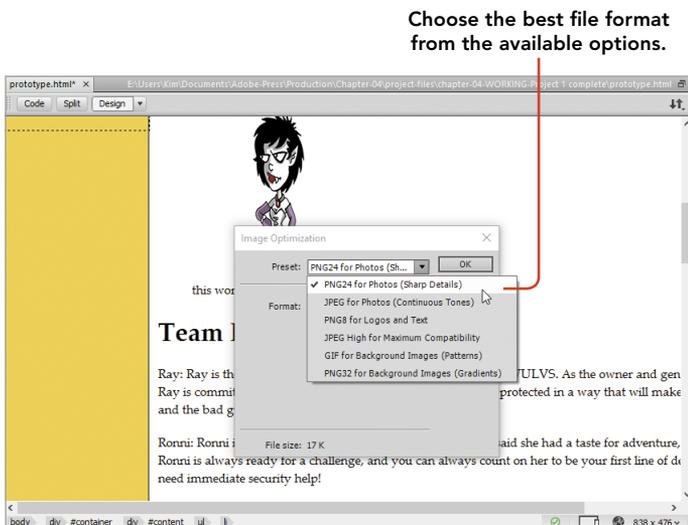
Web designers *also* have to work in a setting like the one you see here, where the client has provided artwork and instructions on how they should be used. Once you find that you need to modify an image in any way—changing its size, adding text, or even making color corrections to a photograph—you should create a source file so you can have a backup on hand in a format that lets you go back and make additional changes.

Dreamweaver provides a way to do this by allowing you to make **round-trip** edits from a source Photoshop file. Let's see how that works.

- 1 From the Files panel, expand the source folder and locate the file named `ronni-original.psd`.
- 2 Drag the `ronni-original.psd` file onto the page at the end of the last bulleted sentence.

Dreamweaver will display the Image Optimization window, where you'll choose the appropriate image format for the file. For this file, you'll continue to use PNG24 as the format to be saved.

- 3 Click the drop-down arrow in the Preset field, and select the PNG24 option (**Figure 4.17**).

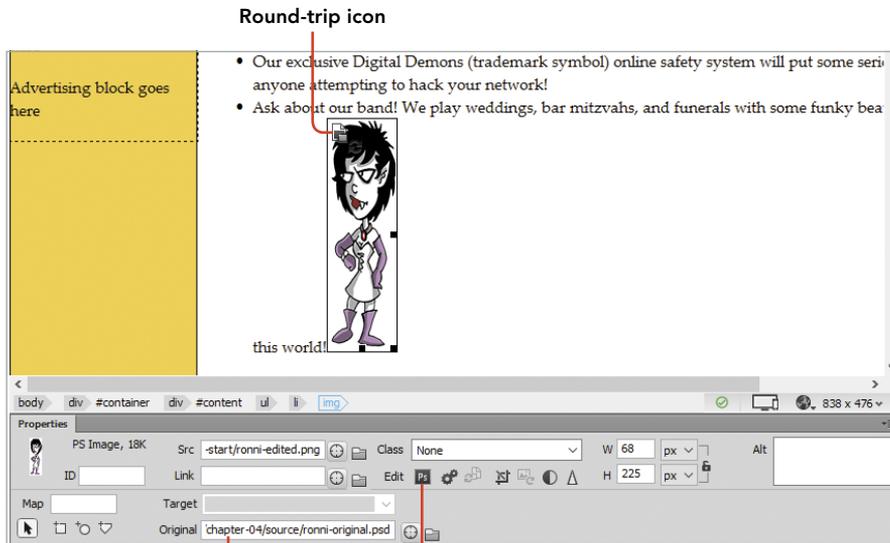


**Figure 4.17** The Image Optimization window appears when you insert a PSD file in a page.

- 4 Click OK to set the format, and Dreamweaver will open an Explorer or Finder window so you can save the image into its web-ready format. Browse to the `images` folder inside the Chapter 4 site, and save the file as **ronni-edited.png**.

In the Property inspector, you'll see that Dreamweaver is now tracking the path and filename of the source image used to create this version (Figure 4.18). The next time you click the Edit button, the PSD file will be opened in Photoshop and not the version that is on the page.

Now look at the image itself. You'll see a small icon in the upper-left corner of the image that indicates this is a version of an image that came from a Photoshop source file and is available for round-trip editing.



**Figure 4.18** Source Photoshop files are tracked by Dreamweaver.

Path and filename of source PSD file

Open the source PSD file in Photoshop.

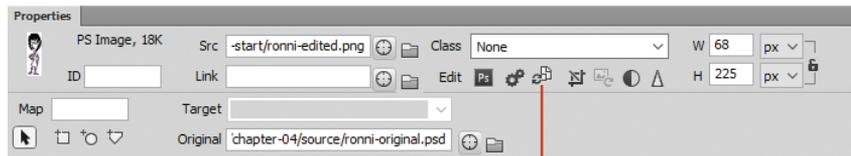
- 5 Click the Edit button and the PSD file will open in Photoshop.  
Let's look at a quick demonstration of how round-trip editing works.
- 6 In Photoshop, choose Image > Image Rotation > Flip Canvas Horizontal. The cartoon character will rotate to face right instead of left.
- 7 Save the PSD file, and return to Dreamweaver.  
Everything looks the same. No worries. To see the change you made in Photoshop, you have to tell Dreamweaver to update the image.

- 8 Click the Update from Original button in the Property inspector (Figure 4.19).

The character is now looking to the right, just as she is in the source file.

- 9 Save the `prototype.html` file you've been working on. This project is complete.

**Figure 4.19** The Property inspector when working with a round-trip image



**Update from  
original source file.**

You've just learned a lot about how images are inserted into web pages using Dreamweaver's toolset. You've also seen how you can use Dreamweaver's built-in image editing tools for making minor corrections to an image and how Dreamweaver and Photoshop can be used together in the recommended workflow, where the designer makes edits to a source PSD file that can be easily updated in Dreamweaver. And don't forget to always back up your originals!

## PROJECT 4.3

# Floating Images into Position

You have inserted the images for the home page into your prototype page, but they still don't look like the design comp. The two images are currently resting on the baseline of the paragraph that contains them. But you can nudge them to the location you want using the CSS `float` property.

★ ACA Objective 1.2

★ ACA Objective 2.6

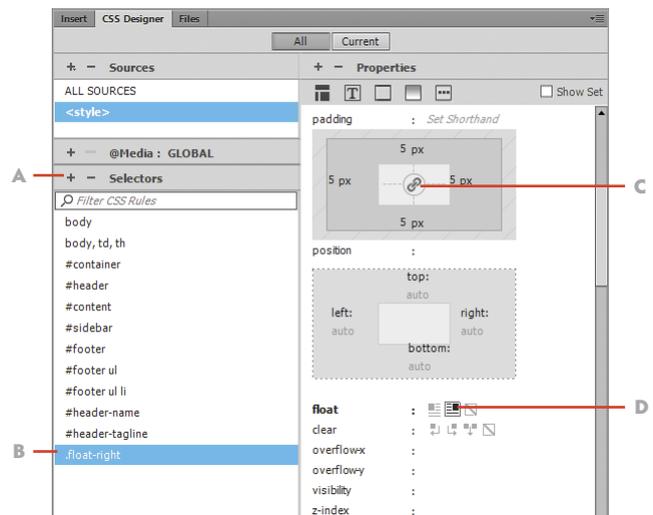
In this project, you'll float the two images into place. You'll also learn some of the principles of floating images, learn how to set the `clear` property so your floats behave themselves, and use some Dreamweaver tools that make this kind of work go more quickly.

▶ Video 4.3 Floating Images into Position

## Create and Apply a CSS Class Selector

By default, images that are inserted into a web page rest on the **baseline** of the element where they've been placed. When images are inserted into a paragraph, the image will set along this invisible line along the same line of text that is adjacent to them. The CSS `float` property is used to disengage the image from the baseline and allow the text to flow around the image.

- 1 Open `prototype.html` in Dreamweaver, and make sure the two final versions of the cartoon characters—Ray and Ronni—that you inserted in the last project are positioned where you want them on the page.
- 2 Open the CSS Designer panel, and click the + at the top of the Selectors section, as you see in **Figure 4.20**. Dreamweaver will create a new blank Selector field where you can name the selector.



**Figure 4.20** Adding a class selector in the CSS Designer panel and assigning properties

- A Click the + to add a new selector.
- B The new selector is listed once a name has been entered.
- C Click to apply padding values to all four sides.
- D `.float-right` property applied

**TIP**

Class selector names always begin with a dot (.). One way to remember this is that your classes at school are usually divided into periods as well—just remember that school classes have periods and so do CSS class selectors.

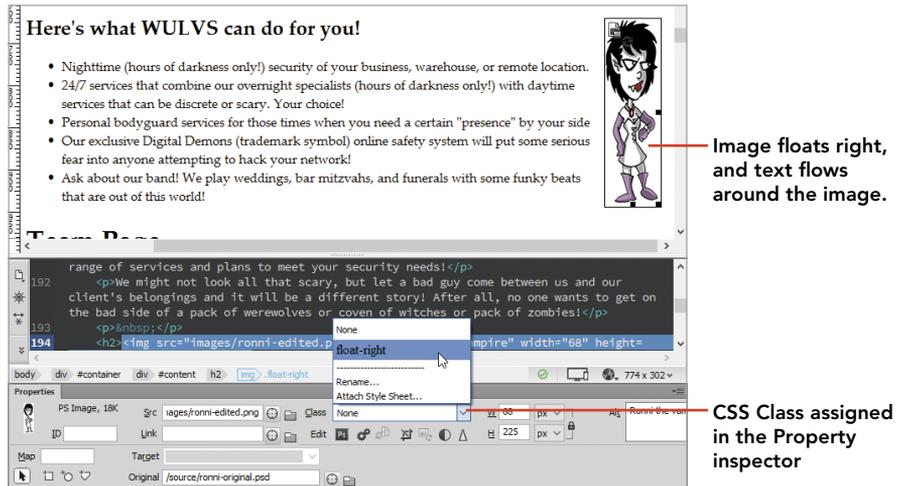
- 3 Name the new selector **.float-right** and press Enter/Return to accept the name.
- 4 In the Layout Properties category, click the **float-right** property button. You have now created and named a new class selector that can be used over and over again within the document.
 

It's a good idea to apply a bit of padding to an image so that text doesn't jam up against the side of the image, making the text harder to read.
- 5 Click the symbol in the middle of the padding properties section (Figure 4.20). Set a padding value of **5 px** in one location, and Dreamweaver will duplicate the setting to the other three sides all at once—slick!
 

Now you can assign the **.float-right** property to one of your images.
- 6 Select the Ronni character on the page.
- 7 From the Class drop-down menu in the Property inspector (Figure 4.21), select **.float-right**.

The image floats to the right of its container, and the text flows around the image. You can drag the image into different areas of the page, but the image will always float to the right of its container no matter where you drop it.

Figure 4.21 Applying a float selector to an image



## Duplicate and Apply a CSS Class Selector

You still need a class selector to float an image to the left, but Dreamweaver has a nice way to make this process quick and easy. You can duplicate the existing selector, modify the name, and change the values for the new selector.

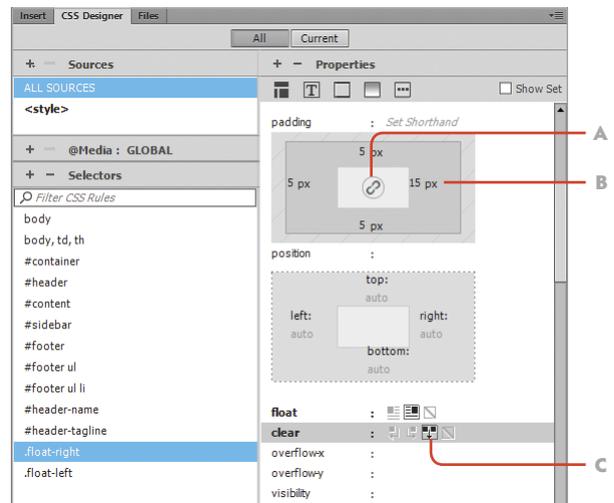
- 1 Right-click the `.float-right` selector, and select Duplicate from the context menu. Dreamweaver makes a copy of the selector, and your cursor will be positioned inside the name of the new rule.
- 2 Change the name of the new selector to `.float-left` and press Enter/Return to accept the new name.
- 3 In the Properties section of the panel, change the float setting to `left`.
- 4 Use the Classes setting in the Property inspector to apply the `.float-left` class to the Ray the Werewolf image. The image floats to the left side of the content container.

## Modify CSS Float Properties and Apply Clear Values

The design is looking good, but the image of Ronni floated to the right is smashed up against the right border of the container. And you could have an issue with overlapping images when you begin floating more images into place. In the next steps, you'll adjust the padding on the `.float-right` class and set the Clear properties that will prevent the floated images from overlapping other elements that have been floated.

- 1 In the CSS Designer, select the `.float-right` selector.
- 2 In the Properties panel, click the center of the Padding section to unlock the settings, and change the right padding setting to 15 px, as in Figure 4.22.

The other settings will remain the same as you change this one value. Setting the Clear values for each float class is just as easy.



**Figure 4.22** Modifying the `.float-right` class selector and setting the Clear value

- A** Unlock the padding settings to change individual padding values.
- B** Set padding right to 15 pixels.
- C** Set clear value to Both.

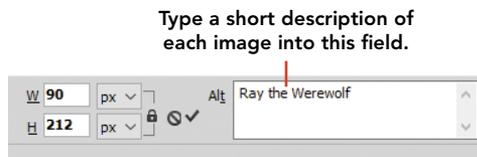
- 3 With the `.float-right` selector still selected, click the Both value for the Clear setting (Figure 4.22).
- 4 Select the `.float-left` selector, and repeat the process of setting the Clear value to Both.

With both float classes properly set to `clear: both`, the next image you float will position correctly without any overlap.

## Add Alternative Text to Images

For web pages to be accessible to individuals with vision and other disabilities, web page images must always include alternative text. Many government agencies require web pages to meet **accessibility** standards, and it's also just a good practice to follow because the description will display in the event the image doesn't display for some reason. Assistive devices, such as screen readers, will read the **alt text** to an individual who might not be able to see the image clearly. It's also helpful when your site is indexed by search engines, so it's never a bad thing to be thorough and include alt text. This task is usually handled with the Property inspector.

- 1 Select the image of Ray.
- 2 In the Alt field in the Property inspector (Figure 4.23), type **Ray the Werewolf**. Click anywhere on the page to accept the value.



**Figure 4.23** The Alt text field in the Property inspector

- 3 Select the image of Ronni.
- 4 In the Alt field, type **Ronni the Vampire**, and click outside the field to accept the value.

That's it; pretty simple.

This project introduced you to the concept of floating images—the process where a CSS rule is applied to an image so that text can flow around the image. You know now how to use Dreamweaver to create a new class selector, how to assign the properties and values you want to the selector, and even how to duplicate a selector to make the design process go a little faster. You've also learned about how to assign alt text to images so that your work meets accessibility standards. Another nice piece of work!

## PROJECT 4.4

# Creating Structured Layouts with Images

In the last two projects, you worked a bit with client-provided artwork to float images onto the page. Where there are single images like the ones you've used so far, inserting images and applying a float to move the graphic into position is relatively easy.

But the About Our Staff page features blocks of text with the name and description of each character in the comic strip. In the olden days, layouts like this—where an image is aligned and positioned next to a paragraph of text—would have been done using tables. In the modern world of web design, designers use more efficient methods that take advantage of advances in browser support for CSS.

Dreamweaver accommodates the kinds of efficient, CSS-based designs that have become the industry standard but also adds some visualization features that allow designers to get things just right.

## The Design Challenge

The page that features all the characters from the comic strip should have a photo of the character, with their name and “bio” alongside the image. The client helpfully provided all the character images at the same width and height, which will make the design work a bit easier.

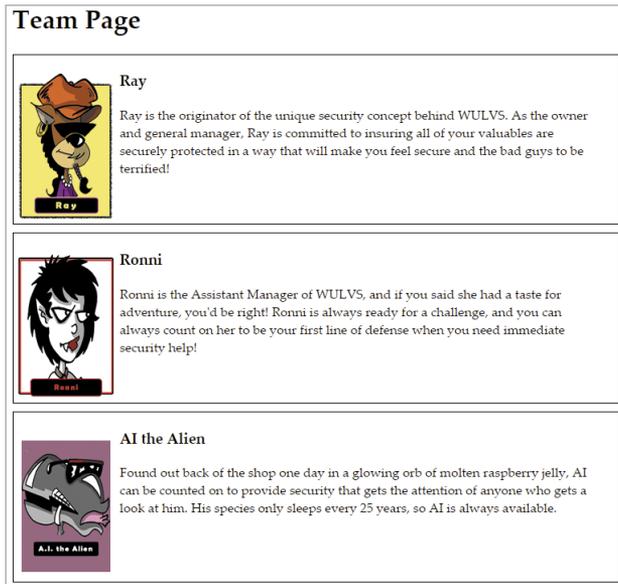
The challenge for this layout is to maintain the association between the text and the image. You might consider inserting the image and using the CSS floats that were created in the last project to slide the image into place. The problem with this approach is that floated images don't create a new “break” within the content on the page. You would have to insert a number of empty paragraphs by pressing Enter/Return over and over until each image appeared next to a line of text and had blank areas between each character. Or you might consider inserting a table and placing the images into the left column and the text into the right column. Both of those approaches are semantically incorrect, difficult to maintain, and just a bad idea.

★ ACA Objective 1.2

★ ACA Objective 2.6

▶ **Video 4.4** *Creating Structured Layouts with Images*

The better approach is to create a container that holds both the image and the text, and then use the power of CSS classes to define the container and all the content that applies to each character. The result of this project's work will have the image of each character appear next to its information (**Figure 4.24**).



**Figure 4.24** When this project is complete, each character will have its own content box to display its image and bio.

## Preparing Content for the First Character Container

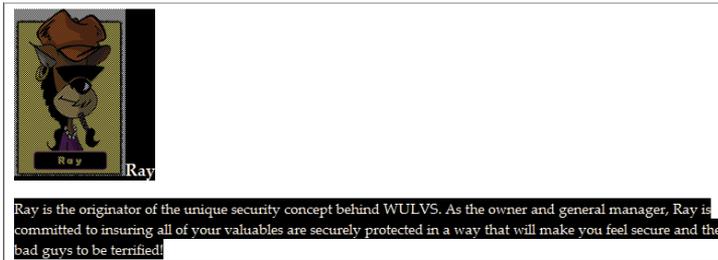
The client provided the name of each character and their bios in the Team Page area of `prototype.html`. With that file open, it's time to begin.

- 1 Place your cursor after the first character's name (Ray), and press Enter/Return to insert a new paragraph. Clean up any extraneous characters or spaces until your text appears as follows:  
Ray  
Ray is the originator of...
- 2 Place your cursor in front of Ray's name, and use the Property inspector to assign the Heading 3 format to the character's name.
- 3 In the Assets panel, locate the `ray-named.png` file, and insert it into the page in front of the character's name.

## Inserting a <div> for the Character Content

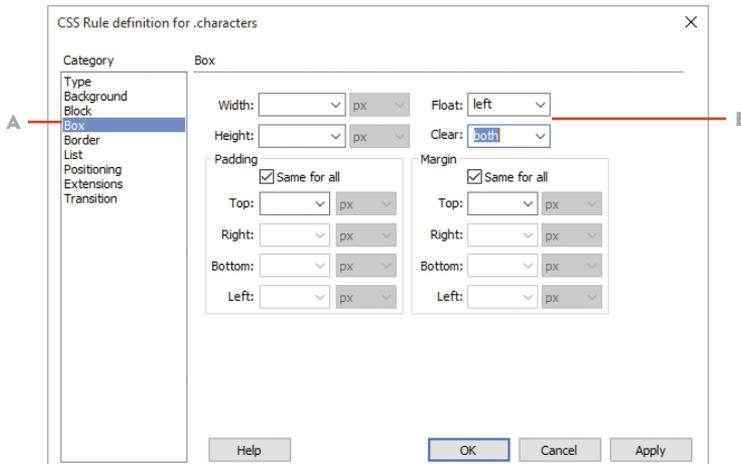
Now you need to create the container that will hold all the contents for each character: their image, their name, and their description.

- 1 Drag to select the image and all the text associated with Ray (Figure 4.25).



**Figure 4.25** The content on the page selected and prepared for styling

- 2 Choose Insert > Div in the application bar. The Insert Div dialog appears.
- 3 Leave the Insert field in this window set to Wrap Around Selection, and click the New CSS Rule button at the bottom of the dialog. The New CSS Rule dialog appears.
- 4 With the Selector type at the top of the window set to Class, name the new rule **.characters**. Click OK. The CSS Rule Definition dialog appears.
- 5 Select the Box category in the Category column, set the Float property to **left** and the Clear property to **both** (Figure 4.26), and click OK. You return to the Insert Div dialog.
- 6 Click OK in the Insert Div dialog to complete the process of inserting the <div>.



**Figure 4.26** The CSS Rule Definition dialog for the new **.characters** rule

- A** Select the Box category.
- B** Set the float and clear properties as shown.

The image and all the text are now contained inside their own container with the attribute of `class="characters"`. Dreamweaver displays this container with a dotted line around it when you are in Design view.

## Floating an Image Inside a Container

Now it's time to style the elements that are contained within this new `<div>`. Take note that these styling rules will use descendent selectors to create styling rules that will apply to *any* content found inside a `<div>` with the class name of `.characters` applied.

- 1 Select the image of Ray.
- 2 In the CSS Designer panel, click the + in the Selectors section to create a new rule based on the item selected on the page.

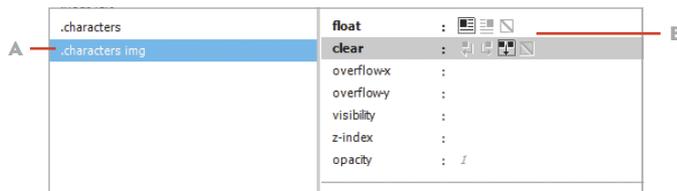
Dreamweaver will suggest that you name this rule `.characters h3 img`. Do you see how the cascade works? This rule will apply to any image that is inside an `<h3>` tag that is located in a `<div>` with the `.characters` class. But this is a little more specific than you need in this case.

- 3 Delete the `h3` descendent so that your rule name is `.characters img`. This rule will now apply to any image found within the `.characters <div>`.
- 4 In the Layout category in the Properties section, set the float property to `left` and the clear property to `both` (Figure 4.27).

The text that's associated with Ray jumps up beside the image, just like you want. Don't be too concerned about the position of other characters on the page. Once all the containers are in place, things will sort themselves out.

**Figure 4.27** Settings for the `.characters img` rule

- A** Choose the selector name.
- B** Apply float and clear properties.



## Create a Second Character <div>

To see the interplay between two boxes that contain a character, let's repeat the process you just followed for the Ronni character and insert a `<div>` around that area of the page.

- 1 Edit the text so that Ronni's name is on a new line with her description below her name.
- 2 Format the character's name as an `<h3>`.
- 3 Insert the image `ronni-named.png` to the left of her name.
- 4 Select all the text and the image, and then choose Insert > Div from the application bar or from the Insert panel.
- 5 In the Insert Div dialog, select the existing CSS `.characters` class from the Class drop-down menu (Figure 4.28). Click OK.

The contents of this new `<div>` snap into place, and you now have two content boxes that you can use to adjust the styling.



**Figure 4.28** You can assign an existing CSS class when you insert a new `<div>`.

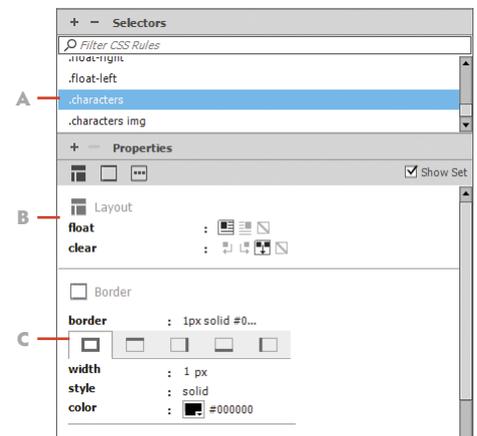
## Styling the Content Containers

Now that you have two character containers, you can fine-tune the styling properties for their contents. The beauty of using CSS rules is that once you adjust the settings for one container, all the other containers with the same class applied will be automatically updated.

- 1 In the CSS Designer panel, select the `.characters` selector, and apply a border to the box. Make the border 1 pixel wide, with the style set to solid and the color set to black (`#000000`).

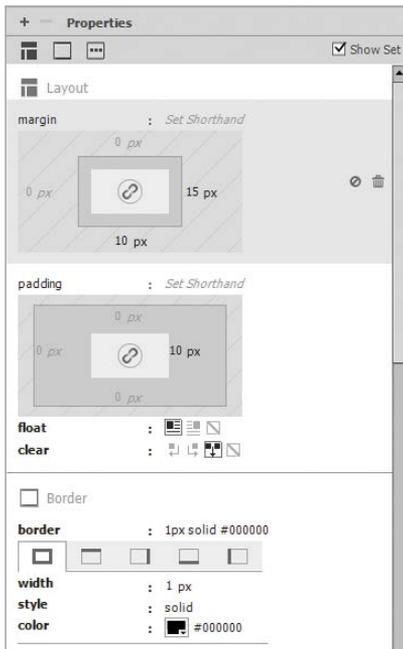
You can select the Show Set checkbox to see only those properties assigned to the selector. Your settings should match those in Figure 4.29.

When you switch to Live view, you'll see that the two containers have no space between their respective boxes. Text within the box is too close to the image, and the boxes are jammed up against the right side of the page. You'll adjust these settings using the margin and padding properties.



**Figure 4.29** Border properties added to the `.characters` selector

- A** Choose the selector name.
- B** Existing float and clear properties
- C** New border property settings



**Figure 4.30** Final settings for the `.characters` rule

- 2 With the `.characters` rule still selected, turn to the Layout category of the CSS Designer panel, and set the bottom margin of the container to **10 px**. This adds space between the two boxes and separates them from each other.
- 3 Apply a right margin of **15 px**. This moves the right edge of the container away from the edge of the page.
- 4 Apply a right padding of **10 px**. This moves the text within the box away from the right border of the box.

The final settings for the `.characters` rule should appear as in **Figure 4.30**.

The final bit of tweaking needs to take place on the image, adding a bit of padding to the image to provide some “air” between the image and the text.

- 5 Select the `.characters img` rule.
- 6 In the Layout category, assign the padding values that you think look best for the right, bottom, and left padding properties. This is a great time to work in Live view so you can clearly see the changes that you make as they are applied.

## Aligning Elements with Guides

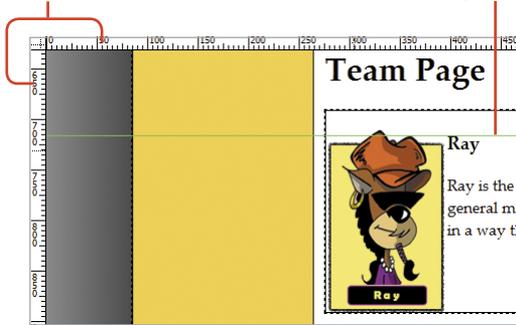
One of the big advantages to using Dreamweaver to create and style web pages is the ability to put visualization tools into place that let you fine-tune your design. Guides are horizontal and vertical lines that Dreamweaver displays as you’re working. You can use these guides to check the alignment of elements on the page.

- 1 With your document in Design view, choose **View > Rulers > Show** to display rulers on the top and left side of the document window.
- 2 Choose **View > Guides > Show Guides**. With this option selected, you can click one of the rulers and drag a guide onto the page.
- 3 Click the top ruler, and drag to position a guide directly above the first character’s name. If you’re having problems getting the guide into just the right place, deselect both **View > Guides > Snap To Guides** and **View > Guides > Guides Snap To Elements**.

As you can see in **Figure 4.31**, the image isn't perfectly aligned with the text. But you can fix that easily enough.

Rulers appear on the left and top of the Document window.

Drag guides onto the page to check alignments.



**Figure 4.31** Rulers and guides allow you to check alignment of page elements.

- 4 In the CSS Designer panel, select the `.characters img` rule, and adjust the top padding property to push the image down until it perfectly aligns with the character's name.
- 5 Drag a guide from the left ruler and position it in line with the right border of the character box.

As you scan up the page, you'll be able to see whether everything falls into alignment with this guide. Everything looks good in your character boxes, but the text in the paragraphs contained in the `#content <div>` are not in line with the borders of the character containers.

- 6 Click anywhere in the first line of the introductory text on the page, then click the + in the Selectors section of the CSS Designer panel. Dreamweaver will suggest that this rule be named `#container #content p`.
- 7 Press the Up Arrow key on your keyboard to make this rule a little less specific. The actual name you're after here is `#content p`, which is specific enough for your purposes.
- 8 Apply a right padding setting to this rule until the text in this paragraph and in others on the page is aligned with the guide.
- 9 Save your page, and preview it in your favorite browser. Look for alignment issues that need to be addressed and return to Dreamweaver to make adjustments to the styling rules you've just created.

## Completing the Character Page Listing

You're almost finished with this project; you have just one more task to complete. Following the same steps you used to get the Ray and Ronni character containers constructed, you need to edit the text, apply a Heading 3 to the character's name, and then insert the character's image next to the name. You'll then wrap the character's contents into a new `<div>` and assign the `.characters` class. As you complete each container, the styling rules you've created will be applied to the contents, and before long all the employees of WULVS will have their own neatly aligned listing on the page.

## PROJECT 4.5

# Styling the Header Area

From the client feedback document you reviewed, you know that the final design needs to feature some of the characters from the comic strip at the top of the page in the header area. That's going to be a relatively easy task to accomplish since you can use CSS floats and a couple of collage images that the client has provided.

While you're styling this area, you should tend to some other matters as well. In this project, you'll insert and float the images into the header area, add links to the home page of the site, and use a web font to style the name of the company.

★ ACA Objective 1.2

★ ACA Objective 2.6

▶ Video 4.5 Styling the Header Area

### LEVEL UP: YOUR DREAMWEAVER SKILLS

Throughout this book, you've seen lots of detailed instructions on how to apply settings and where to find the different working areas of Dreamweaver's user interface. But now it's time for you to build your confidence in using the application by following the steps provided for this project without a great deal of extra instruction. You can follow along with the video of course, but at this point I'm confident that you're capable of doing more on your own. You'll be more confident too if you endeavor to do more and more on your own as you go forward.

## Inserting and Floating the Header Images

Your client has provided new images that he wants to include at the top of the page to the left and right of the company name. In the following steps, you'll insert these images into the header area of the page and then place them into position.

The header area will need some additional room for the images to fit within the container. So let's do that first.

- 1 In the CSS Designer panel, choose the `#header` selector, and set the height in the Layout category to 136 pixels. You're now ready to insert the collage images.
- 2 Place your cursor in front of the first letter in the first line of text in the header area.

- 3 Using the Assets panel, insert the images `header-collage-left.png` and `header-collage-right.png`. These images will be floated, so you can just place them side by side. Name the images in the Alt field of the Property inspector.
- 4 Select `header-collage-left.png`, and assign the `float-left` class to the image in the Property inspector. The image snaps into place on the left side of the header.
- 5 Select `header-collage-right.png`, and assign the `float-right` class to the image.

This time the image *isn't* positioned properly; it falls outside the header. That's not good, but more than likely the issue has to do with the `clear: both` property that is assigned to this rule. Since you don't want to change the existing `float-right` class that's in use in other parts of the design, you need a new rule for this image.
- 6 Right-click the `.float-right` rule in the Selectors area of the CSS Designer panel, and select Duplicate.
- 7 Change the name of the rule to `.float-right-header`.
- 8 In the Layout category of the CSS Designer, change the clear value for this rule from `clear: both` to `clear: right`. This change allows the image to properly float into place within the header.
- 9 Select `header-collage-right.png`, and assign the `float-right-header` class to the image. This image snaps into place.

## Adding Links in the Header

It's common—even expected—for a visitor to a website to be able to click the header at the top of the page and go to the site's home page. You'll add links to the two images and to the company name to make this possible. Later, when this prototype is converted into a template, the links will be in place and ready to go.

- 1 Click the `header-collage-left.png` image. In the Property inspector Link field, enter the filename of the home page: `index.html`.
- 2 Click the `header-collage-right.png` image. Link this image to the `index.html` home page.
- 3 Select the name of the company in the center of the header area on the page. Link the text to the `index.html` home page.

## Removing Default Styles from Text Links

Text links by default have always been shown in blue, underlined text. This obviously won't work for the design the client is after, so you'll remove those properties from the `<a>` tag with a new CSS styling rule.

- 1 Place your cursor anywhere within the company name in the header.
- 2 Click the + in the Selectors area of the CSS Designer panel. Press the Up Arrow key once so that the new selector name is set as **#header-name a**. Press Enter/Return to accept the name.
- 3 In the Text category, set the text color to black and text-decoration to none. Looking at the CSS you just wrote, you should see the following code block:

```
#header-name a {  
    color: #000000;  
    text-decoration: none;  
}
```

## Styling the Company Name with Edge Fonts

The client asked for some “weird- or scary-looking” text for the company name. To get text that goes beyond the common fonts that are standard across browsers, you can use Adobe Edge Fonts, which is a free service, to find more interesting fonts to work with. The company name in this document already has a rule applied to which you can easily add additional properties and values to make the name “weird or scary.”

- 1 Select the `header-name` rule in the Selectors area of the CSS Designer panel.
- 2 Set the text color to black (`#000000`).
- 3 Click inside the font-family field. From the menu that appears, select Manage Fonts.

Adobe Edge Fonts are free font packages that will be loaded into the page with a little snippet of JavaScript that is inserted into the `<head>` of the document. The Manage Fonts dialog you see in **Figure 4.32** allows you to choose one of these fonts to use in your page.



**Figure 4.32** Adobe Edge Fonts can be selected from this window and loaded into a font library on your computer.

- 4 Select the font `amatic-sc` or try out another font that you like. Click Done to load the font into your library.  
Once you load a font into your library, you can assign the font as a property.
- 5 Click inside the font-family field, and select the `amatic-sc` font (or whichever font you loaded) for this rule.
- 6 Switch to Live view to see the appearance of the text. Edge and Web fonts display properly only in Live view.
- 7 Use the font-size and font-weight settings for this rule to make the text appear the way you like. For this example, your styling rule might appear as follows:

```
#header-name {  
    margin-top: 0px;  
    color: #000000;  
    font-family: amatic-sc;  
    font-weight: 700;  
    font-size: 36px;  
}
```

## Styling the Company Tagline

One more job to take care of. You need to style and position the tagline that falls below the company name. Here's the styling rule that was created to get this done in the video demonstration:

```
#header-tagline {  
    margin-bottom: 0px;  
    color: #1B1B1B;  
    font-family: "Gill Sans", "Gill Sans MT", "Myriad Pro",  
    →"DejaVu Sans Condensed", Helvetica, Arial, sans-serif;  
    font-style: italic;  
    font-size: 14px;  
    margin-top: -36px;  
}
```

Can you figure out where to apply these settings in the CSS Designer panel?

In this project you've taken another step in becoming proficient with Dreamweaver by diving in on your own to insert images and links and apply styling to the header of the page. Another job well done! Do you feel your confidence rising?

# Challenge! Create Your Own Design with Photoshop

As you've worked through the material in this chapter you've no doubt thought of lots of ways that this design might be done. The client provided only some general guidance, and the examples you've seen here took off in sort of a hipster-look direction. But you're heartily encouraged to do better by creating your own design.

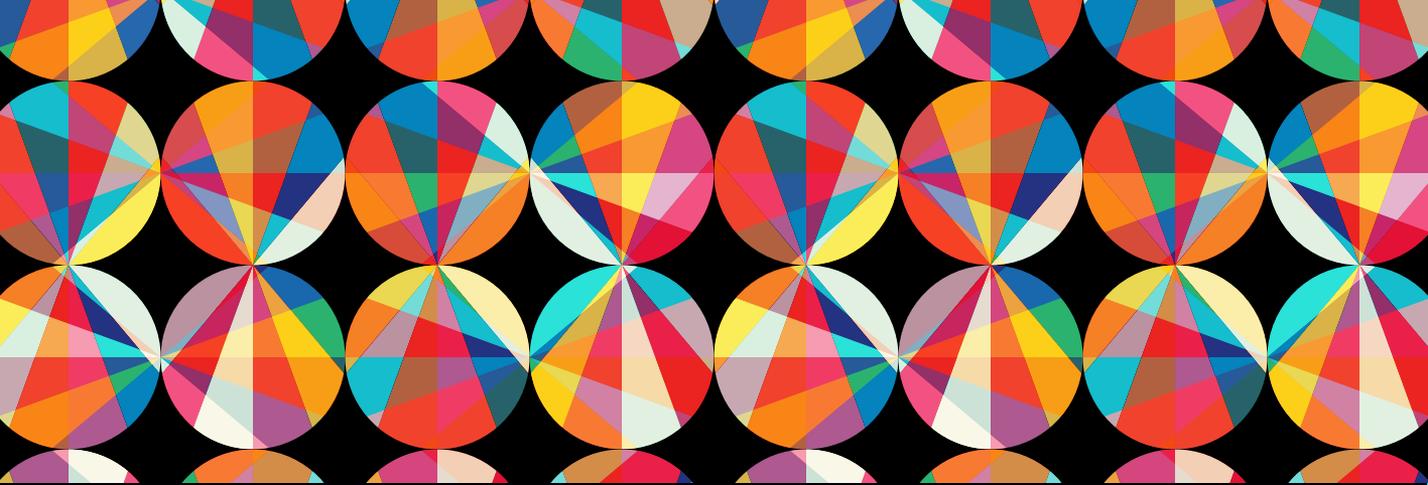
To help get you started, the project folder for this chapter has two files in the `source` folder that you can use to create your own design comps. You'll find a file for Photoshop (`wulvs-layout.psd`) and one for Fireworks (`wulvs-layout.fw.png`). Both files are already organized into layers that match the dimensions used in developing this prototype. It's up to you to open those files and use your own creative ideas to make a totally new design.

## Conclusion

Although the title of this chapter is "Getting the Big Picture," it might as well have been named "Working with Backgrounds, Inserted Images, Named CSS Class Rules, and Dreamweaver Visualization Techniques." Well, that's a little long, but that's just what you've accomplished as you've worked through these projects.

Along the way to learning how to use Dreamweaver to include the visual elements that images provide, you've also learned more about the principles of web design, how CSS styling rules are used to position elements on a page, and even how to style text using CSS and fonts that are located on a web service.

And the great news is that your design work is nearly complete. In the next chapter, you'll work to get the final elements of this design wrapped up as you learn how to style navigation bars, tables, and more.



## CHAPTER OBJECTIVES

### Chapter Learning Objectives

- Respond to new client requirements by modifying an existing design.
- Work with client-provided content to convert Microsoft Word to properly formatted HTML.
- Create navigation elements using CSS to create interactive rollover buttons.
- Learn how to apply special CSS selectors to style links and alternating rows within a table.
- Create and style an interactive HTML form.
- Perform technical tests to check for and correct errors.
- Complete the final steps in the prototype to get the client's okay and prepare to move the site into production.

### Chapter ACA Objectives

For full descriptions of objectives, see the table on pages 240–246.

**DOMAIN 1.0**  
SETTING PROJECT REQUIREMENTS  
1.3, 1.4

**DOMAIN 2.0**  
UNDERSTANDING WEB AUTHORING  
2.4

**DOMAIN 4.0**  
WEB AUTHORING USING ADOBE  
DREAMWEAVER CC  
4.1, 4.2, 4.4, 4.5, 4.6

**DOMAIN 5.0**  
TESTING, PUBLISHING, AND MAINTAINING WEB  
CONTENT USING ADOBE DREAMWEAVER CC  
5.1

## CHAPTER 5

# Working with Web Page Content

You've gotten more great feedback from your client, Chris the Cartoonist, and the design for the WULVS website is really coming together. With the work that you did in Chapter 4, the page design is beginning to define itself. And now the client's list of things to do has provided some great new clues for the remaining design specifications, some new images that need to be put into place, and some specific information about color values and the appearance of the elements of the design.

In this chapter, you'll take the final steps in preparing this website for production by evaluating a list of additional design and functional requirements that the client has provided. You'll then set out to complete your new to-do list by styling the navigation elements for the site, creating rollover buttons, styling the tables of information that Chris provided, creating a contact form, and cleaning up your code. And of course along the way, you'll learn even more about using Adobe Dreamweaver to create and style your pages using good solid coding techniques.

## Bring on the Content!

More great news. Your client, Chris the Cartoonist (**Figure 5.1**), has provided feedback for the WULVS.com website.



**Figure 5.1** Chris the Cartoonist

### Chris Sez:

*Really love all the work you've done. Those characters look awesome. Nice and clean. Great job! It looks like there are still a few things to do, so I made you a list. 😊*

- *The sidebar navigation area looks pretty plain, and of course we need some buttons over there. I have some ideas about how I want that to look, and I gave you some images to get started.*
- *The tables that I gave you before for the Services and Rates pages still need to be styled, right?*
- *Also, the form for the Contact page. That still needs to be done.*

*Overall, I'm really loving the way things are working out! I can't wait to see the final prototype.*

*Chris*

## Get Your Files Here

As you did in previous chapters, copy the folder that contains this chapter's files to your computer.

- 1 Download the project files for this lesson, named `chapter-05.zip`, from the Lesson & Update Files tab on your Account page at [www.peachpit.com](http://www.peachpit.com), and store them on your computer in a convenient location.
- 2 Unzip the files and copy the `chapter-05` folder to the `Learn-Dreamweaver` folder you created in Chapter 1.

★ ACA Objective 4.1

### NOTE

Refer to Video 1.5, "Defining a Dreamweaver Website," if you want a quick demo of how the site definition process works.

## Define the Chapter 5 Website

With the files you've been provided for this chapter in the project folder, you're ready to set up the site in Dreamweaver by giving the site a name and pointing to the project folder for this chapter.

- 1 From the application bar at the top of Dreamweaver, select Site > New Site.
- 2 In the Site Setup dialog, in the Site Name field, name the site **Chapter 5**.
- 3 Use the folder icon to the right of the Local Site Folder field to browse to the `chapter-05` folder in your `Learn-Dreamweaver` project folder.
- 4 Click Save to finish defining this new site.

## PROJECT 5.1

# Styling the Navigation Sidebar

In this project and in Project 5.2, you'll work "outside in" to complete the styling of the sidebar and make some slick rounded buttons from the existing list items in the document. First let's take a look at the client's requirements, then review where you should be in your own design work, then finish this project by inserting images into the sidebar and setting the stage for the rollover buttons you'll make from the list of links.

- ★ ACA Objective 1.4
- ★ ACA Objective 2.4
- ★ ACA Objective 4.4
- ★ ACA Objective 4.5

 **Video 5.1** Styling the Navigation Sidebar

## Examining the Client Requirements

Take a few moments to review the list of things to do that Chris provided.

- 1 Double-click `WULVS-Things-To-Do.doc` in Dreamweaver.
- 2 Read through the list of requirements.

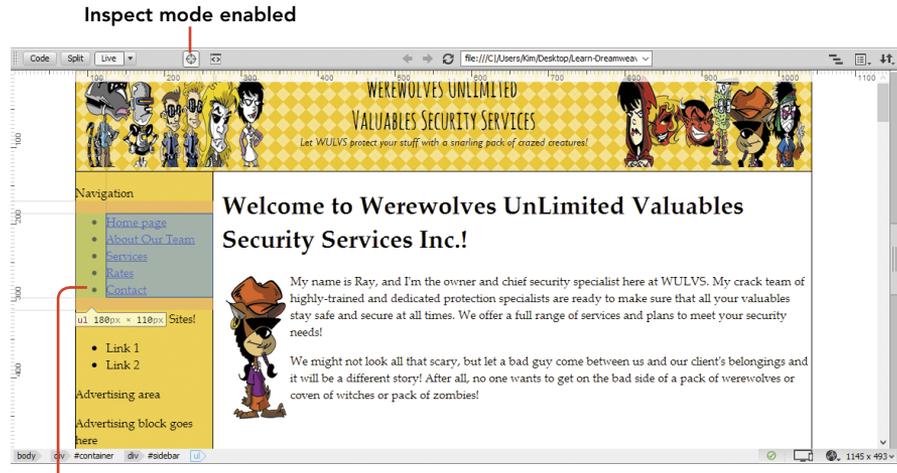
What specific guidance does the client provide? What is being left up to you as the designer? Has the client given you more information about the *feel* for how the website elements should look? You might want to jot down a list of what you want to do, but the really clever ones among you will see that the things to do match up with the projects in this chapter: styling work, images to insert, and some specific ideas from the client.

## Checking Your Work

Take a few moments to inspect the current status of the prototype.

- 1 Open Dreamweaver and review the structure of the site and the new folders and images provided in this chapter. You'll find that the folders and files that were created in Chapter 4 are still in place, but new images and documents have been provided by the client.
- 2 Open `prototype.html` from the Chapter 5 site. Switch to Live view and enable Inspect mode to examine the containers on the page and the way that padding and margins have been defined in your style sheet (**Figure 5.2**). This is a really useful method for examining the layout blocks of your design.

**Figure 5.2** Inspect mode in Live view allows you to visualize the different containers that make up the page.



**Color coding appears as you float your mouse over the page.**

- 3 Review the styling rules that you've written into the `<head>` of the document. Take note of the relationship between the named containers in the document and how the styling rules have been "hooked" to the tags using IDs and classes.

Now you're ready to get to work.

## Styling the Sidebar: Work from the Outside In

A useful technique when styling elements with HTML and CSS is to work from the "outside in." This means that you start with the outermost container within an area of the page and work your way inside to get all the individual styling rules set for the items *inside* the container. You'll use this technique to style the sidebar of the design and the navigation elements that will be placed in this area of the page.

The client has stated that the solid color of the sidebar is too plain and has supplied a replacement image. Your first task is to replace the image that provides the sidebar background.

- 1 Open the CSS Designer panel, and select the `#container` selector.

Previously in the `background-image` setting, an image was set here to create the faux column look for this design. You'll change this to one that the client provided.

- Click the folder icon next to the url field for the background-image setting (Figure 5.3). Browse to the new container-bg-gold-lines.png image, and set that as the background image for the #container selector.

You'll now see the new image in the sidebar and get another clue as to the design the client is looking for.

- Select the #container selector.
- Browse to the replacement image file.

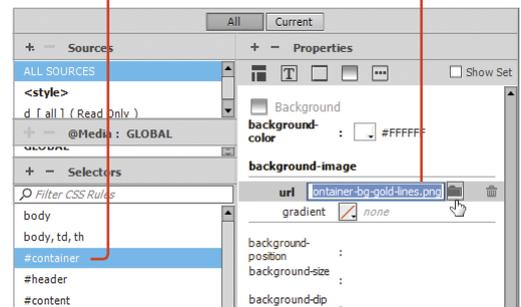


Figure 5.3 Changing the background image for the #container selector

## Modifying the Sidebar Margins and Adjusting Widths

Now the text in the sidebar area of the page is too crowded and pushes up against the borders of the container. You'll address this by setting new margin properties for the #sidebar selector. Switch to Design view for these operations.

- In the CSS Designer panel, select the #sidebar selector, and go to the margin settings in the Layout category. Apply margins of 10 px to the left and right of the container (Figure 5.4).

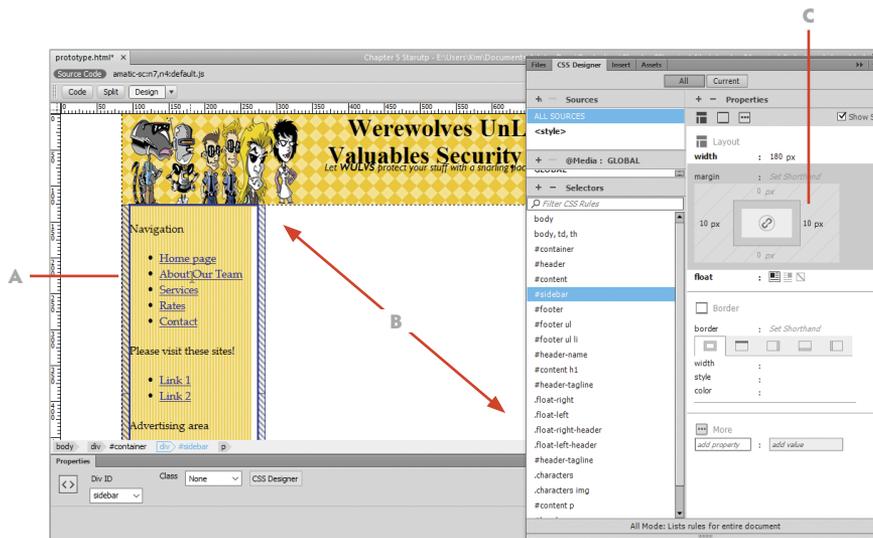


Figure 5.4 Applying margins to the #sidebar selector

- Left and right margins applied
- Content <div> slips below the sidebar div.
- Margin settings in the CSS Designer panel

When you apply this setting to the sidebar, the `#content <div>` no longer has room to sit side by side with the sidebar and it slips down under the element that comes before it. To adjust this you'll turn to the `#content` selector.

- 2 Select the `#content` selector in the CSS Designer panel. In the Layout category, adjust the container's width until the `#content <div>` snaps back into position.
- 3 Select the `#sidebar` selector. In the Layout category, change the width to 160 px.

## ARE FIXED-WIDTH DESIGNS EVIL?

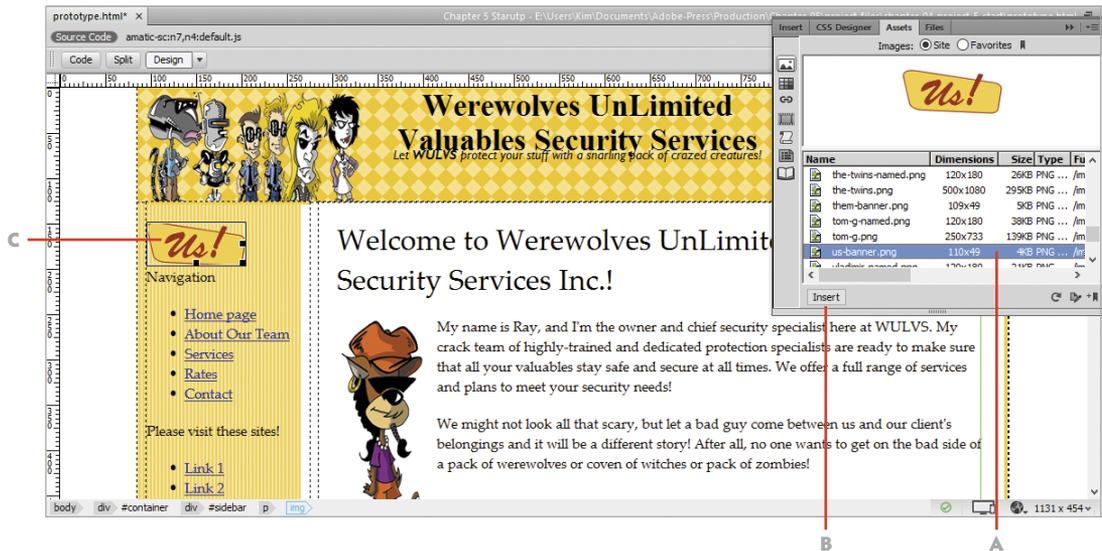
Throughout this project, you'll be using a fixed-width design as you learn how to use CSS layout principles and the tools that Dreamweaver provides to visualize your design. This isn't the way that modern web pages are designed, where a page needs to have flexible design that scales based on the size of the viewport.

You'll learn about flexible and responsive designs in Chapter 7, but while you're learning about CSS, layouts with a fixed-width design, like the one used in these projects, make it easier for you to see and understand how positioning is done with pure CSS designs. That's not evil, but it probably isn't the choice you'd make for a site that is intended for publication.

## Inserting the Sidebar Images

Chris the Cartoonist has provided some nifty images that he wants to use above each of the navigation sections in the sidebar. Not only do these images provide another clue to the design that Chris is after, but he has also asked that the color values used in the images be repeated within the site. Once the images are in place, you'll be able to "pick up" those color values for use in border, text, and other values you'll use as you complete the prototype design.

- 1 If it's not already visible, choose Window > Assets from the application menu to open the Assets panel. Set the view of the document to Design view.
- 2 Place your cursor next to the text *Navigation* at the top of the sidebar.
- 3 In the Assets panel, select the `us-banner.png` file (Figure 5.5), and click the Insert button to insert the image.



**Figure 5.5** Use the Assets panel to locate images to be inserted onto the page.

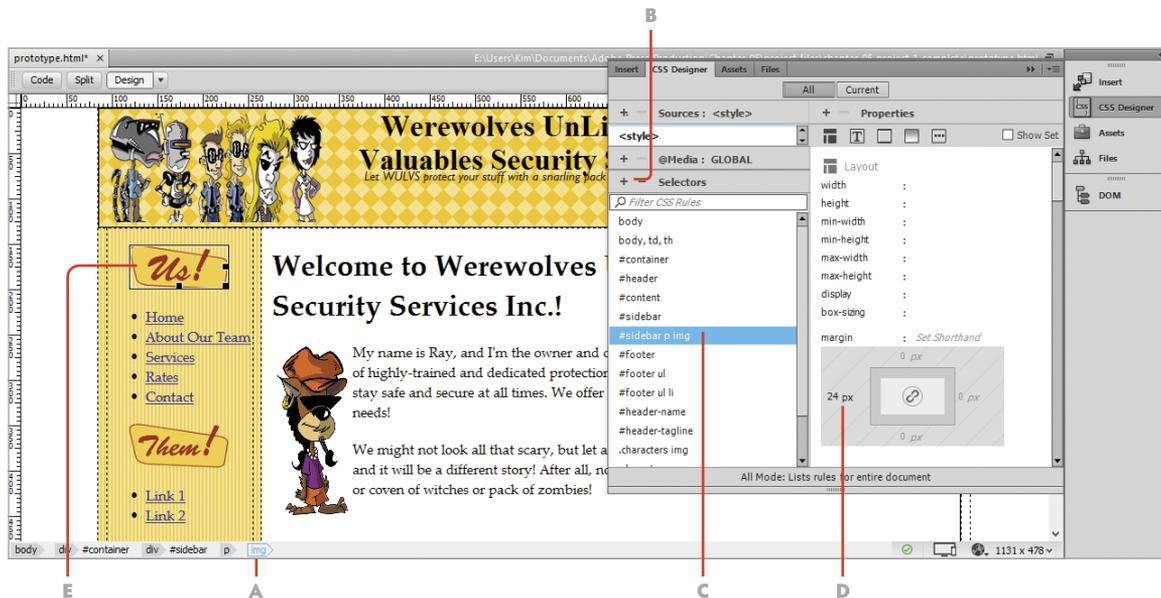
- 4 Delete the *Navigation* placeholder text.
- 5 Place your cursor next to the placeholder text *Please visit these sites!*
- 6 In the Assets panel, select the `them-banner.png` file, and click Insert to insert the image.
- 7 Delete the placeholder text.
- 8 In the Property inspector, set the Alt text for each image as follows:  
`us-banner.png`: WULVS Navigation  
`them-banner.png`: Our Friends

- A Image selected
- B Insert the image at the cursor position.
- C Image inserted into the sidebar

## Positioning the Sidebar Images

To position these two images inside the sidebar container, you need a new CSS selector. The CSS Designer panel makes this a snap (Figure 5.6).

- 1 Select the `us-banner.png` image, and check the Tag selector to be certain the `<img>` tag is selected.
- 2 Click the + in the Selectors area of the CSS Designer panel. Refine the selector name until it reads `#sidebar p img`. Press Enter/Return to accept the selector name.



**Figure 5.6** Create a new CSS Selector to position the images in the sidebar.

- A** Select the `img` tag in the Tag selector.
- B** Add a new selector.
- C** Name the selector `#sidebar p img`.
- D** Apply left margin properties.
- E** Image positioned within the sidebar container

- 3 In the Layout category of the CSS Designer panel, apply a left margin property of 24 pixels.
- 4 Check the position of the images, and when you're satisfied with the result, save your file and preview the results in a browser.

You can fine-tune this setting later, but the goal here is to get the image centered inside the sidebar container.

This first project provided you with a good review of the principles of web design as you evaluated the client's requirements and compared that with the layout of the page. Based on client feedback, you revised the sidebar using the new background image that was provided and put some new images into position. As you completed these steps, the look and feel that the client is after came into focus. In the next project, you'll convert the list of links into interactive rollover navigation buttons.

## PROJECT 5.2

# Styling Navigation Elements

In the last project, you began the process of styling the left sidebar for the pages in this site, and the client's vision is starting to come into clearer focus. That hipster vibe that Ray the Werewolf seems to be going for is starting to show up with lighter colors, the striped background for the sidebar, and those groovy shapes for the two images that were inserted.

And one of the more essential elements of any web design is the user experience created by the navigation. Literally millions of ways to design navigation bars and buttons are possible, but in this project you'll learn the fundamental methods for creating site navigation. These methods involve unordered lists, the list items that are within them, and a new breed of CSS selector known as the pseudo-class. Let's get started.

★ ACA Objective 2.4

★ ACA Objective 4.2

★ ACA Objective 4.5

▶ **Video 5.2** Styling Navigation Elements

### LEVEL UP: ALTERNATIVE DESIGNS!

You have to wonder if the other employees at WULVS had a different idea how this design should be done. The vampires probably would have wanted something darker and more gothic, Robbie the Robot might have argued for more metallic elements, and the Twins might have preferred a more preppy look. But that's the beauty of how web design is done. Because the contents of the prototype stay the same, you only have to define a new CSS style sheet and swap out a few images to change *everything* about the look of this design.

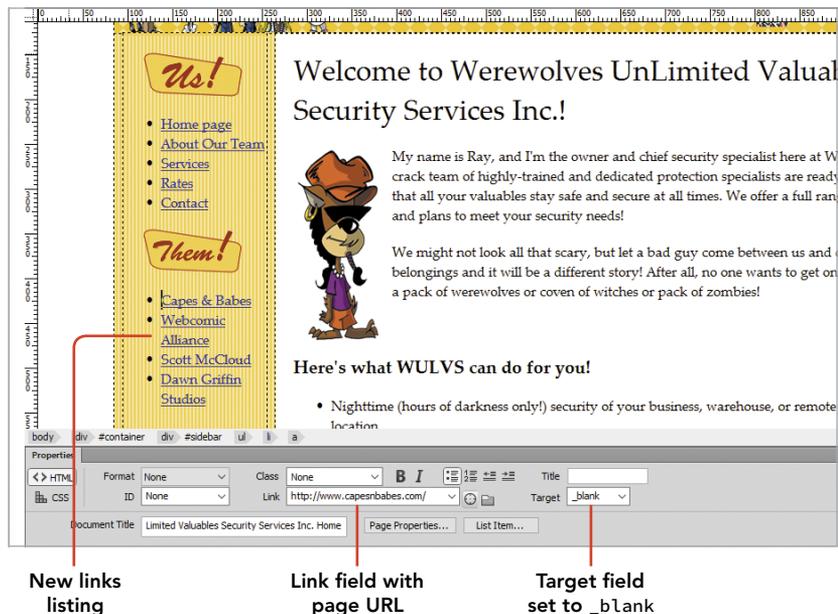
As you work through the design choices in this chapter, you might want to imagine that one of the other characters in the comic was put in charge and how that might affect the final design. How would you go about designing a dark and scary theme or a design that another character would approve of?

## Adding Links to Other Pages

The client has provided the names of the websites that he'd like to link to under the `us-banner.png` file. You'll use these values to replace the placeholder links that are currently in that area of the page.

- 1 Open `WULVS-Things-To-Do.docx` from the Files panel, or open `WULVS-Things-To-Do.rtf` if you want to use a text editor other than Microsoft Word.
- 2 Locate the sites within `WULVS-Things-To-Do.docx` that the client wants his site to link to:
  - Capes & Babes: `http://www.capesnbabes.com`
  - Webcomic Alliance: `http://webcomicalliance.com`
  - Scott McCloud: `http://scottmccloud.com`
  - Dawn Griffin Studios: `http://dawngriffinstudios.com`
- 3 Change the first item in the sidebar list from *Link 1* to **Capes & Babes**.
- 4 In the Link field in the Property inspector (Figure 5.7), type `http://www.capesnbabes.com`.
- 5 Set the target for the link to **\_blank** so that the page opens in a new browser window or browser tab when it is selected.
- 6 Repeat steps 3 through 5 for the *Link2* placeholder link, using the URL for the Webcomic Alliance.
- 7 Press Enter/Return to create a new list item. Repeat the process of adding new list items and linking to the other pages until all four links have been set (Figure 5.7).

**Figure 5.7** Adding the new list of links in the sidebar



## Inserting a Navigation Tag

The navigation elements in this page will use a new tag that you haven't encountered yet. You can designate a navigation area using a `<nav>` tag, which is read by assistive devices in such a way that it's clear the links are for site navigation. The `<nav>` tag is also used in responsive design and HTML5 layouts because it can use properties that a regular `<div>` tag cannot. Next, you'll use that `<nav>` tag to wrap the existing list of links.

- 1 With Design view active, click anywhere within the first list of links, and then click the `u1` indicator in the Tag selector. The entire unordered list will be selected.
- 2 Open the Insert panel with the HTML category selected, and then click the Navigation button located near the middle of the list of items.

The Insert Navigation dialog appears.

- 3 Click the New CSS Rule button.

The New CSS Rule dialog displays (Figure 5.8).



**Figure 5.8** You can create new named selectors using the New CSS Rule dialog.

Create the new class with the name `.sitenav`.

- 4 Assign the name `.sitenav` to the new class rule, and click OK.

The CSS Rule Definition dialog appears. You'll define the properties of this element in the CSS Designer panel, so you don't need to set any properties here.

- 5 Click OK.

You are returned to the Insert Navigation dialog.

- 6 Click OK to accept this new selector name.  
The list of links is now contained within a new `<nav>` tag with a class attribute of `.sitenav`.
- 7 Repeat steps 1 through 6 to wrap the second unordered list in the sidebar with a `<nav>` tag. This time when the Insert Navigation dialog appears, you only need to assign the `.sitenav` class to the new tag using the Class menu and click OK. Your second set of links is now contained in a navigation container.

## Removing Default List Properties

In the next few steps, you'll remove the default styling properties that are assigned to HTML list items. You need to change the appearance of the items in the list to turn them into interactive buttons.

The first order of business is to remove the bullets next to the items in the list.

- 1 Click anywhere within a sidebar list, and select the `ul` indicator in the Tag selector.
- 2 Click the + in the Selectors section of the CSS Designer panel. Set the name of the new rule to `.sitenav ul`.
- 3 With the new rule selected, go to the Text category of the CSS Designer panel, locate the `list-style-type` property, and set the value to **none**.

The bullet symbols next to each list item in the sidebar disappear. Cool!

## Assigning Width and Margin Properties to the List Items

Now, you need to move the list of links a little closer to the left edge of the navigation area by removing the default margins that HTML applies to items in a list. You'll also work on the position and spacing of the individual items in the list.

- 1 Click anywhere in the list of links, and use the Tag selector to choose the `li` tag indicator.
- 2 Click the + in the Selectors section of the CSS Designer panel. Set the name of the new rule to `.sitenav ul li`.
- 3 With the new rule selected, set the width to **160** pixels in the Layout category.
- 4 Set the left margin to **-40** pixels.

This removes the default margin that list items contain.

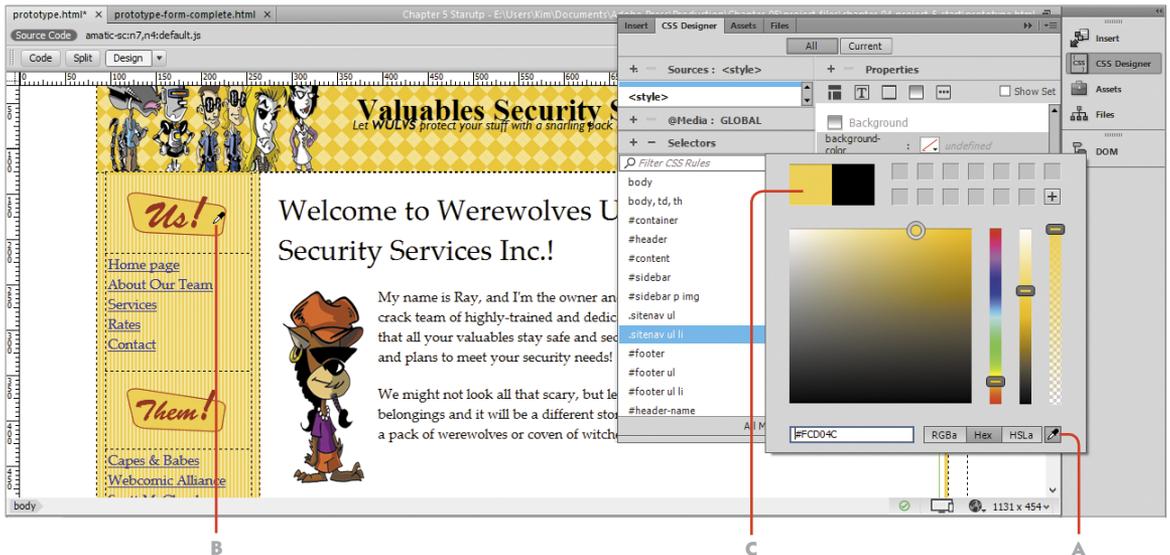
## Choosing a Background Color from an Image

The client asked that you use the colors in the sidebar area within other areas of the design. You can use the Eyedropper tool to pick up a color value for the background of the list items. And having a color in place will make it easier to see how other properties are set as you style the buttons.

- 1 With `.sitenav ul li` still selected, go to the Background category of the CSS Designer panel. Click the `background-color` setting.
- 2 In the lower-right corner of the color chooser is a tiny Eyedropper icon. Click the Eyedropper and position your cursor over the yellow-gold area of the image in the sidebar (Figure 5.9).
- 3 When you see the color in the preview area of the color chooser, press Enter/Return. The sampled color value will be applied to the `background-color` setting and perfectly match the color in the image.

With the background color set, you can more clearly see the spacing between the individual list items.

- 4 Apply a little top and bottom padding to the list items to spread them apart.



**Figure 5.9** The Eyedropper tool allows you to “pick up” a color from any visible element on the page.

- A** Select the Eyedropper tool.
- B** Position the Eyedropper over the image at the top of the sidebar.
- C** The selected color is displayed in the color chooser.

## Setting the Link Values Using Pseudo-Class Selectors

### TIP

You can quickly jump to any CSS rule by right-clicking the selector name and choosing Go to Code. Dreamweaver will jump you to the last line of the selected rule, and you can get to work on the code.

A pseudo-class is a special kind of CSS selector that is triggered by an action that takes place on the screen. In the case of navigation elements, you can use these special selectors to create a rollover effect that changes the appearance of the link when the viewer's mouse moves over the top of it. You'll use these selectors with the `<a>` tag, which is the rule you'll define next—with just a little twist. Pseudo-class selectors must be written in a specific order.

- 1 Click anywhere in the list of links, and use the Tag selector to choose the `<a>` tag indicator.
- 2 Click the + in the CSS Designer panel to create a new selector, name it **.sitenav ul li a**, and press Enter/Return.

The syntax of pseudo-class rules is very specific and best done in Code view for the page.

- 3 Locate the **sitenav ul li a** rule you just created. It is in the `<head>` of the document.
- 4 Change the name of the selector to **.sitenav ul li a:link, a:visited** by typing directly in the code.

Your first set of pseudo-class selectors is now in place.

- 5 Select the **.sitenav ul li a:link, a:visited** selector in the CSS Designer panel.
- 6 In the Layout category, set the `display` to **block**.

This makes the entire container that contains the link active and clickable, rather than just the text itself. Now, you need to remove the underlines.

- 7 Set `text-decoration` to **none** and the underlines disappear.

Now you have some styling work to do.

- 8 Set the following properties to create rounded buttons with a red border that matches the red in the image the client provided. Can you locate the parts of the CSS Designer panel where you would set these properties and values?

```
font-family: "Gill Sans", "Gill Sans MT", "Myriad Pro",  
→ "DejaVu Sans Condensed", Helvetica, Arial, sans-serif;  
color: #000000;  
font-size: 18px;  
border: 1px solid #AC1307;
```

```
border-radius: 10px;
padding-left: 7px;
padding-bottom: 3px;
padding-top: 2px;
background-color: #FCD04C;
```

- 9 Once your buttons are set, you can remove the background color that was applied to the list items themselves. Click the `.sitenav ul li` rule in the CSS Designer panel, and click the trashcan icon next to the `background-color` property to remove it.

Switch to Live view and you'll see that the result of all this work is beautifully styled rounded buttons (Figure 5.10). Do yours look like these? If they do, it's time to create the hover effect. If you're not happy with your results, return to the CSS Designer panel and make adjustments until you are.



**Figure 5.10** The result of styling the links in the navigation area of the sidebar

## Setting the Hover Values Using Pseudo-Class Selectors

The selector you just finished provides the styling of the buttons when they are in their “up” state, where the mouse has not rolled over the link. To make the rollover effect work, you'll use a second set of pseudo-class selectors to control the hover and active states of the link.

- 1 Duplicate the existing rule named `.sitenav ul li a:link, a:visited`.
- 2 In the code, change the selector name to `.sitenav ul li a:hover, a:active`. Make sure this new selector is listed after the first set of pseudo-class selectors so that it falls into the correct LVHA order.

You don't need the full list of values repeated for the hover state—only those that you want to change when the mouse encounters the button.

### LOVE MAKES ME HAPPY

Pseudo-class selectors must always be listed in a particular order: link, visited, hover, and active (LVHA). It used to be that the way you'd remember the order is by remembering Love Hate. But I prefer to use the mnemonic device *Love makes me happy* to remember the correct order.

- 3 Delete all the extra properties so that only the background color remains. Change the background color to a lighter shade than the original color value. You can leave the block designation in place, but your final rule for this selector should appear similar to the following:

```
.sitenav ul li a:hover, a:active {  
    display: block;  
    background-color: #FEF4D5;  
}
```

- 4 Save your file, and preview the page in a browser.

Cool, right? The buttons change colors as you roll over them, adding a whole new level of usability and interest to this design.

In this project, you've learned how to use pure CSS styling rules to create interactive, highly attractive navigation buttons that respond to the position of the mouse on the page. No additional programming was required—no JavaScript or any other technology besides CSS pseudo-classes and the `<a>` tag on the page. This is a powerful method for designing navigation elements and is used on countless websites to create all sorts of visual effects.

## PROJECT 5.3

# Styling Structured Data

Human beings respond well to information that is organized. Every day, we see data presented in rows and columns. From a listing of the high and low temperatures on any given date, to a teacher's grade book, to financial statements and personal budgets found in spreadsheets, we all are accustomed to seeing ordered and structured data.

In HTML, structured data is presented using the `<table>` tag, the `<tr>` tag (which creates rows), and the `<td>` tag (which defines each individual cell inside the table).

In this project, you'll work with tables that the client has provided and see how to provide spacing and borders. You'll create a new CSS selector to style the alternate rows in a table.

★ ACA Objective 2.4

★ ACA Objective 4.6

▶ Video 5.3 Styling Structured Data

## Cleaning Up a Table from Microsoft Word

The client supplied a table in a Microsoft Word document. You need to do a bit of cleanup before you can use the table in your web page.

- 1 Open `prototype.html`, and scroll to the bottom of the page, where you'll find two tables.

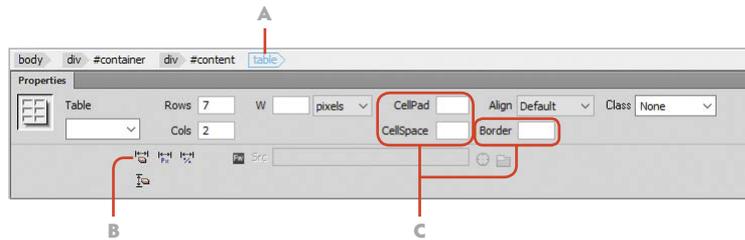
The Services table was inserted using the Dreamweaver Insert Table command, but the Rates table was copied and pasted from the client's Word document.

Even though these tables came into being in different ways, the structure is the same. The `<table>` tag defines the table, the `<tr>` tag defines each row of the table, and the `<td>` tag is used for each cell within the table. A fourth tag, the `<th>` tag, is used to define the table heading and appears within the first row of the table.

Your first task is to clean up the HTML in the Rates table. This involves removing any markup that Word included when the table was copied into place.

- 2 Click anywhere within the Rates table, and click the table tag indicator in the Tag selector (**Figure 5.11**) to select the entire table.
- 3 Click the Clear Column Widths button to remove the `width="312"` attribute from all the `<td>` tags.

**Figure 5.11** The Property inspector for the Rates table



- A** Select the table indicator.
- B** Click the Clear Column Widths button to clear column widths.
- C** Remove the values in these three fields.

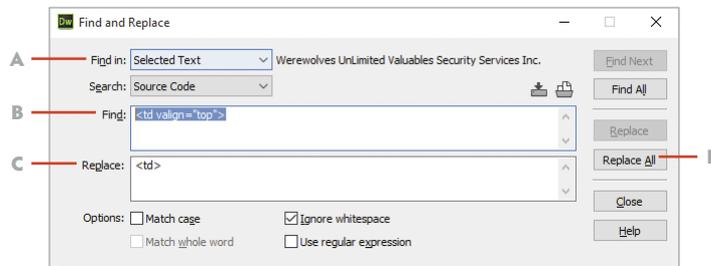
- 4 Delete the values in the CellPad, CellSpace, and Border properties in the Property inspector.

When your settings match those in Figure 5.11, you have cleared the table widths and collapsed the space between the table cells.

Next you'll use the Find and Replace feature to remove the `valign="top"` attribute from all the `<td>` tags within the tables.

- 5 Directly in the code, highlight one of the complete `<td valign="top">` tags within any of the tables, and copy the selection.
- 6 Select the `table` indicator in the Tag selector, and press CTRL+F/Command+F to open the Find and Replace dialog (Figure 5.12).

**Figure 5.12** You can use the Find and Replace dialog when you need to replace specific tags within a document.



- A** Choose Selected Text from the drop-down menu.
- B** Paste the copied tag into the Find field.
- C** Type `<td>` into the Replace field.
- D** Click the Replace All button.

- 7 Paste the copied tag into the Find field.
- 8 Type `<td>` into the Replace field.
- 9 With the Find In setting set to Selected Text, click the Replace All button and the vertical align attribute will be removed from all the `<td>` tags.

To convert this Word table into a semantically correct HTML table, change the `<td>` tags for each column of data to `<th>` tags.

- 10 Change the `<td>` tags that are at the top of each of the columns into `<th>` tags by typing directly in Code view, making sure that you have both tags closed.

11 The final step to get this table ready for styling is to remove the paragraph tags that were inserted by Word. Click inside the top-left cell in the table (Service Name), then drag to the lower-right corner of the table until all the individual cells are selected.

12 In the Property inspector, set the format for the selected cells to None.

This table can now be styled right along with the other table in the page without any conflicts from properties that Word created.

## Setting Table and Table Cell Properties

Next, you'll set the table width and apply border and padding properties for the table's contents. Time to flex your coding muscles!

- 1 Create a new selector named **#content table**.
- 2 Set the width property of **#content table** to **750 px**.
- 3 Create another new selector named **#content table, td, th**.
- 4 Set the properties for this selector to apply the borders, to collapse the borders, and to add a little padding around the contents of the containers within the table. Your final code block should be similar to the following:

```
#content table, td, th {  
    border: 1px solid #000000;  
    border-collapse: collapse;  
    padding-left: 13px;  
    padding-right: 9px;  
    padding-top: 3px;  
    padding-bottom: 3px;  
}
```

## Setting Column Widths and Heading Properties

You can control the width of the columns in an HTML table by setting a width value for the divisions within the first row of the table. In these steps, you'll create two new classes to apply to the `<th>` tags at the top of each table.

- 1 Create a new selector named **#content table.column1**.
- 2 Select the first `<th>` tag in each table, and use the Property inspector to assign the `column1` class to the tag.

- 3 In the Layout category of the CSS Designer panel, set the width for the class selector to **250 px**.  
The left column snaps to this dimension, and the other column takes up the remaining space.
- 4 In the Text category, use the Eyedropper tool to set the text color to match the red color in the images at the top of the sidebar.
- 5 Modify the size of the text to **120%**.  
This setting makes the text 120 percent larger than the default text size used in the document.
- 6 Duplicate the rule you just created, and name it **#content table .column2**.
- 7 Remove the selector's width value.
- 8 Select the second `<th>` tag in each table, and assign the `column2` class to the tag in the Property inspector.

Your tables now have a nicely labeled heading at the top of each column.

## Styling Alternate Rows in a Table

The client requested that the table rows have alternating background colors to make the tables more readable. Lucky you can do this easily because the browser automatically counts each row within a table. This allows you to use a new CSS selector type—the **child selector**—to assign a specific property based on whether the row is an odd or even number. This is an easy styling trick to perform because the computer does all the work for you.

The syntax of this selector is quite specific, so it's best to go directly to the `<head>` of the document and create the new selector in the style sheet code.

- 1 In Code view in the `<head>` section of the document, create a new selector named **#content table tr:nth-child(odd)** directly below the `#content table.column1` selector.
- 2 Return to the CSS Designer panel, and apply a medium gray background color to this selector.

The odd-numbered rows in the table, starting with the table header, have the background color applied, when seen in Live view.

- Return to the `#content table.column1` and `#content table .column2` selectors, and assign a background color to the table headings. The gold color in the sidebar makes an excellent choice, allowing the headings to really stand out while remaining consistent with the rest of the design (Figure 5.13). Setting font-weight to bold will also make the text pop.

Rates Page	
Service Name	Price and Payments Accepted
Basic Overnight Security	\$100.00 per week. Certain discounts apply based on the availability of the client's neck
Round-The-Clock	\$200.00 per week. See note above. Daytime discounts may be applied where the client is able to provide raspberry jam and quantities of wood pulp and motor oil.
Digital Demon™ Online Security	\$5.95 annually. Smack on the Back of the Head service is priced at \$0.25 per instance of inappropriate behavior. Payment plans available.
Hauntings	\$75.00 per night. Money-back guarantee if the individual does not change the specified behavior.
Spells and Potions	Priced per potion, from \$3.95 for basic sneezing spells, to \$29,495.00 for guaranteed love potions.

**Figure 5.13** The result of your work in Project 5.3

You're one step closer to finishing the design work for Chris the Cartoonist. In this project, you learned how to take a table created in a Microsoft Word document and remove all the Word-specific markup. You then created the styling rules that established the borders, width, padding, and background colors for the table and the table headings. You even applied a cool little trick that lets you provide alternating row colors to the table—just as the client requested.

## PROJECT 5.4

# Building and Designing Interactive Forms

★ ACA Objective 2.4

★ ACA Objective 4.6

▶ **Video 5.4** *Building and Designing Interactive Forms*

It would be pretty unusual for a company or organization in this day and age not to provide a way for potential customers and clients to contact them from their website. Contact forms and other forms that let the customer request more information, report a problem, or even submit answers in an opinion poll are accomplished through HTML forms.

In a regular production setting, an interactive form (**Figure 5.14**) would be submitted to a web server where the information the customer sends is stored and processed. For real live commercial websites, the company that is hosting the website on their servers would provide instructions on how their e-mail processing works, and how to set the form in Dreamweaver to work on their servers.

**Figure 5.14** At the end of this project, you will have a nicely styled, functional HTML form.

**Contact Page**

Thank you for your interest in WULVS!

We stand ready to serve your security needs and *we're waiting to hear from you!*

Please complete the form that you see below and someone will get back to you **within 24 hours**.

Fields marked with an \* are required

\* Your Name:

\* E-mail Address:

Phone:

\* Subject :

\* Your Message:

Lucky for you—since you don't have a real live server to work with—your client wants this form to point to a web page where the potential visitor would be reminded that this isn't a *real* company and you can't *really* hire werewolves and vampires to guard your home. This page will allow you to work through the process of building the form and setting the form to open a web page.

Oh, and the styling? This time you're on your own. In this project, you'll build the form and learn which tags you can target for styling, but creating and applying the styling rules will be up to you.

## Inserting Text from Microsoft Word

The client supplied text that he wants to appear on the Contact page. Let's insert that text now.

- 1 Open `client-feedback.docx`. Locate the following block of text, and copy it. Notice the styling properties the client used:

**Thank you for your interest in WULVS!**

We stand ready to serve your security needs, and *we're waiting to hear from you!*

Please complete the form that you see below, and someone will get back to you within 24 hours.

- 2 Open `prototype.html`, and replace the text *The form will go here* with the text you copied from `client-feedback.docx`.

Depending on the version of Word that you're using, Word may insert extra markup when you paste text into your HTML page.

### WHAT'S SO WRONG ABOUT THE MAILTO: LINK?

Once upon a time, it was standard procedure to create links that would open the viewer's default e-mail program so that a visitor could fill out a blank message and send the message to the specified address. This was done by inserting a special link that used `mailto:` to send the message, like this: `mailto:info@wulvs.com`.

But this method has all sorts of problems. First, it depends on the visitor having an e-mail program set as the default handler for mail on their computer. Nowadays, with so many people using web mail from Google and Microsoft and Apple and others, many people don't have a dedicated e-mail program on their computer.

Second, every moment of every day, special little programs known as spiders are searching web pages looking for that `mailto:` indicator. These little buggers harvest the e-mail address and that mailbox starts getting all sorts of interesting offers. So unless you really love spam, listing e-mail addresses using a `mailto:` link isn't a good idea. Modern web pages use forms instead and avoid both of those problems.

- 3 Check the code carefully to ensure that only the text and the `<strong>` and `<em>` tags are present, bolding the text inside the `<strong>` tags and creating italicized text with the `<em>` tag. Clean up the text as necessary.
- 4 In the Property inspector, format the first line of text as `<h3>`, and make sure the other lines of text are contained in paragraph tags.
- 5 Add the text **Fields marked with an \* are required.** after the last line that was copied into place.

Now you're ready to build your form.

## Insert the Form Tag and Text Fields

Just like everything else in HTML, a form is composed of tags that provide specific information to the browser. In the steps that follow, you'll begin building your form by inserting the necessary tags. Set the view of the document to Design view to complete these steps.

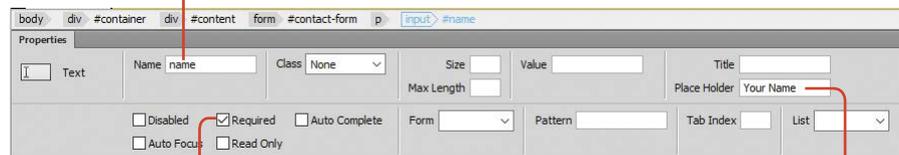
- 1 In the Insert panel, switch to the Form category using the drop-down menu at the top of the panel.

HTML forms are always enclosed within the `<form>` tag.

- 2 Click the Form button at the top of the Insert panel.
- 3 In the Property inspector, assign the ID **contact-form** to the `<form>` tag.
- 4 Click inside the red dashed line that marks the boundaries of the `<form>` tag. From the Insert panel, insert a Text object inside the `<form>` tag.
- 5 In the Property inspector, assign the Text object the ID of **name** in the Name field, mark the object as a required field, and set the Place Holder field to **Your Name** (Figure 5.15).

Set the name of the object to name.

**Figure 5.15** Use the Property inspector to assign properties to the Text object in your form.



Select this checkbox to make the field required.

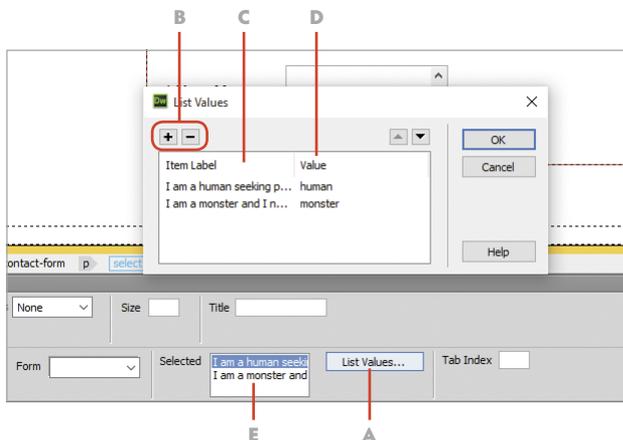
Provide the text that will appear to viewers before they enter their own information.

- 6 Replace the text that Dreamweaver provides (Text Field:) with **\* Your Name:**.
- 7 Create a new blank line by pressing Enter/Return, and insert a second Text object from the Insert panel. Name this object **mail**, make it a required field, and set the Placeholder Text field to **E-mail**.
- 8 Replace Text Field: with **\* E-mail Address**.
- 9 Add a third text field named **phone**. Enter **Phone:** as the replacement text. This field is not required, so leave the Required checkbox unselected.

## Insert Additional Form Elements

The next form element you'll insert will provide viewers with a drop-down menu where they can choose from a list of options when they complete the form.

- 1 In the Insert panel, insert a Select form object in a new paragraph after the phone field.
- 2 Name this form object **subject** and make it a required field.
- 3 In the Property inspector, click the List Values button. The List Values dialog displays (**Figure 5.16**).
- 4 Set two list labels and values as follows:
  - Label 1: **I am a human seeking protection**
  - Value 1: **human**
  - Label 2: **I am a monster and I need a job!**
  - Value 2: **monster**
- 5 Click OK to close the List Values dialog.



**Figure 5.16** The values for a drop-down list are set in the List Values dialog window.

- A** Open the List Values dialog.
- B** Add or remove list values.
- C** List item labels
- D** List item values
- E** The top item in the list is the first to appear on the page.

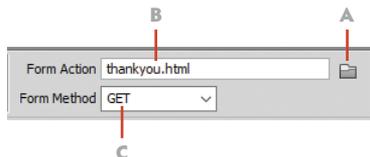
- 6 Change the text label for this field to **\* Subject:**.
- 7 On a new line, insert a Text Area object from the Insert panel. Name this object **message** and make it a required field. Set the Placeholder text to **Your Message**.
- 8 Change this text label to **\* Your Message:**.
- 9 On a new line, insert a Submit Button item from the Insert panel. In the Property inspector, you will see that this object has the ID of **submit** already applied to it.

## Adding Function to the Form

The client requested that someone who submits this form be taken to a new web page where they'll see a message reminding them that this isn't a *real* website and you can't *really* hire werewolves and vampires to guard your property. You'll create a new page and then hook the Submit button to the page so that it opens when the form is submitted.

- 1 In the Files panel, right-click the folder at the top of the panel (the site folder), and select New File from the context menu.
- 2 Name the new file **thankyou.html** and press Enter/Return to accept the filename.  
  
You'll add content to this page later on, but the filename is needed now to make the form function correctly.
- 3 Select the Submit button. In the Property inspector, use the Browse for Folder icon next to the Form Action field to browse to **thankyou.html**.
- 4 Set Form Method to GET. This setting will cause the page to load into the viewer's browser when the button is clicked (**Figure 5.17**).

**Figure 5.17** Setting the functional properties of the Submit button



- A** Click to browse to **thankyou.html**.
- B** Path to the file that will open
- C** Set Form Method to **GET**.

- 5 Save your file, and preview the page in a web browser.  
  
Your form is now fully functional (but not very stylish), and when you click the Submit button you'll be taken to the blank Thank You page. Congratulations, you made a form!

## Styling the Form

Your form is now functional, but it certainly needs some styling work. But you've become so proficient in this process that you can complete this styling work on your own. Following is a list of things to do and some tips on what selectors you'll need to create to get the job done.

- 1 Create a new class named **.required**. Set the styling for text on this class so it has a red color and the font weight is bold.
- 2 Highlight each of the asterisk symbols in the form area of the page and use the Property inspector to assign the **.required** class to the selection.  
This will wrap the symbol in a `<span>` tag, and the symbol will appear with the red, bold text you defined with the class.
- 3 Create a new selector named **#contact-form** that targets the form itself. Use this selector to style the border of the form, its width, and its margins. If you wish to center the form inside its container, you can set the left and right margins to **auto**.
- 4 Create a new selector named **#contact-form label** to position all the labels within the form away from the left border. You can use padding properties with this selector to position the left edge of each of the text labels.
- 5 Create individual selectors for each named field to position the input fields next to their labels. The name field, for instance, will require this selector: **#contact-form p #name**. As you create each new selector, use margin properties to align all the form fields so they're nice and neat like the example in Figure 5.14.
- 6 Use margin and padding values on a selector named **#contact-form #submit** to place the Submit button in a location that is consistent with your design.

As you work, be certain to preview your styling choices both in Dreamweaver's Live view and in a web browser. By the time you're finished, you should have a beautifully styled and functional form.

## PROJECT 5.5

# Finalizing the Prototype for Production

★ ACA Objective 1.3

★ ACA Objective 5.1

▶ **Video 5.5**  
*Finalizing the Prototype for Production*

It might be hard to believe, but you're nearly ready to put the WULVS.com website into production. Do you remember how you started this mission with just a little information from the client and some very basic HTML? You've come a long, long way.

Now you're ready to tend to the final housekeeping that every website production requires. This involves a careful review of the client's requirements and tending to any final details, then doing some technical tests on the prototype to check for errors and finishing up the design. The goal in this final project is to get the prototype file ready to present to the client for final approval. From there your site will be ready to put into production.

This is a good time to carefully check the complete list of requirements that the client provided. Create a checklist of every item the client requested. Work your way down the checklist to make certain every detail has been covered. This checklist will help prevent last-minute changes that might delay your client's final okay. In many design contracts, the final client approval is an important milestone to reach so you can get paid, and you don't want to delay that!

## Adding the Advertisement Block

After running down the checklist, you'll find that an important part of the design was missed in all the design work done to this point. The advertising block where the client wants to place ads is missing. Let's tend to that item.

The client expects to add more advertising as time goes by, so this container needs to have some flexibility but still be consistent with the rest of the design elements.

- 1 With `prototype.html` open, go to Code or Split view, and locate the placeholder advertising text that is in the sidebar of the page.
- 2 Delete all the placeholder text and the paragraph tags that contain the text.
- 3 Working in Code view for the document, place your cursor directly after the closing `</nav>` tag after the list of links in the sidebar, and create a new

`<div>` with the class set to `side-advert`. Close the tag, and add the text **placeholder** to pop open the `div`. Your completed code block should appear as follows:

```
<div class="side-advert">
  placeholder
</div>
```

- 4 In the CSS Designer panel, create a selector named **#sidebar .side-advert**.
- 5 Using this new selector, assign a width of **160 px** to the container, and set the borders so they match the color, width, and radius of the buttons that are above this area. Make the background color white, and set a minimum height to fill out the shape of the container. Once these styling rules are set, you can remove the placeholder text.

The client provided a sample advertising image from the fine folks at BrainBuffet.com. This will allow you to set up a sample ad for the client's review.

- 6 Click inside the `side-advert` area, and enter **Learn More About Adobe Design Tools at BrainBuffet.com**. Set the formatting for this text to a paragraph format.
- 7 Highlight all the text, and insert the URL **<http://brainbuffet.com>** into the Link field of the Property inspector. Set Target to `_blank`.
- 8 Insert a new paragraph, and from the Assets panel insert the file named **brainbuffet-logo.png**. Don't forget to enter the Alt text.
- 9 Link the image to the same URL you used in step 7, and use the same `_blank` value so the link opens in a new window.

Now it's time to provide additional styling for the elements within the container. As you review the structure of this content, it should be easy for you to pick out the potential styling hooks. The `<a>` tag, for instance, can be styled against to remove the default underline for the text link and assign a new color for the text. As you complete your own design work, your goal is to have an area in the sidebar where new advertising content can be placed in the future with design properties that are consistent with the rest of the page. **Figure 5.18** shows an example of how this area might be styled, but you are free to come up with ideas of your own.



**Figure 5.18** The completed advertising block should be consistent with the other design elements of the sidebar.

## Fixing Up the Fonts

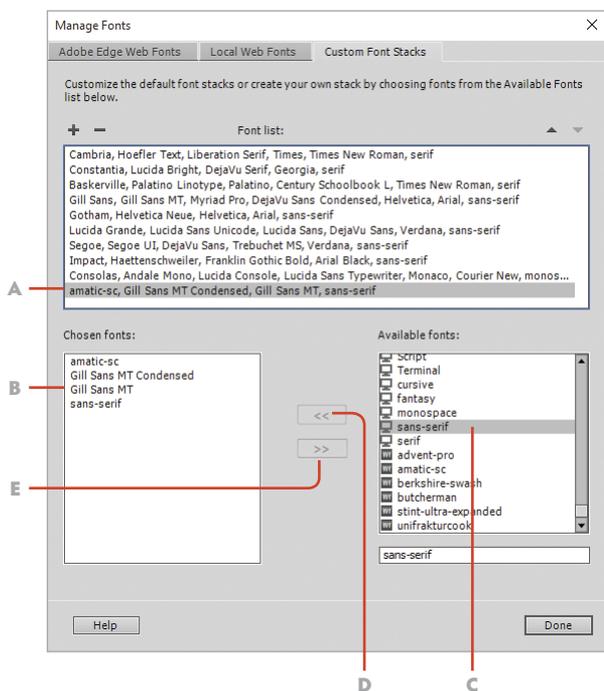
Readability is a critical component for web pages, and the client has also asked that you experiment with the same font that is used in the header to see if it works in other areas of the page. This is a good time to review the fonts that are in use in your prototype, resolve a potential issue that might come up with the web font you're using in the header, and check that your design is consistent with sound design principles.

### NOTE

You may have a slightly different list of fonts than the ones seen here if you are using a Mac.

The **web font** that's used in the header of the page is just what the client is looking for. That's a great thing. But web fonts function correctly only if the page can load the font from the server where they are stored. If the connection fails, or if someone is viewing the page offline, the web font won't load and the design will fall apart. To address this, you'll need to create a font stack that will allow an alternative font to load when your perfect font is not available.

**Figure 5.19** Custom font stacks are created in the Manage Fonts dialog.



- A Listing of current font stacks
- B Listing of fonts in the new custom stack
- C Select from available fonts.
- D Add the selected font to the stack.
- E Remove a font from the stack.

- 1 From the application bar, select Modify > Manage Fonts.

The Manage Fonts dialog appears (Figure 5.19). You should see a blue check mark next to the font that's in use in the header—Amatic SC.

- 2 Click the Custom Font Stacks tab. Locate **amatic-sc** in the list of fonts, and add it to the custom stack by clicking the arrow that points to the left.
- 3 In order, add the Gill Sans MT Condensed, Gill Sans MT, and generic sans-serif font to the stack. Compare your settings to those you see in Figure 5.19, then click Done to accept the new font stack.
- 4 Return to the **#header-name** selector that defines the font for the company name, and replace the **amatic-sc** font with this new font stack.

When you view the page in Design view, you'll see the first font in the stack after the web font displayed in the header.

- 5 Adjust the margin value for the `#header-tagline` selector so that the second line of text is positioned correctly.

The remaining work with your font settings involves some trial and error and a critical eye. You can experiment with setting different font styles for the existing selectors, or you can create new selectors to style the text differently for headings and other elements of the design. Remember that it's considered a good design practice to limit the number of fonts on the page and that you'll have to defend your choices when you meet with the client.

## Cleaning Up the Design

Now is the time to be extra critical of the work that you've done. It might even be a good idea to have someone else take a look at your design. Are all the elements of the design neatly aligned? Do the border properties you set previously still look consistent with this design? Do any padding or margin properties need to be adjusted? Does every area of the page present your best work, or do you need to make improvements? The footer, for instance, still has the links displayed as blue underlined text. Shouldn't those `<a>` tags be styled in a way that they're consistent with the rest of the design? You now know where to find these settings in your style sheet and how to create new styling rules, if needed, to clean up the design. And even though Chris the Cartoonist isn't going to be showing up with a check in his hand, you should be thinking like a designer who wants to get that next important contractual milestone met so you can get paid!

## Clean Up Your Code

Whew! Doesn't it feel great to be finished with a job and to be proud of your work? There are just a few more things to do before taking this work to the client, and that involves doing some technical testing to check for errors. We're all human and we all make mistakes, but you have Dreamweaver's tools for catching those errors and fixing them. These operations must be performed while you are in Design view.

- You should tend to the appearance of the code in your page. Dreamweaver has a nifty tool that will clean up the appearance of your code by removing empty lines and setting indents to make the code more readable. The actual contents of your code won't change, but it will appear neater and will be easier to troubleshoot after you run this operation.

From the application bar, choose **Commands > Apply Source Formatting** to have Dreamweaver make your code more readable.

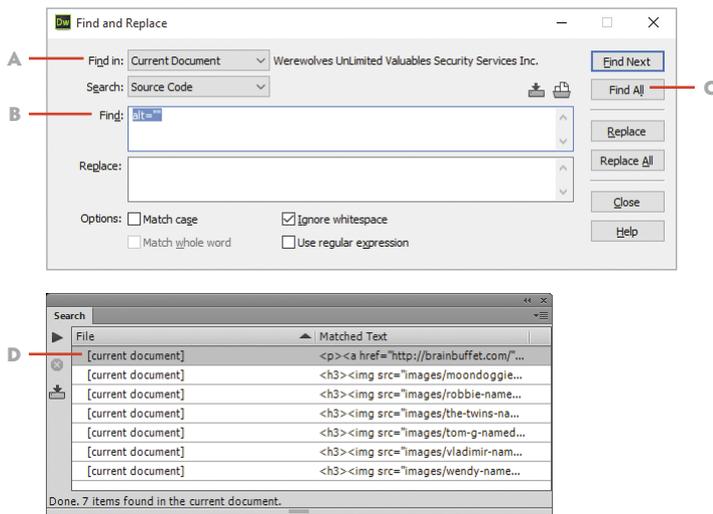
- You'll want to check the spelling in the page. Dreamweaver doesn't give you the squiggly red underlines when you misspell a word like a word processor does, but it does provide a tool to check your spelling. With it, you can replace words, add words to the built-in dictionary, or choose to ignore the spelling of different words. This is a decidedly old-school spell checker, but it does function, and it's always a good idea to check your spelling.

Choose **Commands > Check Spelling** to open the built-in spell-check dialog.

- An even older old-school tool included with Dreamweaver is the ability to run site reports that will check for errors in the HTML. To be honest, this feature hardly functions any longer, but in the event you decide to take the ACA exam for Dreamweaver you should know that these reports can be run.

Choose **Site > Reports** to access the reports.

- The **linting** technology that is built into the latest versions of Dreamweaver works quite well, and you've probably seen evidence of this in action as you work in two different locations. At the lower-right corner of the Document window is an error indicator (  ) that changes from a green check mark (when everything's good) to a yellow warning symbol (when there's a potential issue with your code) to a red exclamation point (when there is a fatal error in your code). You'll also see the line numbers in Code or Split view display in red when there is an error such as an unclosed tag. Keep an eye on both of those areas as you work to resolve potential problems as they occur.
- Dreamweaver has gone backward in its ability to check for accessibility errors. In previous versions, you could check for missing alt text by running accessibility reports, but that feature has been removed, so you'll need to turn to the Find and Replace dialog to find images that lack the alt text required to meet accessibility standards. A simple trick is to locate an image with missing alt text and then run a Find command for the code `alt=""`. When you select Find All, the Results window opens, as you see in **Figure 5.20**. You can click each instance where the alt text is missing to go to that image and add the necessary description of the image.



**Figure 5.20** Use the Find and Replace dialog to find and correct images that are missing the required alt text.

- A** Search in the open document.
- B** Paste in an empty alt text block.
- C** Click Find All.
- D** Click an instance of the missing text to go to the image and add an image description.

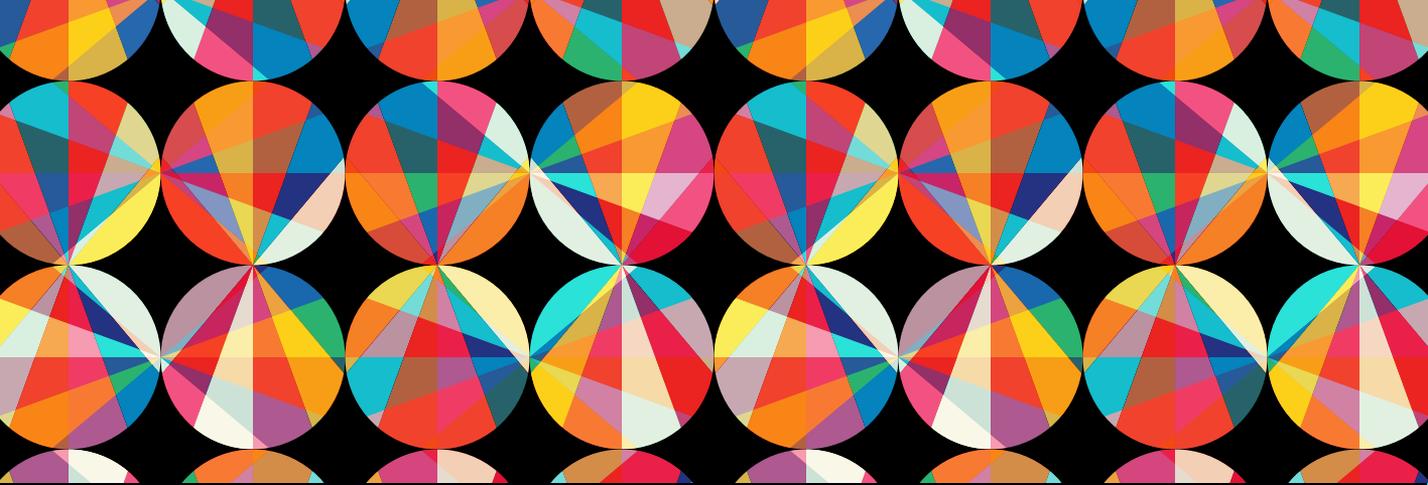
## Preview in Multiple Browsers

When you feel as though all your errors are corrected and your design is finalized and perfect, preview the page in multiple browsers. Use the Preview in Browser button to open the page and check for discrepancies in the appearance of the page and as a final check to be sure you haven't missed anything. No matter what platform you're working in, you should preview the page in Google Chrome and Mozilla Firefox. On Windows computers, you should also check your page in Internet Explorer. If your computer has Windows 10 installed, you can also check your page in the new Microsoft Edge browser. Macintosh users need to check the design in Safari.

## Conclusion

To say you've come a long way in understanding how to design web pages with HTML and CSS and the tools that Dreamweaver provides would be a serious understatement. In this chapter, you've taken all the steps required to get a complete prototype design finished and ready to present to the client for final approval.

With the prototype complete, you're now ready to meet with your client and review your design. Are you ready to defend the design decisions you've made? Does the design represent your very best work? If it does, then you'll be ready to get that final okay from your client and move on to the next step in the website building process: turning the prototype file into Dreamweaver templates that you can use for creating and maintaining all the documents in the site.



## CHAPTER OBJECTIVES

### Chapter Learning Objectives

- Prepare web documents for a production environment.
- Use Dreamweaver to modify, edit, and sort Cascading Style Sheets (CSS).
- Create an external CSS file and link the file to the pages in a site.
- Create Dreamweaver templates and apply templates to existing web pages.
- Use Dreamweaver templates and library items to maintain and update a website.
- Publish a website to a live web server and update the pages on the site when changes are made to documents.

### Chapter ACA Objectives

For full descriptions of objectives, see the table on pages 240–246.

**DOMAIN 1.0**  
SETTING PROJECT REQUIREMENTS  
1.4

**DOMAIN 2.0**  
UNDERSTANDING WEB AUTHORING  
2.2, 2.4

**DOMAIN 3.0**  
UNDERSTANDING ADOBE DREAMWEAVER CC  
3.1, 3.2

**DOMAIN 4.0**  
WEB AUTHORING USING ADOBE  
DREAMWEAVER CC  
4.1, 4.4, 4.5

**DOMAIN 5.0**  
TESTING, PUBLISHING, AND MAINTAINING WEB  
CONTENT USING ADOBE DREAMWEAVER CC  
5.1, 5.2, 5.3

## CHAPTER 6

# Moving from Prototype to Production

As you move from the prototype to the production phase, there are a number of things that need to be done. The site must be ready to have the content moved from the prototype page to all the individual pages in the site. But here's the thing about websites and web design in general: Change is constant. Your client has already asked how easy it will be to update the site in the future, and making changes is an inevitable part of maintaining a website. Luckily for you, you're using Dreamweaver, so you can use the tools that the software provides to publish all the files that make up your client's site to a web server *and* easily revise the documents in the site when your client wants something changed.

In this chapter, you'll learn how to prepare the final master style sheet document that will be used in all the pages of the site. Then you'll prepare Dreamweaver templates and library items to create and update the pages and see how to publish and update all the assets in the site to a web server. By the time you finish this chapter, you'll be ready to go live with an entire website. How cool is that?



**Figure 6.1** Chris the Cartoonist

## Let's Go Live!

Chris the Cartoonist (**Figure 6.1**) certainly is pleased with the final prototype you finished in the last chapter, and he's given you the go-ahead to move on to the production phase of the *wulvs.com* website. This is an exciting time!

**Chris Sez:**

*Wow! You've really nailed the look I was going for. I can't wait to see all of this come together.*

*OK, so you can consider this my formal approval of the design you created. Now what? I guess you have to make a bunch of copies of the web pages and hook them all together, right? Will I be able to make changes after the pages are all built? I might add some new characters or want to put some new links to some of my other cartoonist friends' websites. How hard will that be to do in the future?*

*Oh yeah, and don't forget that I might have some new advertisers lined up real soon. Can I put different ads on different pages? Will it be easy to change the ads in the future?*

*I'm really excited to see all the pages filled up with all the content you worked out. Thanks again for doing such an awesome job!*

*Chris*

## Get Your Files Here

As you did in previous chapters, copy the folder that contains this chapter's files to your computer.

- 1 Download the project files for this lesson, named `chapter-06.zip`, from the Lesson & Update Files tab on your Account page at [www.peachpit.com](http://www.peachpit.com).
- 2 Unzip the files and copy the `chapter-06` folder to the `Learn-Dreamweaver` folder you created in Chapter 1.

# Define the Chapter 6 Website

With the files you've been provided for this chapter in the project folder, you're ready to set up the site in Dreamweaver.

★ ACA Objective 4.1

- 1 From the application bar, choose Site > New Site.
- 2 Name the site **Chapter 6**.
- 3 Browse to the **chapter-06** folder in your **Learn-Dreamweaver** project folder.
- 4 Click Save to complete the site definition.

**NOTE** Refer to Video 1.5, "Defining a Dreamweaver Website," if you want a quick demo of how the site definition process works.

## Reviewing CSS Syntax

You've spent a lot of time in the CSS Designer panel, but it's important that you have a thorough understanding of the kinds of rules that are in the style sheet and the different ways they are used. The syntax that selectors are written in is important to the browser and to your future work as a web designer, and there are likely to be questions about it if you decide to sit for the Adobe Certified Associates exam for Dreamweaver. Let's review.

★ ACA Objective 2.4

## Types of Style Sheets

You can write CSS styling rules in three ways:

- **Internal style sheets** are contained in the head of a document and are enclosed by the `<style>` `</style>` tags.
- **External style sheets** are separate files that have the `.css` file extension. These style sheets are linked to an HTML document by using a `<link>` tag in the document.
- **Inline styles** are written directly within HTML tags and should be used sparingly, if at all, as in this example: `<h1 style="font-size: 150%">`.

## Selector Types

Let's take a look at the different types of selectors, most of which you have already used in the `prototype.html` document.

- **Element selectors** define the properties of an entire HTML tag. An example of this kind of selector is the `<body>` selector in this document, where the margins are set to zero for the entire page.
- **ID selectors** are used to assign styling properties to tags by their ID attribute. IDs must be unique within an HTML document. You see an ID selector in the styling rules for this document for the `#container` selector. ID selectors begin with the `#` symbol.
- **Class selectors** are applied as needed and are targeted at tags that include a class attribute. The `.float-right` selector in this document is a class selector. Class selector names are preceded by a dot (period) symbol.
- **Descendant selectors** combine more than one selector with a space between the individual selectors. The browser reads the order of the selectors and applies the styling rules to the last item in the list. The `#sidebar p img` selector is an example of a descendant selector.
- **Group selectors** are similar to descendant selectors but affect *any* element found after a parent element, not just the last selector. Group selectors are separated by a comma, as in `body, td, th`.
- **Pseudo-class selectors** evaluate a condition that exists and apply the style when a condition is met. The pseudo-classes you used in this design create the rollover effect and use the `link`, `visited`, `hover`, and `active` states to determine how the rule is applied as in the initial state you set for the navigation buttons using the selector `.sitenav ul li a:link, a:visited`.
- **Child selectors** are used where a selector is immediately followed by a specific rule that gets applied to the “parent” element that precedes it. You saw an example of a child selector when you created the styling rule that set a background color for alternate rows using the `#content table tr:nth-child(odd)` styling rule.

## PROJECT 6.1

# Putting the Style Sheet into Production

★ ACA Objective 2.4

★ ACA Objective 4.4

★ ACA Objective 4.5

The first task in taking the site to production will be the style sheet that defines the site's appearance. First you should make a backup copy of the file the client has approved. Then you'll take all that beautiful code you created in your earlier exercises, do some cleanup and organizing, and move the styling instructions written into the CSS into a separate external file that will be the master styling document for all your pages.

 **Video 6.1** *Putting the Style Sheet into Production*

## Setting Shorthand Notations in CSS

Dreamweaver does a great job of writing CSS rules when you use the CSS Designer panel, but sometimes the code isn't as efficient as it might be. In this exercise you'll optimize your styling rules by replacing some of the lengthy styling rules with more compact shorthand versions.

- 1 Select `prototype.html` in the Files panel, then right-click and select Edit > Duplicate. Dreamweaver will automatically create a file named `prototype - Copy.html`. Change the filename to **`prototype-client-approved.html`**.

This file serves as a backup in the event there is a problem with the production phase you're about to enter. Do not open this file unless you need to refer back to the original.

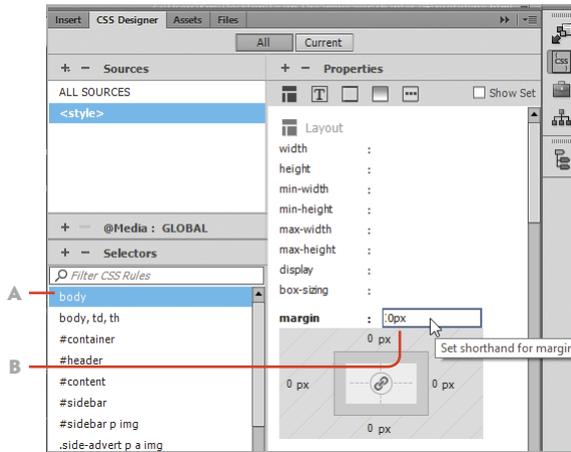
- 2 Open `prototype.html` from the Chapter 6 site files.

As you review this page, you'll see that it's fundamentally the same file that you finished at the end of the last chapter. A few new styling rules have been added to get the final design elements in place, and the placement of the content on the page has been tightened up. This was the work you were tasked with at the end of Chapter 5.

- 3 Switch to Code view, and scroll to the top of the page. As you review all the styling rules you've created, be on the lookout for any problems, errors, empty styling rules, or other issues that need to be adjusted.

Two issues immediately come into view. First, the styling rules include some properties that could be combined into a shorthand version to reduce the amount of the code in the page. Second, the styling rules are not listed in a logical order. You can deal with both problems in the CSS Designer panel.

- 4 Select the `body` selector in the CSS Designer panel. This rule can be trimmed down with a shorthand notation.
- 5 Locate the Set Shorthand field directly above the margin properties for this selector, and type `0px` (Figure 6.2).



**Figure 6.2** You can set CSS shorthand notations in the CSS Designer panel.

- A** Select the `body` selector.
- B** Set the margin shorthand to `0px`.

- 6 View the code for the `body` selector; the four lines that were being used to define the margins have been reduced to a single notation: `margin: 0px;`
- 7 Set shorthand padding values for the `.float-left` and `.float-right` styling rules.

When working on your own design, examine your code carefully to find any opportunities to reduce the amount of code in your style sheet.

- 8 Save your work.

## Arranging the Source Order of the Style Sheet

In these next steps, you put your styling rules into a logical order that makes it easier for you to find the styling rules you want to work on in the future. The CSS Designer panel makes this easy to do since it allows you to drag and drop selectors that are listed in the “stack” of rules in your style sheet.

- 1 Examine the listing of the CSS selectors in `prototype.html`.

The document has higher-level rules that were applied using ID selectors, such as `#content`, and a number of styling rules that are descendant selectors of the upper-level rule, such as `#content p`. Your goal is to group these selectors in order, with the highest-level selector at the top of the grouping and with longer, more specific selectors underneath.

## LEVEL UP: SMART RULE CREATION

When you create a new styling rule, Dreamweaver writes the code for the new rule directly after whatever rule is selected in the CSS Designer panel. When you create new styling rules, it's a good practice to select carefully so that the selectors are grouped together.

- 2 In the CSS Designer panel, drag the `#container #content h1` selector up until it falls below the `#container` rule.
- 3 Select the list of rules that begin with the `#content` ID, as you see in **Figure 6.3**. (Shift-click to select a contiguous group.) Drag this group up until the rules fall below the `#content` selector.

As you examine the source order of the style sheet, be sure that you maintain the proper order for any pseudo-classes that are in the document. Remember that these rules must always follow the correct order of link, visited, hover, and active, or LVHA.

- 4 Order the selectors in the style sheet until the selectors are grouped from the shorter, more general rules to longer, more specific rules.

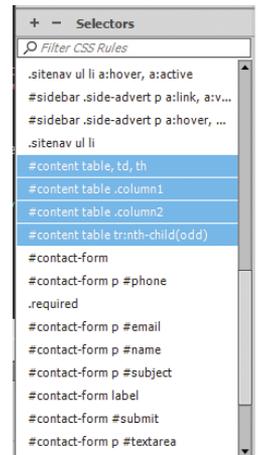
**Figure 6.4** shows a portion of the CSS Designer panel where the `#sidebar`, `.sitenav`, `#content table`, and `#contact-form` selectors have been grouped with the more specific rules below them.

- 5 Switch to Live view to ensure that none of your reordering work has had an effect on the appearance of the page. Complete the work of ordering and checking the design on your own.
- 6 Save your work.

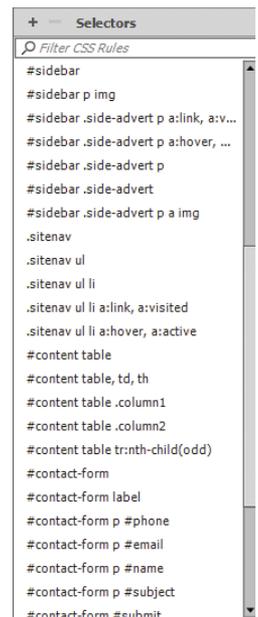
## Commenting the Style Sheet

You write comments into computer code, as a note to your future self, providing a way for you to recall the why and how of the selectors you designed. Comments are also a way that you as a designer can make notations for others to read when you are working in a collaborative environment.

- 1 Switch to Code view.
- 2 Enable the Coding toolbar, if it isn't visible to the left of your code, by choosing View > Toolbars > Coding.



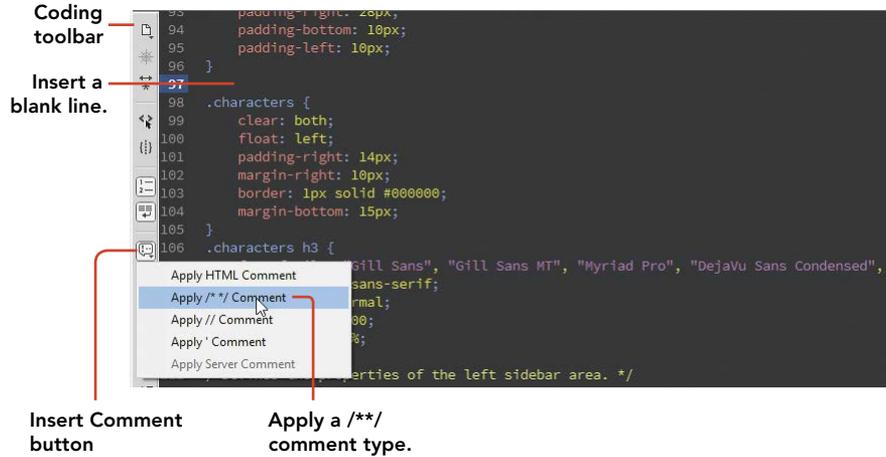
**Figure 6.3** You can choose multiple selectors in the CSS Designer panel by holding down Shift while you select the rules.



**Figure 6.4** Grouping CSS selectors from general to more specific rules

- 3 Place your cursor at the beginning of the `.characters` selector, and press Enter/Return to create a blank line. Press the Up Arrow key on your keyboard to move the cursor to the blank line (Figure 6.5).

**Figure 6.5** Comments allow you to make notes that you and your co-workers can use as a reference in the future.



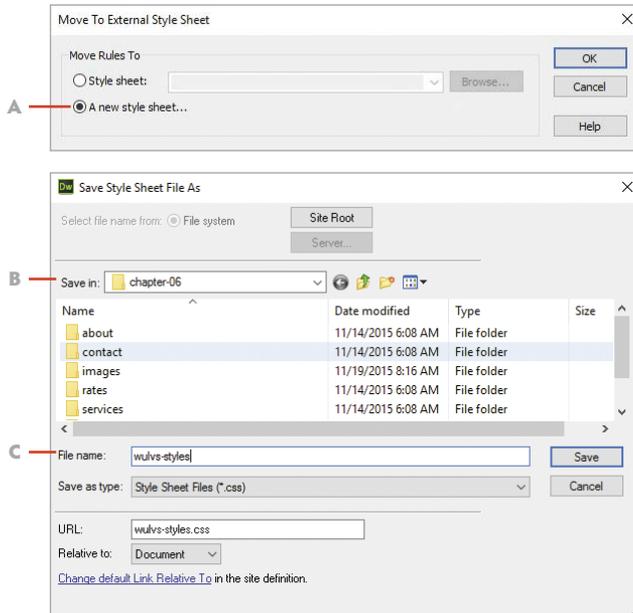
- 4 Click the Apply Comment button, and choose the `/**/` comment type. Comments that are in between the two asterisks are visible in the code but are not read by the browser.
- 5 Type the following comment for this block of CSS rules: **`/*These rules define the character boxes*/`**.
- 6 Work through the style sheet, adding comments above each major grouping of selectors.
- 7 Save your work.

## Converting the Style Sheet to an External CSS File

The final step to prepare for production is converting all the styling rules in the `<head>` of the document into a separate CSS file that will be linked to the HTML page. This is a surprisingly quick and easy operation to perform.

- 1 In Code view, locate the first rule that follows the opening `<style>` tag, and place your cursor in front of the `body` selector.
- 2 Highlight all the selectors in the document, being sure not to include either the opening or closing `<style>` tags.

- 3 Choose Format > CSS Styles> Move CSS Rules.
- 4 In the Move to External Style Sheet dialog, select the “A new style sheet” option (Figure 6.6), and click OK. The Save Style Sheet File As dialog opens.
- 5 Be sure the file is being saved into the `chapter-06` root folder for the site, and name the file `wulvs-styles`. Click Save when your settings match those in Figure 6.6.



**Figure 6.6** The process of moving selected styling rules to an external CSS file

**A** Select “A new style sheet.”

**B** Set the location to the root folder for the Chapter 6 site.

**C** Set the filename to `wulvs-styles`.

When you save the file, all the selectors are removed from the document and are replaced with a link to the external file. Take note of this line in your code:

```
<link href="wulvs-styles.css" rel="stylesheet"
→ type="text/css">
```

This code directs the browser to retrieve the external CSS file and use the styling rules in that document for this page.

- 6 Locate the empty opening and closing `<style>` tags, and delete these two lines of code:

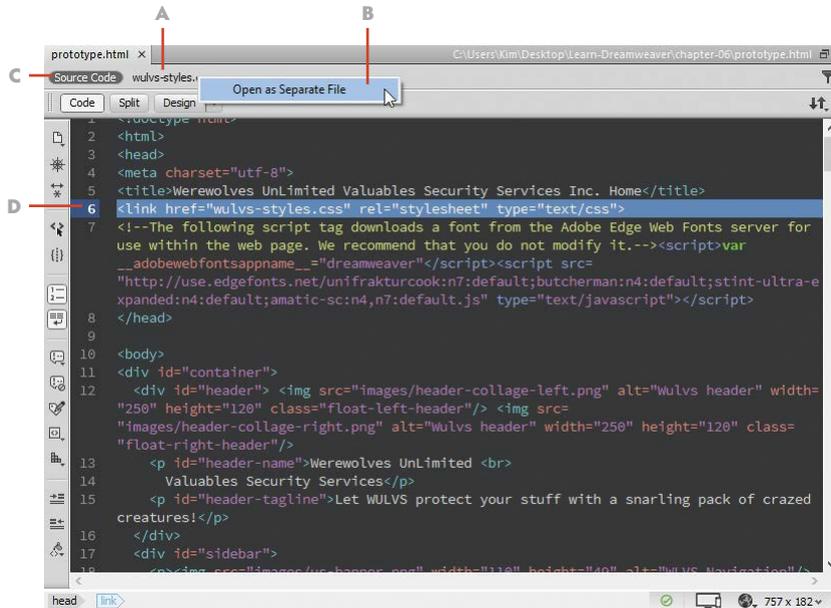
```
<style type="text/css"
</style>
```

Dreamweaver now displays the name of the CSS file in the Related Files toolbar below the filename, as you see in **Figure 6.7**. You can now switch between the code for the page by selecting the Source Code button, or you can work on the code in the style sheet by selecting the name of the CSS file. You can also open the CSS file in a separate window by right-clicking the name of the file and selecting Open as a Separate File.

- 7 Save both the HTML file and the CSS file by choosing File > Save All.

**Figure 6.7** Dreamweaver displays a link to the external CSS file in the Related Files toolbar.

- A** Related filename indicator
- B** Right-click to open the external file in its own window.
- C** Click the Source Code indicator to work on the HTML file.
- D** Link to the external CSS file.



You have moved the styles in your prototype file into an external CSS file that can be linked to all the files in the site. This produces a much more efficient way of modifying a styling rule, since you need to make a change only to this single CSS file to make a styling change in all the documents that are linked to the external file.

## PROJECT 6.2

# Building Out the Site with Dreamweaver Templates

A Dreamweaver template is a special type of web page used within the Dreamweaver work environment. It locks parts of the page structure in place while allowing the designer to edit other parts of the page that need to change. Templates are how Dreamweaver users give all the pages in a site a consistent design. And the beauty of working with templates is that when one of those inevitable changes takes place, you can update the areas that are shared among all the pages in the site without having to open each file or worry about having the content in the page be modified when the template is revised.

You're going to get a lot done in this project. To begin, you'll convert the prototype page to a Dreamweaver template file and then insert some special Dreamweaver markup that will allow you to place the content that goes into each page. At the end of this project, you'll have moved from a single prototype web page to an entire functional website with all the pages designed just the way you want.

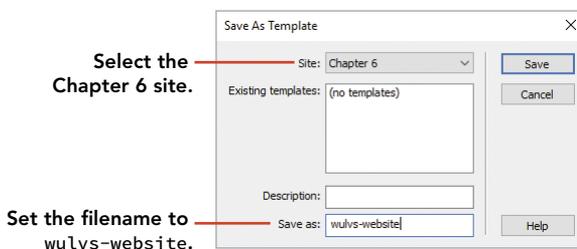
- ★ ACA Objective 2.2
- ★ ACA Objective 3.1
- ★ ACA Objective 4.4
- ★ ACA Objective 4.5

▶ **Video 6.2** Building Out the Site with Dreamweaver Templates

## Create the Template File

When you create a template file, you set up a special kind of web page that includes coding specific to Dreamweaver. This markup allows you to lock some areas of the template while allowing changes to other, editable areas of the page.

- 1 With `prototype.html` open, choose File > Save as Template. The Save as Template dialog opens (**Figure 6.8**).
- 2 In the Save As field, name the file **wulvs-website** and click Save.



**Figure 6.8** When you convert a file to a template, you must designate the site and the filename for the template file.

- 3 Click Yes when Dreamweaver prompts you to update the links in the document.

Several significant changes occurred. Take note of the new Templates folder inside the Files panel. Dreamweaver creates this folder automatically. This is the location where all template files are located.

Inside the Templates folder is a file named `wu1vs-website.dwt`. The `.dwt` file extension designates this file as a Dreamweaver template.

## Insert an Editable Region

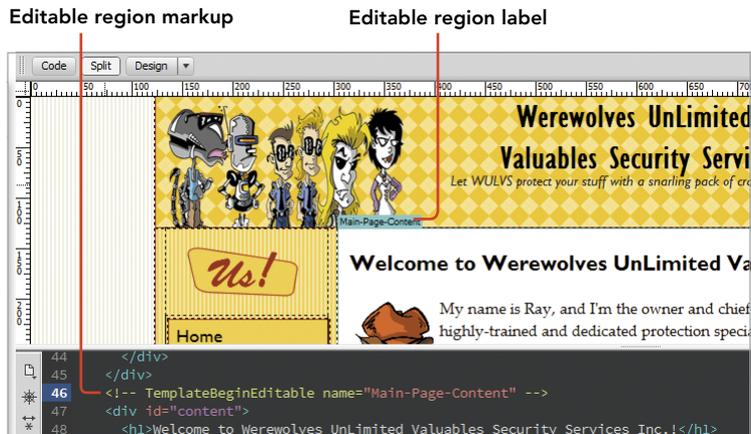
When you work on a page that has been constructed from a Dreamweaver template, some areas of the page can be changed and others cannot. The areas that are locked by the template are those that should appear the same on every page. The editable regions—or those areas where changes are allowed—are created in the Dreamweaver template file.

The area of the template that will be editable is contained within the `<div>` with the ID of `content`.

- 1 With the `wu1vs-website.dwt` file open, click anywhere within the main content area of the page and select the `<div id="content">` in the Tag selector.
- 2 Choose Insert > Template > Editable Region.  
The New Editable Region dialog appears.
- 3 Name the editable region **Main-Page-Content**. Click OK.

Dreamweaver indicates the editable region with a blue label (Figure 6.9) and places special markup in the code.

**Figure 6.9** Dreamweaver displays a label that indicates the editable region.



Now you'll remove all the existing content in this editable region and replace it with placeholder text. This operation is best done in Code view, where you can easily select all the contents in the `<div id="content">` section.

- 4 Switch to Code view, locate the opening tag for the content section—`<div id="content">`—and click the arrow next to the line number to collapse the entire tag. This makes locating the closing tag easier. Once you have identified where your CSS rules begin and end, click the arrow again to expand this area of your code.
- 5 Highlight all the tags and content inside the `<div id="content">`, and press Delete to remove the code and content from the main area of the page.
- 6 Insert your cursor inside the empty `<div>` tag, and type **Content goes here** as the placeholder text. Your document should look similar to **Figure 6.10**, with the empty `<div>` tag on the page.



```
44 </div>
45 </div>
46 <!-- TemplateBeginEditable name="Main-Page-Content" -->
47 <div id="content">
48     Content goes here
49 </div>
50 <!-- TemplateEndEditable -->
51 <div id="footer">
52     <ul>
53         <li> <a href="../index.html">Home </a>|</li>
```

**Figure 6.10** The result of replacing all the code in the `<div id="content">` with placeholder text

- 7 Save and close your template page. You're now ready to build pages from this template.

## Replace the Home Page

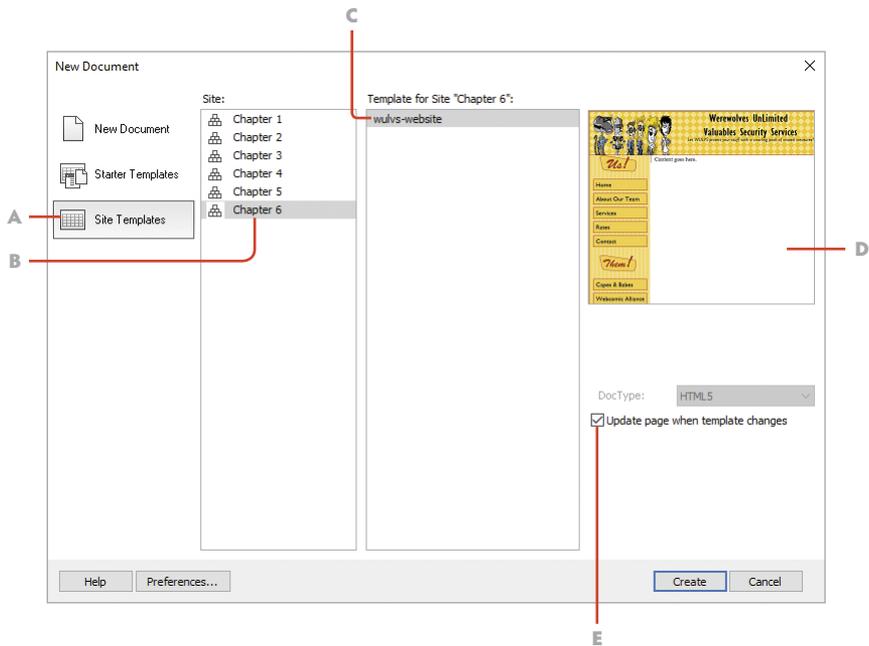
Now that you have a template to work from, you can start moving content from the prototype page to the individual documents in the site. The home page is a great place to start.

- 1 Choose File > New to create a new file. You should be in Design or Split Design and Code view.
- 2 From the New Document dialog, select the Site Templates category on the left and Chapter 6 in the list of defined sites in the center of the window (**Figure 6.11**).

A thumbnail version of the template you just created displays on the right.

**Figure 6.11** The New Document dialog allows you to create a new page based on a template file.

- A** Select Site Templates category.
- B** Select the Chapter 6 website.
- C** Select the template filename.
- D** Template thumbnail
- E** Ensure that this option is selected.

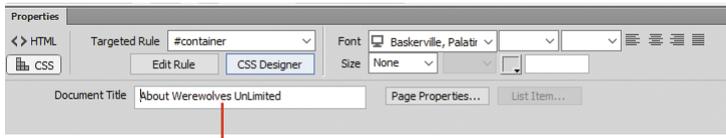


- 3 Select the `wulvs-website` template.
  - 4 Verify that the “Update page when template changes” checkbox is selected, and click Create.
  - 5 Choose File > Save As, browse to the root folder for the site, and select the `index.html` file. Click Save.
  - 6 Choose Yes/Replace when Dreamweaver asks if you want to replace the existing file.
  - 7 Open `prototype.html`, and copy all the text and images from the top of the page through the bulleted item list just above the Team Page text. Press CTRL+C/Command+C to copy the selected text and images.
  - 8 Remove the placeholder text in the `index.html` page, and press CTRL+V/Command+V to paste the copied text and images into the home page. You can close this file after it has been saved.
  - 9 View the contents of the new home page, and save your work.
- Congratulations! The home page for your website is finished!

## Build the Team Page

The next page in your prototype file is the area marked Team Page. You'll create another new document in these next few steps and replace the existing (empty) page in the site with a new page that uses your template.

- 1 Choose File > New to create a new file. As you did for the home page, select the Site Templates category, the Chapter 6 site, and the `wulvs-website` template. Click the Create button.
- 2 Choose File > Save As, and browse to the `about` folder. Select `index.html` as the new filename. Click Save and then Yes/Replace to overwrite the existing file. Template files do not overwrite the page title when a new page is created. To change the page title, you edit the title in the Property inspector.
- 3 Change the Document Title field to **About Werewolves UnLimited** (Figure 6.12).



Change the document title to About Werewolves UnLimited.

**Figure 6.12** Page titles may be changed in the Property inspector.

Now, you need to remove the placeholder text and replace it with the area of the prototype page that includes all the different characters and their descriptions.

- 4 In `prototype.html`, select the area of the page under the Team Page heading all the way down to the last character box. You might find this easier to do in Code view. Copy the selection.
- 5 Return to the `index.html` page, and remove the placeholder text. Paste the copied text and images into the editable region on the page, staying within the `<div id="content">`.

You'll immediately see that the images in the page are not displaying correctly because the path to the image files is no longer correct. If you look at the source for the image in the Property inspector, you'll see that the path to the first image is listed as `images/ray-named.png`. This needs to be corrected to point to the correct location, which is one folder level up in the folder structure for the site. For example, the correct path to the first image needs to be `../images/ray-named.png`.

- 6 Correct the path using one of the following methods:
  - Use Find and Replace to change the source code for the page so that all instances of `images` are replaced with `../images`.
  - Browse to the correct file from the Property inspector using the Browse for File icon.
  - Type the correct path to the `images` folder in the Property inspector for each image or directly in the code.
- 7 Save the file.

Your second page is complete!

### LEVEL UP: COMPARING CODE

Here's a trick you can use when you run into a situation where you need to determine the correct path to a number of images. In the Property inspector, for the file that isn't showing correctly, browse to the file so that the path is corrected, and then compare the correct code to the code that isn't working. You can then see what changes need to be made to get everything working correctly, and you can more quickly set about making the revisions.

## Create the Additional Pages for the Site

Now that you know how to use a template file to create new pages and replace an existing page, you can set about the task of replacing all the remaining pages in the site. Using the techniques from the previous steps in this project, your task is to replace the three additional remaining index pages in the site. Here are the general steps:

- 1 Create new pages from the template file, and replace `contact/index.html`, `services/index.html`, and `rates/index.html` with the new pages that you created from your template file.
- 2 Copy and paste the content from `prototype.html` into the correct pages in the site. Remember that you'll also need to supply a new page title for each page. Each page should also include the heading that leads the section in the prototype file.
- 3 Replace the file that viewers are directed to if they complete the contact form. That page is named `thankyou.html`. On that page, insert the `ray.png` image and a short message that thanks the person for their interest and reminds them that this isn't a *real* company.

## PROJECT 6.3

# Adding Optional Content with Library Items

It's pretty cool to see the website go from just a single prototype page to an entirely functional site, isn't it? You now have all the contents of the prototype copied into the individual pages within the site, and you can click through your navigation to see how the site has come together. Great job!

There's one last thing on the client's list of requirements. The advertising block that sits in the left sidebar needs to have two qualities. First, you need to have the ability to show the advertising area on some pages but hide it on others. Second, the client has indicated that there will be some additional advertisers coming onboard, so the actual ads that show on each page also need to be changed from time to time.

This kind of work—where you show or hide and change one block of content for another—can best be handled with a new template object combined with another tool in your Dreamweaver toolset: the library item.

## Add an Editable Optional Region

An editable optional region in Dreamweaver is created by inserting special markup that allows a certain area of the page to show when its visibility value is set to true, and to be hidden when visibility is set to false.

- 1 Open the `wulvs-website.dwt` file from the `Templates` folder.
- 2 Click anywhere within the advertisement area in the sidebar, and select the `div .side-advert` indicator in the Tag selector.
- 3 Open the Insert panel, and change the category in the drop-down menu at the top of the panel to `Templates` (Figure 6.13).
- 4 Click `Editable Optional Region`.
- 5 In the `New Optional Region` dialog, name the region `advertising1` and click `OK`. Leave `Show by Default` selected so that this region will continue to be present in this document. You'll hide or show the region in the following steps.

★ ACA Objective 1.4

★ ACA Objective 2.4

★ ACA Objective 3.2

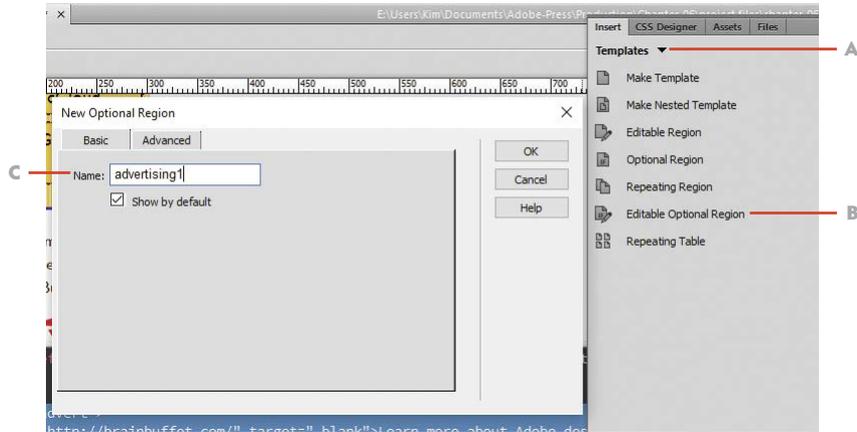
★ ACA Objective 4.4

★ ACA Objective 4.5

 **Video 6.3** Adding Optional Content with Library Items

**Figure 6.13** Use the Insert panel to insert a new editable optional region.

- A** Switch the Insert panel to the Templates category.
- B** Click to insert an editable optional region.
- C** Name the region in the New Optional Region dialog.



- 6 Save the file, and you will be prompted to update all the pages that use this template. Click Update to modify all the files in the site where this template is in use. Click the Close button when the Update Pages window indicates that the updates are done.
- 7 Open `contact/index.html`. In the head of the document, locate the following line of code that Dreamweaver has inserted to allow you to show or hide the optional region:

```
<!-- InstanceParam name="advertising1" type="boolean"
→ value="true" -->
```
- 8 Change `value="true"` to **`value="false"`**.  
When you view the page, you'll see that the advertisement still shows.
- 9 Choose `Modify > Templates > Update Current Page` to complete the process of hiding the advertising block.
- 10 Save this file, and close the document.

You now know how to show or hide an optional region by setting the value in the Dreamweaver markup to either `true` or `false`.

## Convert Content to a Dreamweaver Library Item

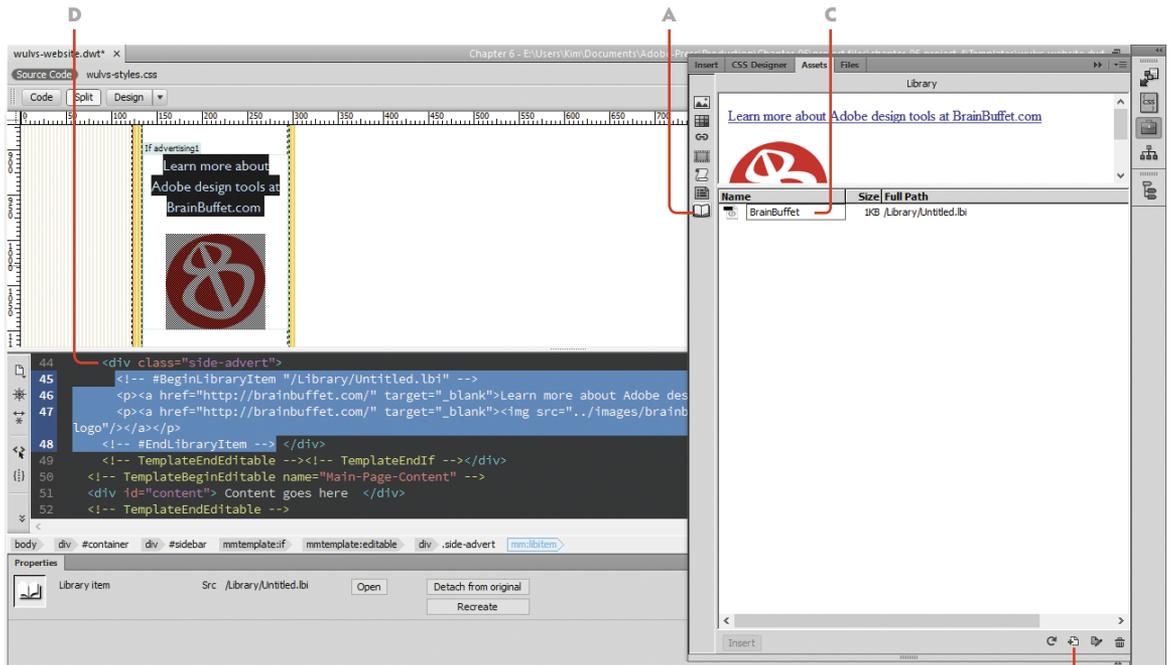
Dreamweaver library items are self-contained chunks of HTML that can be inserted into a document when needed and updated when they need to be changed. The Assets panel is used to add, edit, and manage the Library Items within a site. In the

next steps, you'll create a new library item from the existing advertisement that's in the sidebar.

- 1 Open `wulvs-website.dwt` from the `Templates` folder.
- 2 In Code or Split view, highlight the code block that is inside the advertising block in the sidebar. Be sure that you select only the *contents* within the `<div>` tag, and not the `<div>` itself.
- 3 Open the Assets panel, and switch to the Library Item category (**Figure 6.14**). Click the New Library Item button at the bottom of the panel.

Dreamweaver will provide a warning that styling information does not get included with the library item. Not to worry. Your CSS styling rules will still apply based on the named elements that are within the document.

- 4 Enter **BrainBuffet** for the new library item name (Figure 6.14).



**Figure 6.14** Use the Insert panel to insert a new library item.

- A Select the Library Item category in the Assets panel.
- B Click the New Library Item button.
- C Name the new library item.
- D The library item is inserted inside the existing `<div class="side-advert">` tag.

- 5 Press Enter/Return to accept the name. Click Update when Dreamweaver prompts you to update the files in the site.
- 6 Save the template file. Click Update when prompted to update the pages that use the template file. Click Close when the updates are complete.
- 7 Open the home page for the site, `index.html`.  
Dreamweaver will not automatically replace the code within the advertising block, so the next step in replacing the ad with the library item is to delete the existing content within the `<div>`.
- 8 In Code or Split view, select the contents that are inside the `.side-advert <div>` and delete it.
- 9 With your cursor inside the `.side-advert <div>`, open the Assets panel, select the BrainBuffet library item, and click the Insert button at the bottom of the panel.  
The content of the `<div>` now displays the advertisement that belongs on this page.
- 10 Save your file.

## Create a Library Item from Scratch

So what's the big deal anyway? You already had this content in the advertising block, and all you've done so far is replace what was already there with the same content.

The real value of library items is in your ability to replace content like this easily using the Assets panel. When your client has more advertising content to display, all you'll need to do is replace one library item with another to make the switch. And if you need to edit the contents of a library item, Dreamweaver will allow you to automatically apply the changes everywhere within the site where that item is used.

To see this in action, you'll create a new library item.

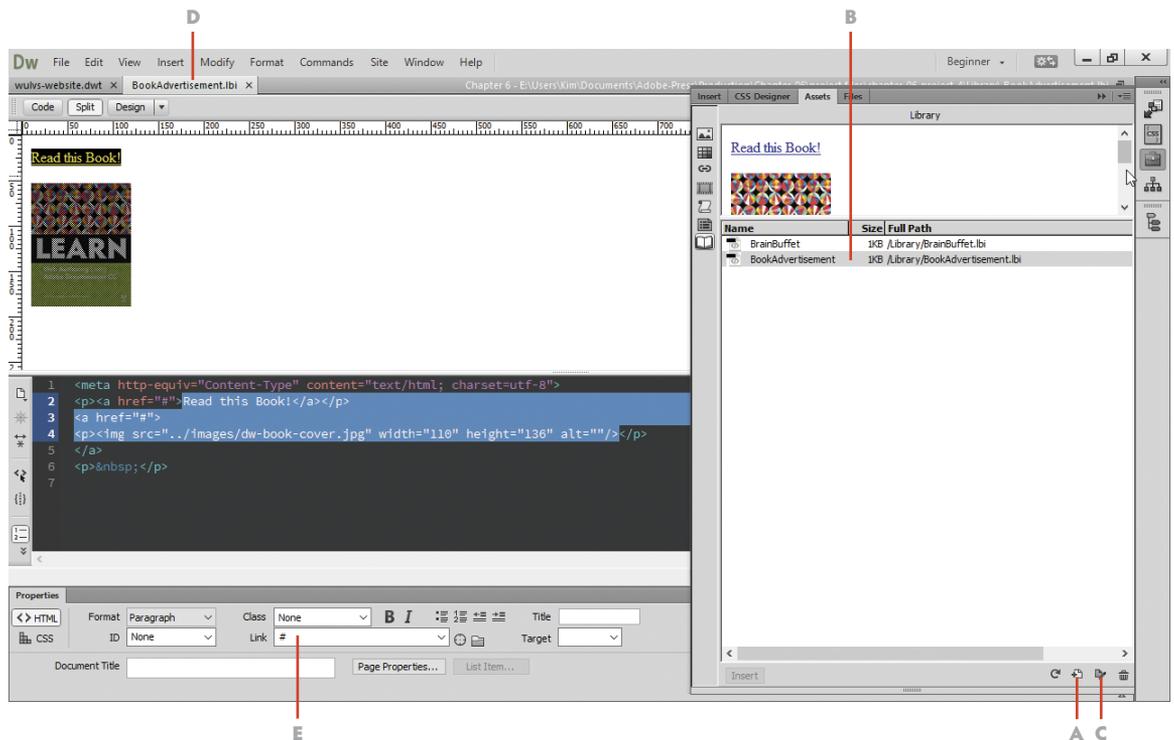
- 1 Open the home page for the site, `index.html`. Make sure that nothing on the page is selected.
- 2 Open the Assets panel, and click the New Library Item button at the bottom of the panel (**Figure 6.15**).
- 3 Name the library item **BookAdvertisement**.

- 4 Click the Edit button at the bottom of the Asset panel. Dreamweaver will open a new document window for the file named `BookAdvertisement.lbi`. The `.lbi` file type is unique to Dreamweaver and is used for all library items.
- 5 In the `BookAdvertisement.lbi` document window, type **Read this book!** and press Enter/Return.
- 6 Insert the `dw-book-cover.png` image for this book from the `images` folder. Remember to add the alt text.
- 7 Select the text and the image, and in the Property inspector's Link field, create a "dummy" link by typing the `#` symbol. Press Enter/Return to accept the link. Verify your settings match those in Figure 6.15.
- 8 Save and close the library item.

Your library item is complete!

**Figure 6.15** ▼ Use the Assets panel to create a new library item and edit its contents.

- A** Click to create a new library item.
- B** Name the new library item.
- C** Click to open the library item in a document window.
- D** The Document window for the open LBI file
- E** Links are created and displayed in the Property inspector.



## Insert and Update Library Items

With your new library item created, you can open individual pages within the site, swap out the content in the advertising block of a page, and add the new library item. When you change the library item by editing and saving it, Dreamweaver will automatically update the pages where the LBI file is used.

- 1 Open the `rates/index.html` page.
- 2 Delete the contents of the advertising block in the sidebar.
- 3 Open the Assets panel. Select the `BookAdvertisement.lbi` item, and click the Insert button. Save the index page.
- 4 From the Assets panel, select the `BookAdvertisement.lbi` item, and click Edit.

The library item opens in a new Document window.

Now you need to change the link in the Property inspector to the actual link for this book.

- 5 Search *Peachpit.com* to locate *Learn Adobe Dreamweaver CC for Web Authoring*, and copy the URL. Paste the URL into the Link field.
- 6 Save the library item.
- 7 Click Update when Dreamweaver prompts you to update the pages that use the LBI file.

Dreamweaver automatically changes the link on all the pages that use this library item.

- 8 Click Done when the update is complete.

Your work is done!

In this project, you've seen how to create an editable region within a template that can be set to either show or hide a region of the page. By combining that option with a Dreamweaver library item, you now know how to quickly change content within a page and update all the pages that use these Dreamweaver content management tools.

## PROJECT 6.4

# Publishing and Updating a Dreamweaver Site

★ ACA Objective 5.1

★ ACA Objective 5.2

★ ACA Objective 5.3

So far in this chapter, you've accomplished a tremendous amount of work, and now it's time to take the next big step and publish your completed work to a live website.

Well, you *could* publish your files to a web server if you owned the wulvs.com domain name and hired a web hosting company to provide the services needed for your site to connect to the rest of the World Wide Web. In a real production environment, the web hosting company would provide instructions for how to set up Dreamweaver to connect to the IP address where the live website would be stored. Thousands of companies provide web hosting services and maintain the web servers that make live websites possible.

Of course it's not really practical for that to be done in this case, so instead of a live server, you'll create a folder on your computer that you'll use to simulate the process of copying files you've completed on your computer to another location. Fundamentally, this is how web designers interact with the servers where their live files are stored, so this will give you a good idea of how the process of publishing and updating from a local drive to a live server takes place.

 **Video 6.3**  
Publishing and  
Updating a  
Dreamweaver Site

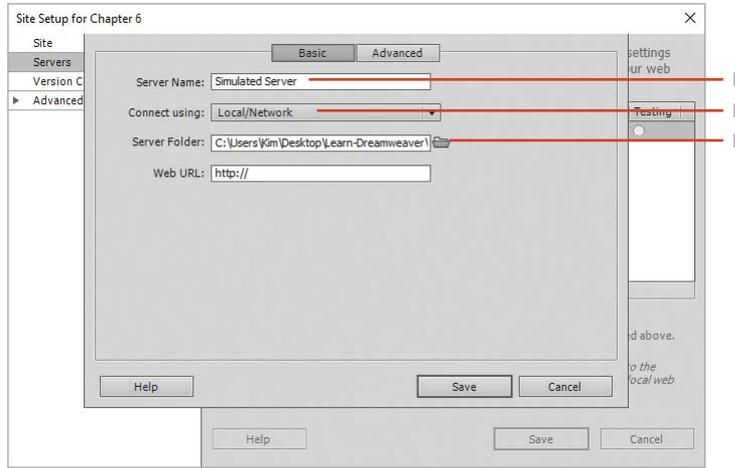
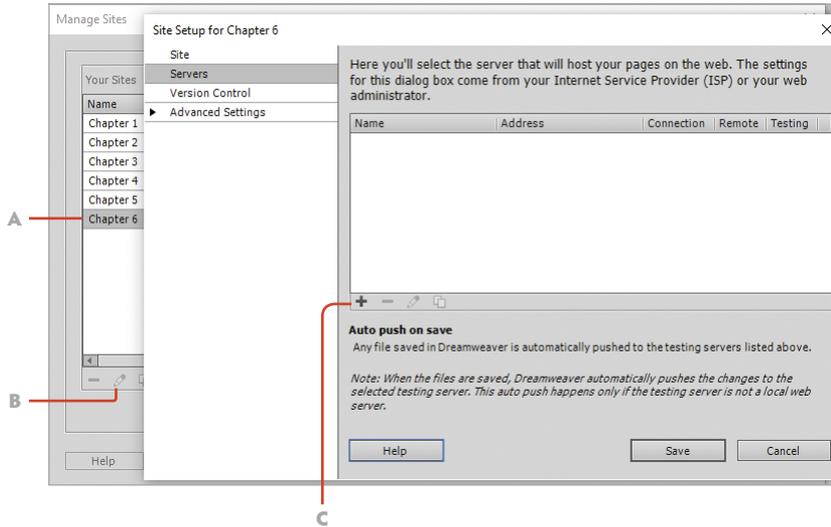
## Defining the Remote Server

For this project, you'll create a folder on your computer that you can use to simulate the process of publishing to a live web server. Publishing files essentially involves placing a copy of the files on your computer into the folder that a web hosting company has prepared for your site. The following operation mimics that process.

- 1 Using Windows Explorer (Windows) or the Finder (Mac), create a new folder inside the **Learn-Dreamweaver** folder where all your project sites have been stored. Name this new folder **wulvs-website**.
- 2 In Dreamweaver, choose Site > Manage Sites. Refer to **Figure 6.16** on the next page.
- 3 Select the Chapter 6 site, and click the pencil icon in the Manage Sites dialog to edit the site. The Site Setup dialog appears.
- 4 Select the Servers category to open the server settings window. Click the + sign to add a new server definition.

**Figure 6.16** Remote server settings are set in the Manage Sites window.

- A** Select the **Chapter 6** site.
- B** Click the pencil icon to edit the site settings.
- C** Click the + to add a new server.
- D** Set the site name to **Simulated Server**.
- E** Choose **Local/Network** from the drop-down menu.
- F** Browse to the `wuLvs-website` folder.



- 5 Enter **Simulated Server** as the server name.
- 6 Change the Connect Using field from the default FTP setting to **Local/Network**.
- 7 Use the Browse icon to navigate to the folder you created in step 1.
- 8 Leave the Web URL field blank, and check your settings against the ones in Figure 6.16.
- 9 Click **Save** to save the server settings. Click **Save** again to save the site settings. Click **Done** in the Manage Sites dialog to finish defining the remote server.

## Publishing to the Remote Server

With the remote server location designated, you're ready to see the process of publishing to a web server in action. You'll use the Files panel for these operations.

- 1 In the Files panel, select the home page for the website, `index.html`.
- 2 Click the Put button at the top of the panel (**Figure 6.17**).
- 3 Save the file if prompted, then click Yes when Dreamweaver asks if you want to put the dependent files for this page.

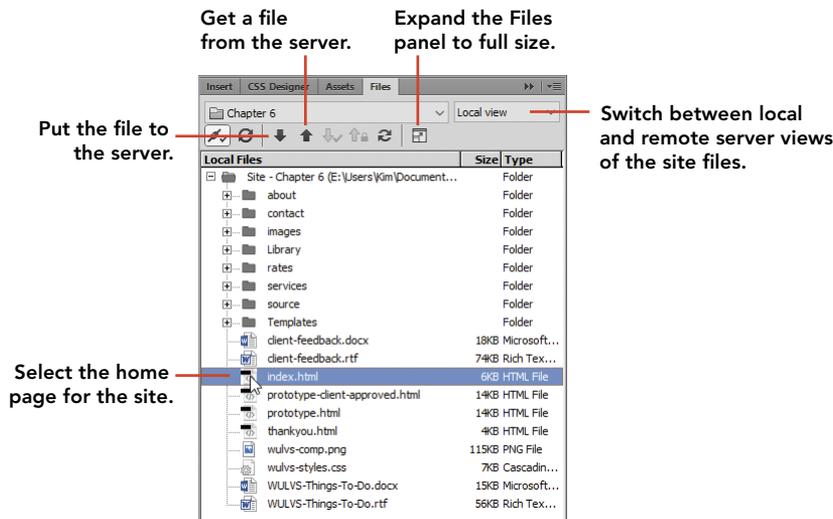
Dependent files are all the files that are linked within a document. Dreamweaver copies the image files and the CSS file that is linked to the document onto the server along with the page itself.

**NOTE** If you select the "Don't show me this message again" checkbox, Dreamweaver will automatically put dependent files to the server. This setting can be changed by choosing `Edit > Preferences (Dreamweaver CC > Preferences on a Mac)` and then selecting the Site category. By default, both `Prompt on Get/Check Out` and `Prompt on Put/Check In` are selected.

- 4 Switch the Files panel to the Remote Server view.

Dreamweaver has copied the home page and the images that are on the page along with the CSS file into the folder that you're using as a simulated web server.

You can now put the other pages for the site onto the server with one operation.



**Figure 6.17** File operations for a site are performed in the Files panel.

- Switch the Files panel to Local view, and then expand the view of the `about`, `contact`, `rates`, and `services` folders so that the `index.html` file in each folder is visible.
- While holding down the CTRL/Command key, click each of the `index.html` files along with the `thankyou.html` file. Click the Put button, and click Yes to including dependent files. Save the files if prompted.

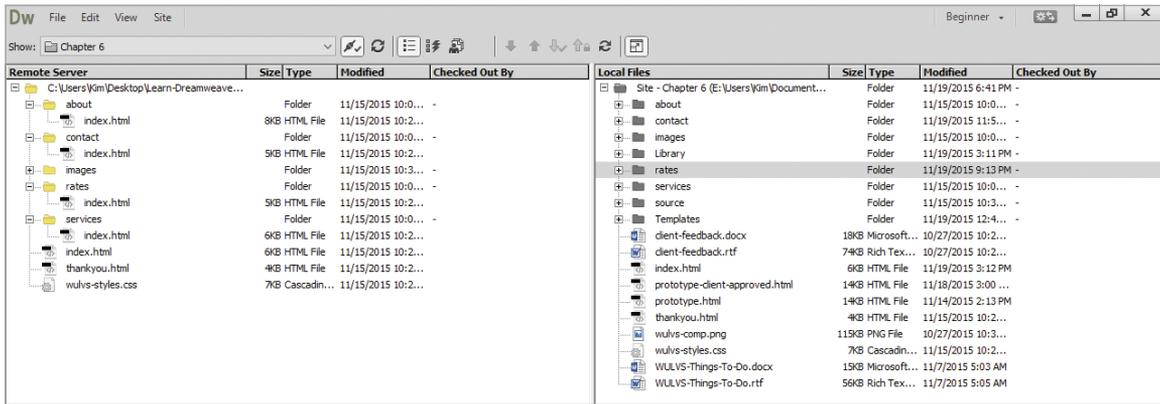
Dreamweaver will copy all the HTML files plus all the image files to the server.

- Open the expanded view of the Files panel (Figure 6.18).

When you compare the files on the server against those on your local computer, you'll see that all the files that make your website functional are present.

The source files and images that have not been used within the site remain on your local computer and have not been placed onto the server. Unless you are working with other people on the site and wish to share those files, they are not needed on the server.

**Figure 6.18** You can use the Expanded view of the Files panel to compare the files on your computer with those that are on the server.



## Updating Files on the Server

When you make revisions to documents on your local computer, you will need to upload the revised files to the web server so that the changes become visible.

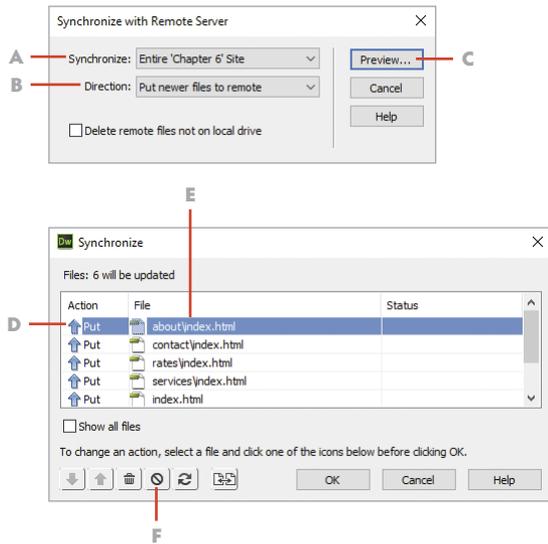
- Open the home page of the site, `index.html`.
- Place your cursor after the last bullet item. From the Assets panel, insert `wulvs-comic-sample.png` (the sample cartoon the client has provided), and provide alt text in the Property inspector.

- 3 Create a new class selector in the CSS Designer panel so that this image can be centered in its container. Name the new selector **.centeritem**.
- 4 Set the properties for the new selector by setting display to **block** and the left and right margins to **auto**.
- 5 Select the cartoon image, and assign the **.centeritem** class to the image. The cartoon will now be centered in the content container.
- 6 Choose File > Save All to save both the CSS file and the HTML file.  
The two files that have been changed must be put to the server so that the changes you made can be viewed online.
- 7 Select the **index.html** file and the **wulvs-styles.css** file, and click Put to copy them to the server. Click Yes when prompted for dependent files, and the cartoon image will also be copied to the server.

## Synchronizing Local and Remote Files

Dreamweaver allows you to synchronize modified files by comparing the files on your computer with those on the server. This operation is performed when numerous changes have been made and you need to put all the modified files to the server at once.

- 1 Open **wulvs-website.dwt** from the **Templates** folder.
- 2 Place your cursor directly after the Contact text in the left sidebar. Press Enter/Return to create a new list item in that location.
- 3 Type **Learn More**, highlight the text, and provide a dummy link for the text by typing the **#** symbol in the Link field of the Property inspector. The text will take on the appearance of the other buttons once the link is created.
- 4 Repeat the operation in the footer area of the page to create a new link in that location, using the **|** symbol to separate the new link from the existing ones.
- 5 Save the template file, and update all the pages that use the template.  
You now have multiple files that have been modified and saved by Dreamweaver.
- 6 Right-click the site folder in the top of the Files panel, and select Synchronize from the menu that appears.  
Dreamweaver will ask how these files are to be compared and which actions are to be taken (**Figure 6.19**).



**Figure 6.19** Synchronizing files allows you to put multiple modified files to the server in one operation.

- A** Choose selected files.
- B** Set direction to Put Newer Files to Remote.
- C** Click the Preview button to see the files to be synchronized.
- D** Lists the action to be performed
- E** Files that are to be put to the server
- F** Remove files you do not wish to put to the server.

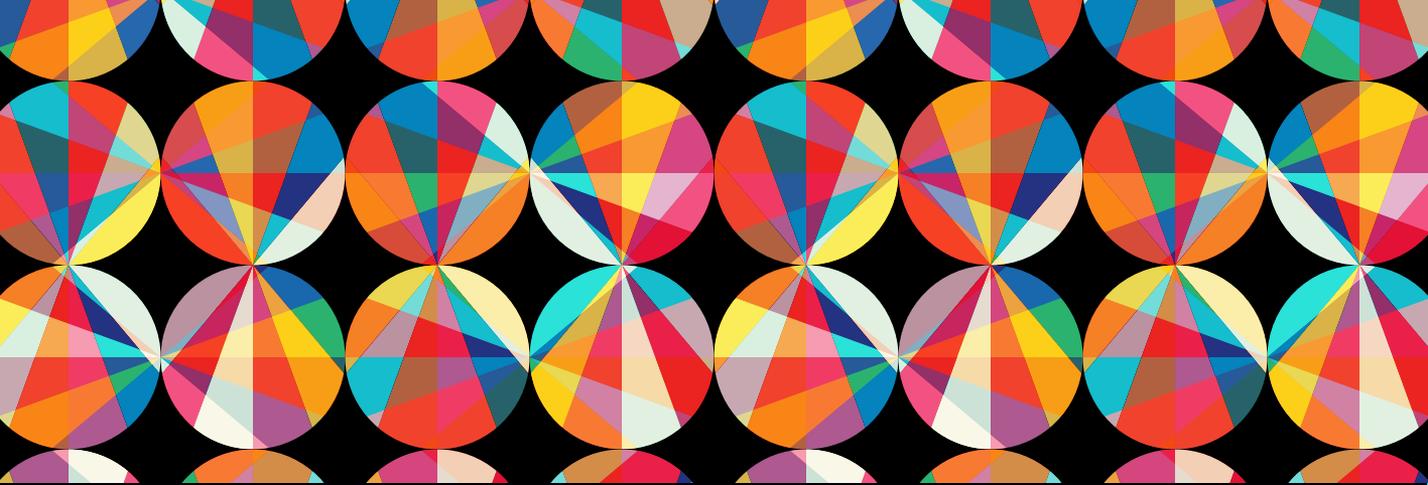
- 7 Set the Synchronize setting to Entire 'Chapter 6' Site, set the Direction setting to Put Newer Files to Remote, and click Preview.

You will see a report of the files that Dreamweaver has determined need to be copied to the server with the action that is to be performed.

- 8 Review the list of files. Those files that you do not wish to copy to the server, such as the source files, may be removed from the operation by selecting the file and clicking the Ignore button at the bottom of the window.
- 9 After you have reviewed your file, click OK and Dreamweaver will perform the requested action. In this case the files displayed will be put to the server.

# Conclusion

You're now ready. With the skills that you've learned, you can develop an entire website in Dreamweaver by carefully structuring the site, creating folders and files that are needed to meet the requirements of your client, and designing the style for the site using modern CSS styling methods. What's more, you can share your work with the entire world by publishing a website to a live web server, and you can modify and maintain the site when those inevitable changes happen.



## CHAPTER OBJECTIVES

### Chapter Learning Objectives

- Learn how mobile development is driving modern web design work.
- Create new documents in Dreamweaver using the included Responsive and Bootstrap starter files.
- Evaluate the time required to create responsive designs in preparation for giving the client a cost and time estimate.
- Use the Media Queries toolbar and the CSS Designer panel to examine and visualize responsive designs.
- Create and edit the media queries that define a fluid design.
- Understand how jQuery and Bootstrap components are used in responsive design.
- Create styling rules that override the default settings found in Bootstrap components.
- Insert HTML5 video into a web page and understand the methods for preparing video for production.
- Use Dreamweaver to write and edit CSS3 animations and transitions.

### Chapter ACA Objectives

For full descriptions of objectives, see the table on pages 240–246.

**DOMAIN 1.0**  
SETTING PROJECT REQUIREMENTS  
1.3

**DOMAIN 2.0**  
UNDERSTANDING WEB AUTHORING  
2.3, 2.5, 2.6

**DOMAIN 4.0**  
WEB AUTHORING USING ADOBE  
DREAMWEAVER CC  
4.1, 4.2, 4.4

**DOMAIN 5.0**  
TESTING, PUBLISHING, AND MAINTAINING WEB  
CONTENT USING ADOBE DREAMWEAVER CC  
5.1

## CHAPTER 7

# Exploring the Brave New World of HTML5 and Responsive Design

In the projects you've completed up to this point, you've created semantically correct HTML documents that are beautifully styled with cascading style sheets. And bravo for that! You've learned how to do things the right way, but with this one tiny caveat: Your design is not mobile ready. Yes, the web pages you created were to your client's specifications, and the *wulvs.com* website will look just fine on a desktop browser and on a tablet. But when it comes to phones? Sorry, but the fixed-width design you've created simply won't display properly on a phone.

But not to worry. Now that your client has started to consider mobile development and wants to look into a new method for creating pages—one that uses the very latest coding specifications, known as HTML5—you can work on some proofs of concept that could be presented as a solution to what he's asking for in this new development cycle. And as it turns out, Adobe Dreamweaver has some excellent resources already built into the program that will allow you to get a jumpstart on designing for mobile devices and for learning about the brave new world of responsive web design.

## Let's Go Mobile!



**Figure 7.1** Chris the Cartoonist

Chris the Cartoonist (**Figure 7.1**) continues to be pleased with the work you have done and is thinking of ways to expand.

**Chris Sez:**

*Hey, guess what? Things are really taking off with my web comic, and I have this crazy idea. I know, you think all my ideas are crazy, but hear me out.*

*The web comic has become so popular that people want to know more about the characters and even collect character profiles, such as baseball cards. Can you believe it?*

*Can you create a new website that would be viewable on a smartphone and could even be turned into a phone or mobile app? I'm thinking about calling the new website "Wulvs Collectibles." Oh, and I want to work in some video and animations too.*

*I know that's a tall order and it's a whole new design job for you, so there are two big questions to answer:*

*Can you do it?*

*How much would it cost?*

*Thanks. I look forward to getting your estimate.*

*Chris*

## Get Your Files Here

As you did in previous chapters, copy the folder that contains this chapter's files to your computer.

- 1 Download the project files for this lesson, named `chapter-07.zip`, from the Lesson & Update Files tab on your Account page at [www.peachpit.com](http://www.peachpit.com).
- 2 Unzip the files and copy the `chapter-07` folder to the `Learn-Dreamweaver` folder you created in Chapter 1.

# Define the Chapter 7 Website

With the files you've been provided for this chapter in the project folder, you're ready to set up the site in Dreamweaver.

- 1 From the application bar, select Site > New Site.
- 2 Name the site **Chapter 7**.
- 3 Browse to the **chapter-07** folder in your **Learn-Dreamweaver** project folder.
- 4 Click Save.

★ ACA Objective 4.1

## NOTE

Refer to Video 1.5, "Defining a Dreamweaver Website," if you want a quick demo of how the site definition process works.

★ ACA Objective 2.3

★ ACA Objective 4.4

# Understanding HTML5, CSS3, and Responsive Design

Before we dig into the first project, let's discuss a little about how HTML5 and the new set of styling rules known as CSS3 came into being. As it turns out, the project you've been working on is a perfect example of the two big challenges that were facing web developers as the web matured and new expectations for the delivery of web-based content came into play.

If you look back at the beginning of the journey to create the *wulvs.com* website, you'll recall that the structure of the site was created using five `<div>` tags. Each tag was assigned a name that we made up on our own but that followed the common naming conventions that web designers typically use. Thousands, if not millions, of web pages were built following this kind of structure and using names similar to, or the same as, the ones you used. Common practices were followed, and professional web developers did their best to follow the standards, but the standards themselves were loosely written. That was all fine for humans, but not so great for computer programming, where *naming* things is so very important.

But even as these methods were being refined and revised, many people began to recognize the limitations of this kind of design on two major fronts. First, without a standard naming convention it wasn't possible to efficiently build programming frameworks that could take advantage of what web browsers could do. There were just too many variables in how web pages could be constructed and not enough HTML tags to support standardization.

▶ **Video 7.1**  
Exploring  
Responsive Design  
with HTML5

And then the iPhone arrived.

The release of the iPhone completely changed the landscape of web design and the expectations of consumers. Suddenly, even the web-standard methods that web developers had been using for years became a real problem. The much smaller screen sizes available on iOS, and on the Android and Windows phones that quickly came along after Apple's success, simply couldn't display a web page that was designed for a computer screen in a version that gave a satisfying presentation to the viewer. And Apple's now-famous decision not to include Adobe Flash Player on its phones also presented tremendous problems for the creators of web video and interactive content.

It's not like any of this came as a surprise and every web developer woke up one morning to a world that was fundamentally different. In fact, the World Wide Web Consortium (W3C) had been working on solving some of these very questions:

- How can web browsers, device manufacturers, and companies such as Apple, Adobe, and Microsoft develop tools that allow web pages to be designed and delivered in a more standard way?
- How can developers leverage a new set of standards so that new methods can be devised that make programming for the web more efficient?
- How can the companies and programmers who make browser software conform to these new standards so that a web page will look the same when seen in the Firefox browser (from Mozilla), the Chrome browser (developed by Google), or the native browsers from Apple (Safari) and Microsoft (Internet Explorer and Windows Edge)? And how do these new standards translate into the browsers available on the iOS, Android, and Windows mobile platforms?
- How can browsers be programmed so that video and interactive content no longer requires a helper application, such as Adobe Flash, but can function natively within the browser?

After a lot of work by the W3C, new HTML standards were finally released and almost all the standards were accepted for use in the web browsers that were available on desktop computers, tablets, and smartphones. The new HTML standards were to become known as HTML5, and the new CSS standards were given the name CSS3.

These new coding standards addressed the issue of HTML tag compatibility. Instead of leaving things up to individual designers to name a `<div>` with the

id of header and style against those properties, new tags such as <container>, <header>, <section>, and <article> and many others were agreed to and became available as new browser versions were released.

These additional HTML tags and CSS specifications made it easier to tackle the other problem of how to scale and reformat a web page so that the experience of viewing on a phone is satisfying to the end user and allows access to the information or entertainment they're after. As of today, two methods of accomplishing this are available, and both are supported by Dreamweaver:

- Responsive design (sometimes called fluid design) lets you write conditional statements known as *media queries* into a document. Media queries are written into the CSS rules for a web page so that when the size of the screen changes, new styling rules are used to position and even hide certain parts of the page. In a responsive design, all the content of the page is loaded into the browser and CSS determines how the content is presented.
- Mobile design also uses HTML5 but takes things a lot further by using a **framework** of standard CSS rules along with a **JavaScript library** that actually reformats the content that is visible to the end user on the fly. This allows for much lighter web pages and provides a user experience that is similar to what someone would see in a native app built for their device. The most widely accepted framework in use right now is the **jQuery Mobile** JavaScript library and the **Bootstrap** collection of CSS styling rules. Dreamweaver supports this design environment, as do many other web programming applications.

Dreamweaver provides you with an excellent starting point for learning about these two design methods, with built-in starter pages for both. In the projects you'll tackle in this chapter, you'll see both design methods in action and learn how to provide access to multimedia and animations using the new HTML5 and CSS3 standards. Responsive design and mobile design are two skills that you'll want to develop if you're serious about becoming a web designer, and Dreamweaver provides a great entry point into this brave new world.

★ ACA Objective 1.3

★ ACA Objective 2.3

★ ACA Objective 2.5

★ ACA Objective 4.2

★ ACA Objective 5.1

▶ **Video 7.2**  
Exploring  
Responsive Design  
with Media Queries

## PROJECT 7.1

# Exploring Responsive Design with Media Queries

In the projects you complete in this chapter, you won't be working toward a final design prototype. In a typical design environment, you would create your design compositions in Photoshop or Fireworks and then translate the design into HTML and CSS. But where your client has only an inkling of an idea of what they want, it's not unusual to dig around and determine what kind of work you'll need to do. Proof-of-concept design exercises let you make a more educated estimate of the time it will take to do the work and the cost involved in getting the work done.

## Examining Client Requirements

In the site files for this chapter, you'll find instructions, in the document named `wulvs-collectibles-concept.rtf`, from Chris the Cartoonist on how he wants to get into mobile development. Open that file and review very carefully what your client is asking you to do.

You're really lucky. Chris is a designer too—he understands that new work costs money, and he's not asking you to build something for free. Many clients are likely to add new requirements without taking a look at the cost involved. This kind of **project creep**—where new work is tacked on with the expectation that the designer will do the job for little or nothing (and have it done immediately!)—is all too common. You should always avoid project creep (also known as scope creep) by discussing with the client the time and resources needed to add new work to a project and be clear with them as to the cost and time involved.

In a situation like this—where no budget has been established and you need to calculate the cost involved in a development project—you'll need to begin by considering what resources you have and the time required to assemble and organize what it will take to meet the client's requirements. Don't make the mistake of thinking of resources as just the client's images and the copy that's been written. *Time* is the single most critical resource you have as a designer, and calculating the time required to complete a job is also central to the project development process.

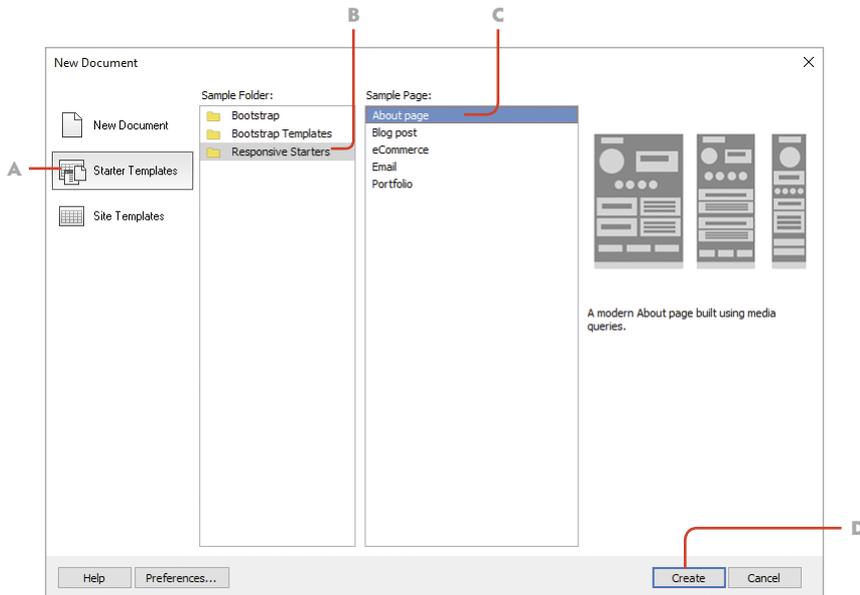
The most critical requirement you'll find in the client's document is his desire to get into responsive design by developing a prototype that will display on devices at all sizes. To do this (and to learn more about media queries and the Dreamweaver work environment), you'll turn to the starter templates that Dreamweaver provides.

## Creating a New Responsive Design Page

To learn about the design interface found in Dreamweaver and the programming methods used in a fluid design, you need to have a page to work with. Dreamweaver provides the resources you need to create this page in the New Document dialog.

- 1 With the Files panel open in Dreamweaver and Chapter 7 active, right-click the site folder at the top of the panel and select New Folder.
- 2 Name the folder **responsive** and press Return/Enter.
- 3 Choose File > New to open the New Document dialog.
- 4 Select the Starter Templates category, the Responsive Starters sample folder, and the About Page template, as you see in **Figure 7.2**. Click Create.

Dreamweaver will prompt you to save the file.



**Figure 7.2** The New Document dialog allows you to create new pages from the starter files that Dreamweaver includes.

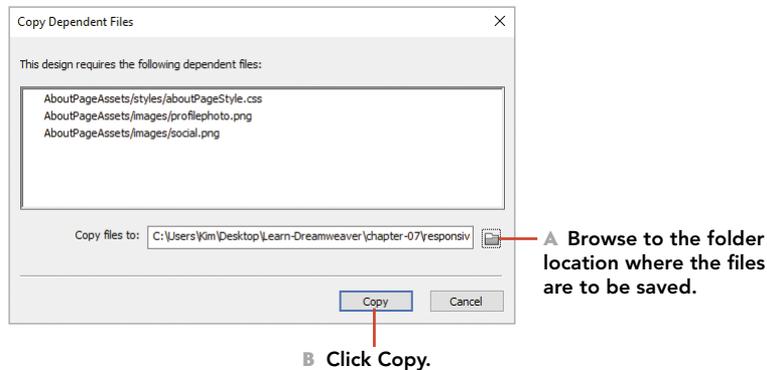
- A Select the Starter Templates category.
- B Choose Responsive Starters.
- C Select the About page starter file.
- D Click to create the new document.

**NOTE**

There is a known bug on OS X systems in which the system and folder names are the same. If you are not prompted to save the file, choose File > Save As. See the note in the Dreamweaver online forum that discusses this issue: [forums.adobe.com/thread/1921852.pdf](https://forums.adobe.com/thread/1921852.pdf).

**Figure 7.3** Copy the files that Dreamweaver provides to your local computer to make the page operational.

- 5 Select the **responsive** folder as the location to save the file, and name it **index.html**. Click Save.
- After you save the file, Dreamweaver will need to know where the related files that make this page functional are to be copied.
- 6 Ensure that the **responsive** folder is selected as the destination, and click the Copy button, as you see in **Figure 7.3**.



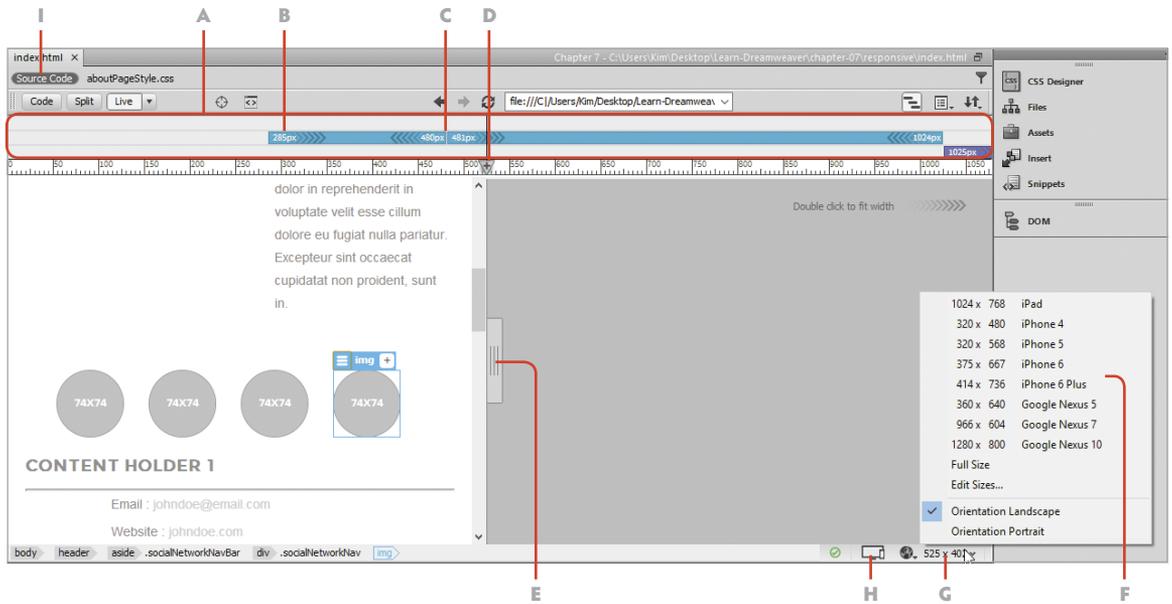
## Examining the Live View of a Responsive Design

When you move from a static page (like the one you built previously) to a fluid design (like the one you’ve just created), the difference between the Live and Design views of Dreamweaver becomes very apparent. When you work on a responsive design, you’ll use the interface elements that the software provides in Live view to visualize the design.

- 1 Switch to Live view, and take note of the different ways that Dreamweaver allows you to view this document.

**Figure 7.4** lists the many ways that you can modify your view of the page to see how the media queries that are linked to the `aboutPageStyles.css` file cause the content in the page to flow and resize.

- 2 Examine each of the interface elements listed here, and become familiar with their use. You can expect to be asked to identify these areas of the Dreamweaver UI if you take the “Web Authoring with Adobe Dreamweaver” ACA exam.



- A Media Query toolbar** displays the media query minimum and maximum widths as color-coded bars.
- B Media query minimum width** shows the smallest screen width supported with the query.
- C Media query maximum width** shows the largest screen width supported with the query.
- D Set new width breakpoint** by selecting the + sign icon. Dreamweaver will display new media query toolbars when this button is selected so that a new media query can be written to the CSS style sheet.
- E Scrubber handle** is used to manually drag the width of the page to new sizes.
- F Phone display presets** allow you to view the page at the width it would display on popular mobile phones.
- G Document view size** shows the dimensions that are currently being viewed. Click this button to display preset sizes to choose from.
- H Device preview** allows you to see a live view of a design on a mobile device either by using the QR code reader on a device or through a URL that is provided when you click this button. Note that this feature will not work in many school and training lab settings unless students have an individual (paid) Creative Cloud subscription and their own Adobe ID.
- I Related files toolbar** shows the CSS file and other files linked to this page.

**Figure 7.4** You can use Dreamweaver interface elements to examine and visualize responsive page designs.

## Examining Media Queries and HTML5 Tags

Media queries are conditional statements that are written into the body of a CSS style sheet. Media queries allow a web page to respond to the size of the screen by determining the screen size and using styling rules that are specific to those sizes. Understanding how media queries are written into a style sheet and how Dreamweaver allows you to access those styling rules is central to working in this kind of design environment. Let's examine the media queries that are in the attached style sheet for this page.

- 1 Right-click the `aboutPageStyles.css` file that is shown in the Related Files toolbar, and select Open as Separate File.
- 2 Examine the code and note how the different parts of the style sheet have been segmented, with comments starting with the `/* Global Styles */` section.

The first media query is located after the `/* Media query for Mobile devices */` comment.

- 3 Take note of the syntax used for writing a media query.

A media query begins with an `@` symbol, followed by the condition that is being evaluated. These conditions can be listed one after the other to build a complete query, as you see in your code—where the first condition is set to check the media and then evaluate the minimum and maximum widths.

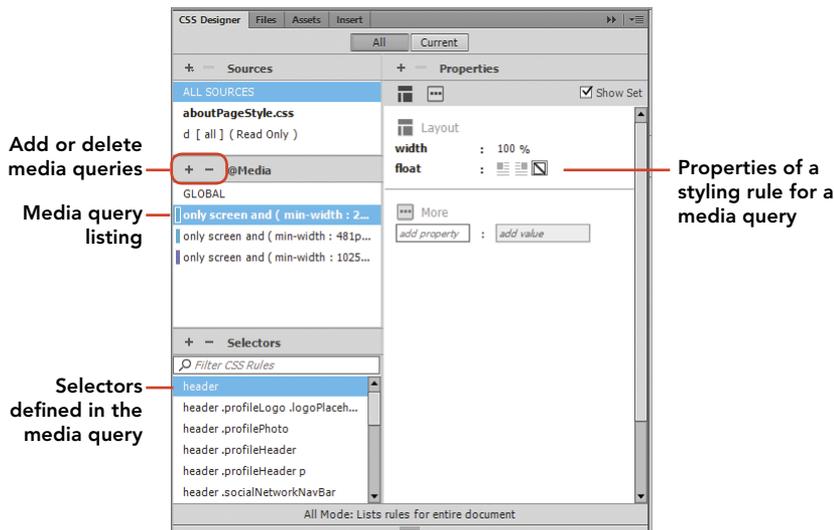
Media queries are written with the same opening and closing curly braces as all CSS styling rules.

```
@media only screen and (min-width : 285px) and  
→ (max-width : 480px) {  
}
```

- 4 Examine the closing tag for the first media query. You'll find the closing tag directly above the next media query comment `/* Media Query for Tablets */`.

All the rules between the opening and closing media query braces—color coded in a green-blue by Dreamweaver—are used to define the properties of elements that should be displayed *differently* when the screen is viewed at that size.

Dreamweaver displays the different queries that are present in the `@Media` section of the CSS Designer panel (**Figure 7.5**). This allows you to select a particular media query and see what selectors have been defined in those areas of the style sheet.



**Figure 7.5** Media queries and the styling rules defined within them are displayed in the CSS Designer panel.

## Creating a Responsive Prototype Page

Creating a prototype file from the client's request provides you with a good way to get comfortable with this kind of design work. In the rest of this project, you'll be given specific tasks to complete using the design tools that Dreamweaver provides. With the skills you've developed, you're ready to accomplish these on your own.

With `index.html` open and the information provided for you in the file named `wulvs-collectibles-concept.rtf`, complete the following tasks on your own:

- 1 Change the document title to **Werewolves UnLimited Collectibles: Ray the Werewolf**.
- 2 Replace the placeholder images in the page with the cartoon characters provided for you in the `images` folder.
- 3 Add a page background image to the design by modifying the styling rule for the `<body>` tag.
- 4 Change the text in the `<h1>` tag at the top of the document to **Werewolves UnLimited Collectibles**. Choose and apply an appropriate font style and size for the text.
- 5 Style the `<header>`, `<footer>`, and `<section class="mainContent">` tags, and apply a background color that contrasts with the page background.

### TIP

Use the *Browse for File* button in the Property inspector to locate the image of the correct size to replace the ones in the starter page.

**TIP**

Use the search function in the Selector section of the CSS Properties panel to filter the rules to show those that include `:hover` in the selector name.

- 6 Examine the design as it appears at different widths. You'll note that the text in the document is crowded against the edges of the container when viewed at a phone-size screen width. Determine the selector and the media query required to modify the padding property to improve the design.
- 7 Create a new styling rule within the existing media query to modify the properties of the containers where you note that the text requires additional padding.
- 8 Examine the rollover effect that exists for the four small 74 x 74 pixel images that you replaced. Modify the selector that uses the `:hover` pseudo-class to create the rollover effect.
- 9 Replace the sample text that Dreamweaver provided with the text contained in the `wulvs-collectibles-concept.rtf` file. Add additional headings and other text as necessary to complete your prototype.



## LEVEL UP: YOUR PERSONAL PORTFOLIO PAGE

This document provides you with a perfect opportunity to create your own custom profile page. Using the techniques you learned here, you could define a new Dreamweaver site and then use the New Page dialog to make a new responsive design just like the one in this project. From there, you can replace the images with ones of your own and build a personal portfolio that describes who you are, what you've accomplished in your life, and what your goals are for the future. That gives you the chance to continue building your coding skills while reflecting on how you might want to present yourself to a future employer or client.

## PROJECT 7.2

# Exploring Responsive Design with jQuery and Bootstrap

- ★ ACA Objective 2.3
- ★ ACA Objective 2.5
- ★ ACA Objective 4.2
- ★ ACA Objective 4.4

You've now seen the first of the common methods for creating responsive web pages. In the second mobile development environment supported natively by Dreamweaver, the way that the page is coded for mobile viewing is fundamentally different. In a jQuery workflow, a JavaScript file is used to query—or evaluate—the size and orientation of the screen. Using the standard library of functions developed and published as a free and open source resource, jQuery has quickly become the world's most popular JavaScript library. According to Wikipedia, jQuery is used in “65% of the top 10 million highest-trafficked sites on the Web” ([en.wikipedia.org/wiki/jquery](http://en.wikipedia.org/wiki/jquery)).

The companion library that works right alongside jQuery in Dreamweaver is found in the styling rules created and supported in another open source project, by the good people at Twitter. The Bootstrap CSS file that comes included with Dreamweaver is massive—over 6000 lines of code. That central Bootstrap style sheet provides access to a set of standardized visual properties but also allows Dreamweaver to provide you with components that can be inserted via the Insert panel or Insert menu.

The cool thing about both of these libraries is the tremendous number of people around the world who now use them and contribute to their future success. Hey, that could be you someday! Many of these phenomenal coding achievements—and many others—were unlocked by people who are your age or just a little older.

The jQuery and Bootstrap development world is supported by thousands of tutorials and other online resources—not to mention entire books and video series—and you can find all sorts of open contributions and ongoing discussions on the best way to take advantage of the power of these tools.

In this project, you'll create a new page using the starter pages that come with Dreamweaver, examine the code that makes jQuery development possible, see how to override Bootstrap styling properties, and determine what resources you'd need to take a proof-of-concept file into production.

 **Video 7.3**  
Exploring  
Responsive Design  
with jQuery and  
Bootstrap

## LEVEL UP: DREAMWEAVER SNIPPETS

In addition to the way that Dreamweaver allows you to insert Bootstrap and jQuery components while in the Live view environment, there is an extensive library of code snippets that do the same sort of thing in Code view. If you open the Snippets panel by selecting Window > Snippets you'll find a huge list of prewritten Bootstrap components, as well as CSS3 animations, JavaScript, and other code blocks that can be inserted directly into the code of your page.

## Creating a New Bootstrap Design Page

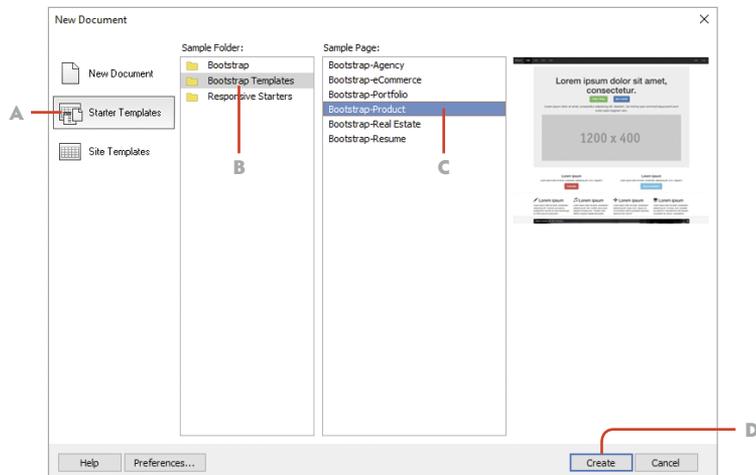
Just as with the responsive starter page you just worked on, Dreamweaver provides all the resources you need to create a new web page that uses jQuery and Bootstrap components in the New File dialog.

- 1 With the Files panel open in Dreamweaver and Chapter 7 active, right-click the site folder at the top of the panel, and select New Folder.
- 2 Name the folder **bootstrap** and press Return/Enter.
- 3 Choose File > New to open the New Document dialog.
- 4 Select the Starter Templates category, the Bootstrap Templates sample folder, and the Bootstrap-Product page template. Click Create to make the new page, as you see in **Figure 7.6**.

Dreamweaver will prompt you to save the file.

**Figure 7.6** The New Document dialog allows you to create new pages from the starter files that Dreamweaver includes.

- A** Select the Starter Templates category.
- B** Choose Bootstrap Templates.
- C** Select the Bootstrap-Product page starter file.
- D** Click to create the new document.



- 5 Select the `bootstrap` folder as the location to save the file, and name it `index.html`. Click Save.

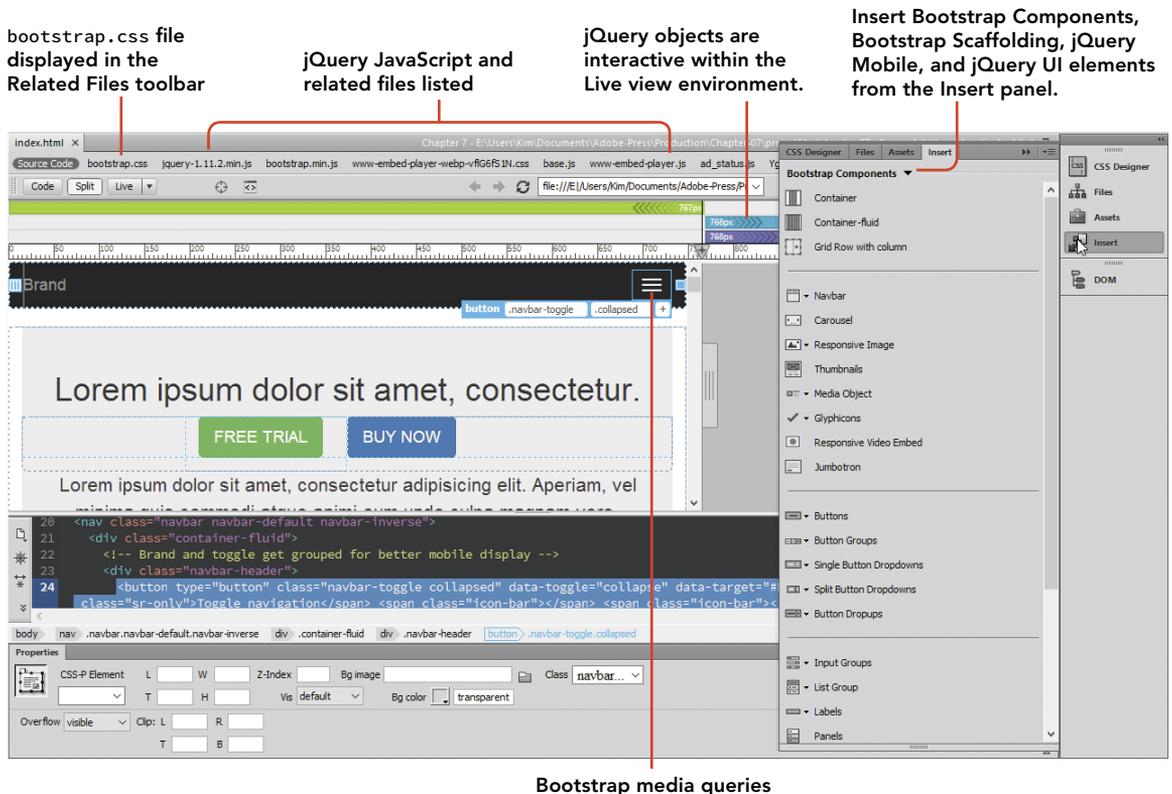
After you save the file, Dreamweaver will need to know where the related files that make this page functional are to be copied.

- 6 Ensure that the `bootstrap` folder is selected as the destination, and click Copy.

## Examining Bootstrap and jQuery Components in Dreamweaver

After you save the new document and copy the related files to your local site, you'll be able to examine the page and all the component and related files that Dreamweaver provides. Take a moment to become familiar with how Dreamweaver displays this page and the tools you can use when working in this environment. **Figure 7.7** shows the major elements of the Bootstrap and jQuery toolset that Dreamweaver provides.

**Figure 7.7** The major user interface controls that you'll use when working with jQuery and Bootstrap in Dreamweaver



- 1 Switch to Live view, and note the files listed in the Related Files toolbar. You will see a link to the single `bootstrap.css` file that Dreamweaver provides as well as links to the two JavaScript files linked to this page.
- 2 Open the `bootstrap.css` file, and note how some elements in this design have been styled using the standard Bootstrap styles. These design properties are intended to provide a standard starting point for styling a page.
- 3 Click the `jquery-1.11.2.min.js` file listed in the Related Files toolbar. (Your file version number might be slightly different.) You'll find a version of this JavaScript file that has been **minified** by removing all the line breaks and white space in the code to make it as compact as possible.

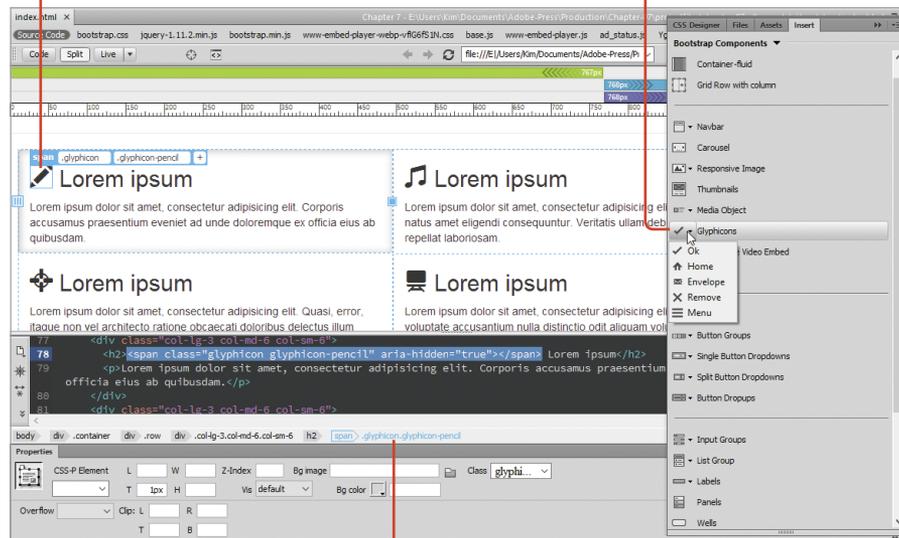
Although this file is meant only for your computer to read, there's still plenty that you can learn about how jQuery works by visiting the home page for the project at [jquery.com](http://jquery.com), where you'll find tutorials, sample files, and a community of developers working to make the most of what jQuery provides.

Note the small icons that appear above the video section of the page (**Figure 7.8**). These snippets of computer code create the images in the page and are inserted by

**Glyphicons present in the page**

**Insert Glyphicons from the Insert panel**

**Figure 7.8** Glyphicons are included in the sample Bootstrap page, and others may be inserted from the Insert panel.



**Glyphicons are assigned by setting a class on a `<span>` tag.**

applying a Bootstrap styling rule to a `<span>` tag. Dreamweaver provides access to the most common **glyphicons** in the Insert panel, but many more of these user interface components are available and already included in the Bootstrap CSS file. You can find a complete library of the available glyphs at [getbootstrap.com/components](http://getbootstrap.com/components).

jQuery and jQuery Mobile components are fundamentally different from media queries in how they provide a different size and appearance for design elements within a responsive design. Although media queries change the display properties of items that are already in the document, the programming behind jQuery allows those components to change on the fly. As the screen size changes, the conditional variables written into the JavaScript files present the appropriate version of the component. You can enable the Live Code view using the button on the Document toolbar to see how the code is rewritten when jQuery components are activated.

Dreamweaver provides numerous jQuery components that can be accessed from two categories in the Insert panel: jQuery Mobile and jQuery UI. From those panels you can insert any number of prebuilt jQuery widgets, including accordion menus, date pickers, password fields, and many more.

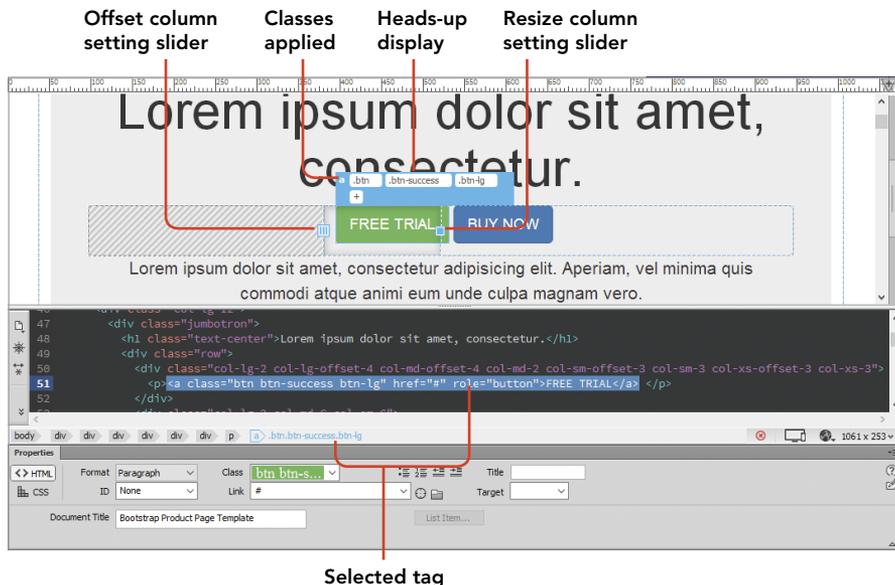
## Using Live View to Modify Bootstrap Properties

Although the Bootstrap CSS file contains many styled elements, you're certainly not limited to those styles. Dreamweaver provides a visual environment that lets you modify the classes that are applied to a selected object—you simply drag the handles that allow you to resize and offset objects in your design.

- 1 Select the button labeled Free Trial just below the banner text. You may find that you need to select the `<a>` tag in Split view used on this object to select it properly. Your screen should match **Figure 7.9**.
- 2 Drag the Offset column slider on the left to a new position to the left or right. Note that as you do so, the classes applied to the `<a>` tag have become quite different.

Because a standard set of named classes are in place in the Bootstrap CSS file, you do not have to touch any of the styling properties when you want to resize or reposition objects. Instead, the `<div>` tag that contains the button is assigned a whole new set of classes, which are dramatically different from the ones you began with.

**Figure 7.9** Dreamweaver uses a heads-up display to indicate the classes applied to an object and presents resize handles to change the class setting.



Dreamweaver provides a live update to the heads-up display to show the revised class structure. With more room available in the row, you can even add a new column by selecting the Add New Column button (**Figure 7.10**).

- 3 Explore the Live view user interface by dragging components into different positions.

You can do all sorts of experimentation by expanding columns, moving columns, and changing the dimensions of containers and their contents within the page, but notice that when you do so, the styling rules in the CSS never change. You are simply assigning new classes to the existing tags, and only the HTML document is being revised.



**Figure 7.10** Dreamweaver's heads-up display shows the revised classes applied to a container and allows you to add a new column.

Is the Live view user interface perfect? Well, no, the Live view interface in Dreamweaver is still a little buggy, and you may find that in this version it can be difficult to navigate and select items in a logical way. But Adobe is still actively improving the performance of Dreamweaver and in particular tinkering with Live view. It may be that the next version of Dreamweaver makes things much better.

# Creating a Custom CSS File to Override the Bootstrap Settings

It would be a very dull world indeed if all web pages had to look like the default Bootstrap template. To change the design of a page or the components in the page, a custom CSS file is required. This file is used to hold alternate classes and custom CSS selectors that will be displayed in place of the Bootstrap selector properties.

- 1 You can continue working on this file or revert it to its initial settings. Or you can simply make a new file by repeating the steps you followed at the beginning of this project.

To override the Bootstrap styling rules, a new CSS file with new selector settings will need to be created.

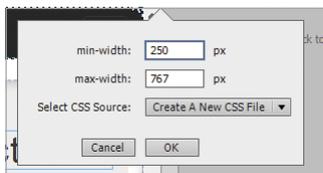
- 2 Position the scrubber so that the + sign indicator appears in the Media Queries toolbar, as you see in **Figure 7.11**.



**Figure 7.11** Dreamweaver allows you to add new media queries and set their width properties in the Media Queries toolbar.

- 3 Click the + sign to add a new media query.
- 4 Click the green bar to the left to establish a maximum width for the media query. When you do so, a pop-up window will appear.
- 5 Set the Minimum width value to **250 px** and leave the setting to create a New CSS file. Click OK when your settings match those in **Figure 7.12**.

You will now be prompted to create and save the new CSS file.

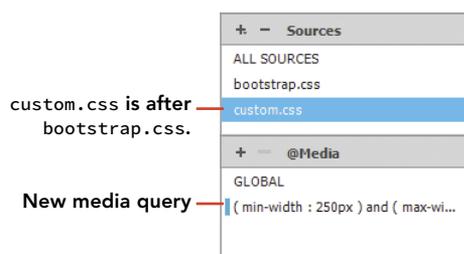


**Figure 7.12** The minimum and maximum widths of a new media query can be set in the pop-up window that Dreamweaver displays.

- 6 Click the Browse button, set the file location as the `bootstrap` folder, and name the file `custom.css`. Click the Save button in your Explorer or Finder window to save the file.
- 7 Click OK when you are returned to the Create a New CSS dialog.

The CSS Designer panel now displays the two source CSS files linked to this document, as you see in **Figure 7.13**.

It is important that the `custom.css` file follows the `bootstrap.css` file in the source order. Browsers interpret the last setting it encounters. Your custom settings will be applied only if they come after the Bootstrap file.



**Figure 7.13** Dreamweaver displays the two CSS files in the Sources section and the new media query dimensions.

## Applying Override Settings in a Custom CSS File

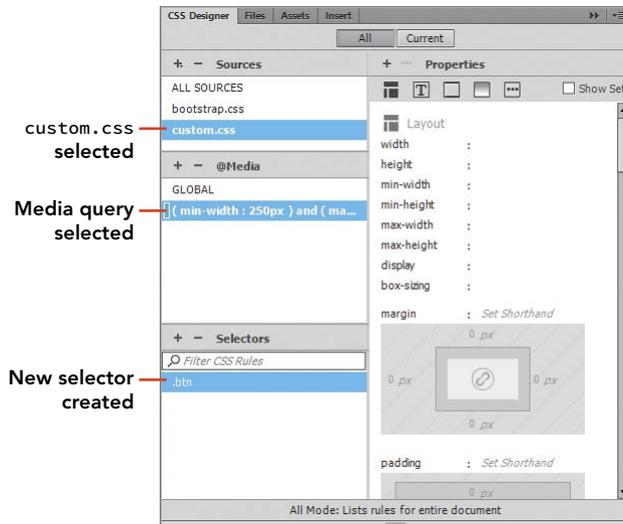
Once your custom CSS file has been created, you can set about creating new selectors that will override the Bootstrap settings. Remember that you aren't changing the Bootstrap source file in any way. You are merely writing new styling rules that the browser will display *instead of* those found in the Bootstrap file.

- 1 Select the link that defines the Free Trial button. You may need to go into the code to ensure that this tag is selected.

```
<a class="btn btn-success btn-lg" href="#" role="button">
```

- 2 You can use the CSS Designer panel to evaluate the existing settings by setting the panel to show the Current setting. Then select the Show Set checkbox and scroll through the list of styling rules. You are only investigating at this point, not writing new styling rules.
- 3 The rule you are searching for is the one that applies padding to the button container. This rule is contained in the `.btn` selector.

- 4 Switch the CSS Designer panel to the All setting, then select the `custom.css` file in the Source section and the media query listed in the @Media section, as you see in **Figure 7.14**.



**Figure 7.14** Preparing to override a Bootstrap styling rule

- 5 Click the + sign in the Selectors section, and filter the results down until you see the `.btn` class listed. Press Enter/Return to create the `.btn` class selector.
- 6 Apply new padding settings to all four dimensions until the button becomes smaller. Remember that a button on a phone still needs to be selectable by using a finger, so don't make them too small.
- 7 Scrub the page width to phone size, and you'll see that the buttons are still overlapping. The container that holds the button must be changed to control the positioning of the button.
- 8 Click the `<div>` that contains the Free Trial button. You'll see that a huge string of descendent selectors are applied to the `<div>`. This is the one that you'll duplicate and override in your `custom.css` file.
- 9 Select the `custom.css` media query in the CSS Designer panel, then click the + sign in the Selectors section to add a selector. You should make this new selector as specific as possible by pressing the Down Arrow key on your keyboard so that it matches the Bootstrap selector.

**10** Modify the padding and margin properties in the new styling rule until the containers holding the buttons are an appropriate size for viewing at a phone width.

Once your design looks satisfactory to you and you've checked it at various screen sizes, your work on this exploratory mission is complete.

**11** As always, you should save your HTML file and the new `custom.css` file you've been modifying.

In this project, you've had an opportunity to see just how to work in a design environment that includes all three of the modern web design methods that are rapidly gaining ground around the world. In combining HTML5, CSS3, jQuery JavaScript files, and Bootstrap CSS components, much has changed in how responsive web pages are built.



### **LEVEL UP: BROWSER INSPECTORS**

Dreamweaver certainly has a wealth of different ways to visualize and edit a design, but even with all the different panels and inspectors and interface widgets that are included, sometimes it's easier to inspect a design using the very cool inspectors that come built into almost all modern browsers. Mozilla Firefox and the cool little browser plugin called Firebug have long been favorites among web developers. But the built-in inspectors in Google Chrome or Apple's Safari browser also do a great job of allowing you to see the DOM structure of a page, look at the CSS styling properties that are applied, and view your design at various sizes. You'll likely decide on your own favorite, but in all those browsers, you can start up the page inspector by right-clicking any web page and choosing Inspect Element.

## PROJECT 7.3

# Inserting HTML5 Video

Our next stop on this exploratory mission into HTML5 is the new video tag that was released as part of the new HTML specifications. It was just a few short years ago that most online video was published in the Flash video format and was viewed through a Flash video playback component that was included in Flash Player. Flash playback was—and still is—incredibly good at delivering video to a web browser.

Unless of course that browser happens to be on an iPhone or iPad. And therein lies the rub, and the need for a separate video tag within HTML5 that provides playback capability right within the native browser. And it's nearly perfect, except for a couple of minor details. First, while most modern browsers support the MP4 format, the encoding technologies behind the MP4 file format belong to any number of copyright and patent holders. Big companies currently pay licensing fees to build encoders like the one you'll find in Adobe Premiere Pro, but that runs contrary to the more open nature of the web. So new video formats have been developed to address the openness of web video formats. Google is the lead proponent of the WebM format, and others continue to push for a third format, known as Ogg Vorbis.

Dreamweaver's solution? Use all three files when you insert HTML5. And with a little bit of advance preparation, it's a snap to add HTML5 video to a web page.

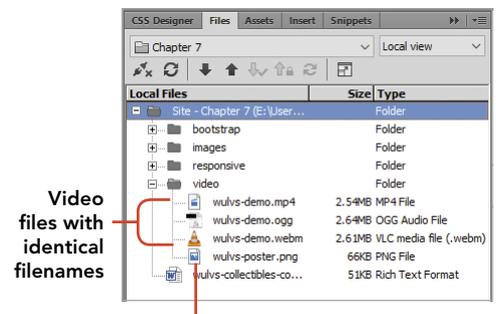
## Examining the Video Assets

The project files for Chapter 7 include a folder named **video**, where you'll find the resources that are required to insert HTML5 video into this page. Since there are some very specific requirements that you'll need to meet to make the process of placing video onto a page, let's start by taking a look at the files provided.

- 1 Open the Files panel for Chapter 7, and expand the **video** folder, as you see in **Figure 7.15**.
- 2 Identify the three video files included for this project. These files have identical filenames but different file extensions.

- ★ ACA Objective 2.3
- ★ ACA Objective 2.5
- ★ ACA Objective 4.2
- ★ ACA Objective 4.4

▶ **Video 7.4** Inserting HTML5 Video



**Image file to show before the video is played**

**Figure 7.15** Files required for HTML5 videos

Some variation still exists in which browsers support different video file types. All three video types are required to be universally compatible with all the web browsers in use. And that applies only to modern browsers. If you needed to support *really* old web browsers, you might even need to include Flash video as a fallback file format.

The MP4 file format is compatible with most web browsers and will usually be the first one displayed. MP4 files are the first ones to be processed in most productions, and they must use H.264 video **encoding**. That encoding format option is present in the Adobe Media Encoder that comes included with the Creative Cloud desktop applications.

The two additional videos are created by converting an MP4 source file to the **webm** and **ogg** file types. There are many online file converter services where you can upload your files for conversion, or you can use a free software program such as VLC Media Player to convert the file to different file types. In a commercial production environment, the individuals who are producing the videos would take on this task and hand the files off to the web team for inclusion in a project.

For this project, that work has been done for you on a short sample video.

- 3 Identify the image file included for this project, `wulvs-poster.png`.  
A poster image will be displayed in the area of the page where the video is inserted before the viewer of the page plays the video. This file has been prepared at the same pixel dimensions as the video itself, allowing for a seamless transition between the static view of the video and how it appears when it begins playing.

## Inserting HTML 5 Video

The starter page that you're working with already contains a video that was placed into the document by way of an embed code that points to a YouTube video. In many settings, it's preferable to have videos hosted on a dedicated server rather than using the YouTube servers.

- 1 Open the `index.html` file that is located in the `bootstrap` folder. You can make a new page following the steps at the beginning of Project 7.2 if you need a fresh version of this file.
- 2 Switch to Design view, and scroll down the page until you locate the Video Tutorial section. Select the video object contained within the `<iframe>` tag. Switch to Split view to ensure that the `<iframe>` tag is selected.

- 3 Press Delete to remove the embedded video.
- 4 Directly in the code, add placeholder text inside the `<div>` tag that reads **video here**. This is done simply to make it easier to see the location of the `<div>` where the new video will go. Click the Refresh button in the Property inspector.
- 5 Highlight the placeholder text, and open the Insert panel. In the HTML category, click the HTML5 Video button near the bottom of the panel.

This action will remove the placeholder text and replace it with the `<video>` tag. The Property inspector displays all the settings for this tag.

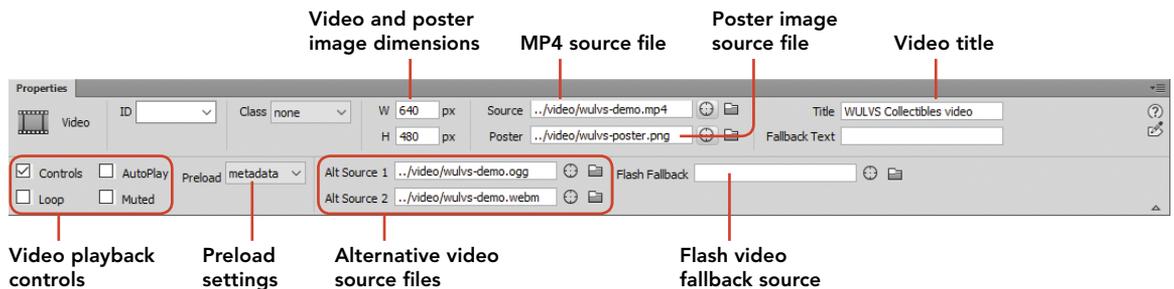
- 6 Click the Browse icon next to the Poster setting, as you see in **Figure 7.16**. Browse to the `wulvs-poster.png` file, and set it as the image source. This will establish a width and height for the video player.
- 7 Click the Browse for File icon next to the Source setting. Browse to the `wulvs-demo.mp4` file, and set that as the primary source file to be played.

When you perform this action, Dreamweaver will automatically populate the Alt Source fields with the other videos, provided they are in the same location and have the same filename as the primary source.

- 8 In the Title field, give the video the title **WULVS Collectibles video**.
- 9 Set the Preload value to `metadata`.

This allows the browser to load a description of the video and other information before the video is ready to play. You can also have the entire video preload, but this may cause the page to load slowly and should be tested prior to putting that setting into production.

- 10 Review the settings as shown in **Figure 7.16**, and then save your file. You have now completed this mission.



**Figure 7.16** The Property inspector settings for the `<video>` tag

The actual process of inserting HTML5 video is quite easy as long as you understand that the majority of the work has to take place before you turn to Dreamweaver. This involves creating, formatting, and converting video files into the three file types in use. At least for now. Web standards and practices are constantly shifting, and in the future you may see that one of the open source videos will replace the MP4 format as the standard for delivering video on the web. But for now, you can see just how this video will be displayed on different devices by changing your view of the page. In a responsive design, the video and poster image will scale up and down as the page size changes, giving end users exactly the kind of experience they expect.

## PROJECT 7.4

# Exploring CSS3 Animation

★ ACA Objective 2.6

★ ACA Objective 4.4

▶ **Video 7.5**  
Exploring CSS3  
Animation

You've likely seen CSS3 animations in action in your daily web surfing, whether you knew it or not. CSS3 animations and transitions are used routinely to display animated flyout menus, rollover effects that reveal more information, and animations that ease images in and out of a container.

Adobe has jumped on the CSS3 animation bandwagon in a big way. The Adobe Edge Animate product is built entirely to take advantage of the animation features that are part of the CSS3 specifications, and it's a great tool for creating banner advertisements, product displays, and user interface elements that operate smoothly with a minimum of code. Edge Animate has features that allow you to go from a design environment, such as Adobe Illustrator, and jump directly into animating HTML. Dreamweaver even includes a simple interface that allows you to insert code written in Edge Animate by using the Insert panel.

At the time of this writing, however, Adobe has announced a complete rethinking of its animation production tools and has decided to rename the software program previously known as Flash CC to Animate CC. This reworking of Flash comes with a greater emphasis on publishing to HTML5 formats and the retirement of the Edge Animate program.

Dreamweaver itself hasn't really kept up on the user interface side of things, but there are two important reasons to be able to read and understand CSS styling rules that are written into a style sheet, whether those styles are written in Edge, Flash, the new Animate program, or even the CSS styles that are available for free from places like Code Pen (*codepen.io*).

First, you or your team are going to need a way to manage and publish that content, and only Dreamweaver allows you to edit, publish, *and* work with others in a single design environment.

Second, you'll need to have an understanding of how CSS animations are created and the theory behind the code if you want to edit or modify code that's been written by someone else.

In this project, you'll create a very simple animation in Dreamweaver and see how the structure of the HTML comes together with the animation methods available in CSS3.

## Creating the Animation Page and Containers

In this project, you'll be working to understand the principles behind CSS3 animation, and you'll need only a simple web page to see animations in action.

- 1 In the Chapter 7 Files panel, right-click the site folder, and choose New Folder. Name the folder **animation**.
- 2 Choose File > New to create a new page. In the New Document dialog, set the category to New Document and the document type to HTML. With Framework set to None and Doc Type set to HTML5, click the Create button to make the new page.
- 3 Save the page in the **animation** folder, and name it **index.html**.
- 4 Change the document title to **Ray's Animated Card**.
- 5 Set the view of the document to Live/Split, and place your cursor directly after the opening `<body>` tag. Press Enter/Return to create a new line in the code.

The way that the `<div>` tags are nested in this instance is very important, and it's often difficult to get Dreamweaver to nest elements properly in Live view. Working directly in the code is preferable in cases like this.

- 6 Insert the following code block between the opening and closing `<body>` tags.

```
<div class="card">
  <div class="infobox">
    <p>Ray the Werewolf</p>
    <p>President & CEO</p>
  </div>
</div>
```

The classes applied to the `<div>` tags will be used to assign styling properties to the image and to the box that contains the information about the character.

- 7 In the code, place your cursor after the `<div class="infobox">` tag.
- 8 Use the Insert panel to insert the character image by browsing to `images/ray-profile-photo.png`. Note that the size of the image is 212px x 257px. You'll use the width and height values for setting styling properties later.
- 9 Enter the Alt description **Ray** for the image, and save your file.

## Styling the Card Container

With the structure of the HTML determined, you can now set about styling the elements of this design using the same methods you've used in the past to create new selectors in the CSS Designer panel.

- 1 In the CSS Designer panel, click the + sign in the Sources section. Choose Define in Page to insert a `<style>` tag block into the head of the document.
- 2 Select the `<div class="card">` tag in the code or by using the Tag selector.
- 3 In the CSS Designer panel, click the + sign in the Selectors section to create a rule. Dreamweaver will automatically name the selector `.card`.
- 4 In the Layout category of the CSS Designer panel, set the width and height to match the image dimensions of the image that you noted in step 8 of the previous section: 212px width and 257px height.
- 5 Set the Position property to `relative`.
- 6 In the More category, click inside the Add Property field, and type `cursor`. Press Enter/Return to assign the property to the class, and then select Pointer from the options provided.
- 7 Add a second property in the More category by typing `overflow`. Choose Hidden from the options available. Your settings should now match those you see in Figure 7.17.



**Figure 7.17** You can add additional styling rules using the More category of the CSS Designer panel.

## Styling the Image Container

Next, you'll style the `<img>` tag to fix the size of that container and set a position property that will allow the box with the character's information to appear over the top of it.

- 1 Select the `<img>` tag in the Tag selectors or directly in the code.
- 2 Click the + sign in the Selectors section of the CSS Designer panel to create a selector. Set the new selector name to **.card img**, and press Enter/Return.
- 3 In the Layout category, set the Position property to **absolute** and the left position to **0px**.

## Styling the Information Box Container

Now you can turn your attention to the `<div>` that contains the information that will appear when the cursor is placed over the image. In these steps, you'll create a new styling rule to define this container.

- 1 In the Tag selector or directly in the code, select the `<div class="infobox">` tag.
- 2 Click the + sign in the Selectors section of the CSS Designer panel to create a selector with the name **.card .infobox**.
- 3 Use the Background category of the CSS Designer panel to set a medium gray background color for the selector.
- 4 Use the Layout category to assign an **absolute** position to the selector and to assign width and height values that match the dimensions of the image.
- 5 In the More category, add a new property by typing **z-index** into the New Property field. Set the value of this property to **100**. This will set a stacking order that places the contents of this container above the image.

When you've finished adding these properties, the contents of the `<div>` with a class of `.infobox` will cover the image. Just as expected!

## Styling the Information Box Text

The final styling rules will be used to set properties for the text that is inside the `.infobox` container. The name and title of the character are contained within paragraph tags that you'll now style against.

- 1 Select either of the paragraph tags in the document.
- 2 Click the + sign in the Selectors section of the CSS Designer panel, and accept the suggested selector name of `.card .infobox p`.
- 3 In the Text category, set the color of the text to white and set text-align to center.  
You can assign different styling properties to the text as you choose, using padding and font properties to position and style the text.
- 4 Save your file.

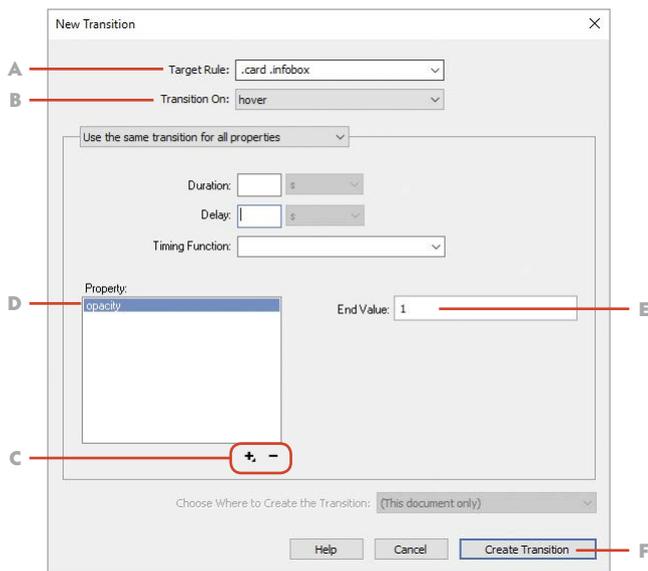
## Styling Animations with CSS Transitions

Now that all the containers and their styling properties are in place, you can begin animating the interaction that occurs when the cursor is positioned over the image. First, you'll hide the container with the character's information and then set a CSS hover pseudo-class to reveal the `<div>`.

- 1 In the CSS Designer panel, select the `.card .infobox` selector.
- 2 Using the Layout category, add an opacity setting of 0 to the selector.  
The container will disappear from view as it becomes transparent. All the code and information inside the container is still present, but using CSS you've just hidden it from view.  
In Dreamweaver, animations can be added and edited in the CSS Transitions panel.
- 3 Choose Window > CSS Transitions to open the CSS Transitions panel.
- 4 Click the + sign at the top of the panel to open the New Transition dialog that you see in **Figure 7.18**.
- 5 From the Target Rule drop-down menu, select the `.card .infobox` selector.

**Figure 7.18** The New Transition dialog allows you to choose a selector and create a new styling rule.

- A** Selector being targeted by the transition
- B** The action that triggers the transition
- C** Add or remove properties to be changed
- D** The property to be changed on transition
- E** The opacity value to change the container from transparent to opaque
- F** Create the new CSS selector



- 6 Set the Transition On setting to **hover**.
- 7 Click the + sign in the Property section, and choose `font` opacity. Set the End Value to **1**, and then click Create Transition.

When you place your cursor over the image in Live view, you'll see the information in the `.infobox <div>` tag come leaping into view. Move your mouse away and the box instantly disappears. That's all a little too abrupt, but you can add additional transition properties to smooth things out.

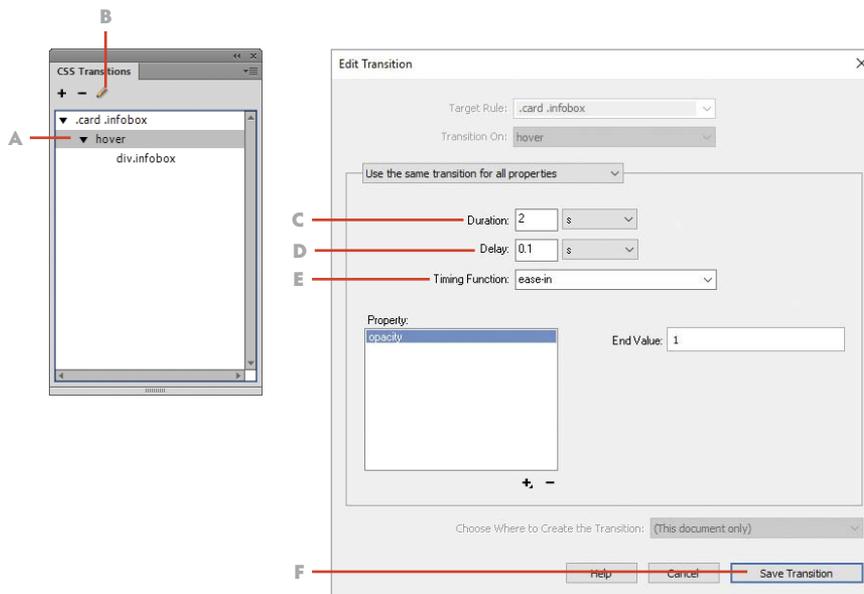
In the event your animation does not appear correctly, the likely culprit is the nesting order in your HTML. Be sure your code matches the block you see here:

```

<div class="card">
<div class="infobox">
<p>Ray the Werewolf</p>
<p>President and CEO</p>
</div>
</div>

```

- 8 In the CSS Transition panel, click the **hover** property, and click the Pencil icon to reopen the Edit Transition dialog, as you see in **Figure 7.19**.



**Figure 7.19** Applying timing, delay, and easing functions in the Edit Transition dialog

- A** Select the property to be edited.
- B** Open the Edit Transition dialog.
- C** Set the amount of time for the transition to occur.
- D** Set the time the animation should wait until the transition occurs.
- E** Preset easing functions
- F** Save the new CSS selector settings.

- 9 Set the Duration value to 2 s, Delay to .1 s, and Timing Function to ease-in. When your settings match those in Figure 7.19, click Save Transition.

Now when you place your mouse over the image, the animation will appear over the course of 2 seconds instead of immediately. You'll also see that the animation is delayed slightly and that as the box appears it gently speeds up.

- 10 Save your file now, and turn back to the CSS that Dreamweaver has created for you.

In the style sheet, you'll find the following code block that defines the transition that is to take place. Because of differences in how browsers implement CSS transitions, Dreamweaver has inserted three separate sets of properties and values to cover all three.

```
.card .infobox {
    background-color: #9B9A9A;
    position: absolute;
    width: 212px;
    height: 257px;
    z-index: 100;
    opacity: 0;
```

```
-webkit-transition: all 2s ease-in .1s;  
-o-transition: all 2s ease-in .1s;  
transition: all 2s ease-in .1s;  
}
```

- 11 Now locate the trigger that has been created using the `hover` pseudo-class. For this new selector that Dreamweaver created, only two sets of properties and values are required.

```
.card .infobox:hover {  
  -webkit-opacity: 1;  
  opacity: 1;  
}
```

You've now created a very simple animation using nothing but CSS transitions, and you should have a good understanding of how these kinds of effects are achieved. Obviously, there is much more that could be done with this animation. You could, for instance, modify the position of an object by setting transitioning from one position value to another. But fundamentally this is how CSS animations are created. You define the containers that hold the elements on the page, apply the initial values that they should have, and then animate against those settings by setting a new value to take effect when something happens on the page. And although you may not use Dreamweaver for creating animations, turning instead to another application, you now know the theory behind CSS3 animations and how the programming is written to take advantage of these functions.



## LEVEL UP: MOBILE APP DEVELOPMENT

One question that hasn't been answered is whether you could build a phone app using Dreamweaver. And the answer is yes. Dreamweaver can be used along with a free service from Adobe, PhoneGap ([phonegap.com/developer](http://phonegap.com/developer)), to turn an entire website into a native application for iPhones, Android phones, and the tablets that use those operating systems.

There are some things you'll need if you want to investigate how the PhoneGap service works. First, you'll need an Adobe ID. (You don't need a paid subscription to Creative Cloud to upload files to create an app.) You'll also need the desktop application for Windows and Macintosh, and you'll need to get the free PhoneGap app for your device on the Apple App Store, Google Play, or Windows Phone Store. With all of those things, you can read through the documentation and discover how you can create simulated phone apps on your computer and then instantly view them on your phone.

Of course, there's much more that would have to be done to package and send a completed app from your computer for distribution. But Adobe has done a great deal to promote the workflow that allows developers to use their design tools, like Dreamweaver, to publish mobile apps of all kinds.

## Your Final Challenge

As you close your look at Dreamweaver and prepare to dig further into the creative process and workflow in the coming chapters, you should be keeping an eye on ways that you can use your new skills to develop websites of your own. Whether you design something purely for yourself, such as a digital portfolio or resume site, or you volunteer to build a website for a nonprofit or religious organization, or you decide to start up your own company and take on real clients, at this point in your transition from web design student to web design professional you should be working with the software as much as you possibly can. As with any skill, it will take hours of practice before you become proficient, and your challenge now is to spend as much time as you can coding, designing, and building websites in Dreamweaver.

## Conclusion

In this chapter, you've dipped your toes into the brave new world of responsive design and learned a great deal about how the interface in Dreamweaver is used to work on sites that feature media queries, jQuery and Bootstrap components, HTML5 video, and CSS3 animations. You've seen how to create new documents in Dreamweaver that already include many of the components required for a responsive design environment, and you should have a better understanding of the work required to design with these new methods.

And even though it seemed at times that you were just noodling around and discovering what you need to do to meet a client's mobile design requirements, you've actually gathered a lot of valuable information that could be used to prepare a time and cost estimate for your client.

*This page intentionally left blank*

# ACA Objectives Covered

DOMAIN OBJECTIVES	CHAPTER	VIDEO
<b>DOMAIN 1.0 Setting Project Requirements</b>		
<b>1.1</b> Identify the purpose, audience, and audience needs for a web authoring project.	<b>Ch 3</b> Project 3.1: Translating Client Project Requirements, 66 <b>Ch 4</b> Project 4.1: Prototyping and Designing with Background Images, 103 <b>Ch 9</b> Who You're Talking FOR and Who You're Talking TO, 9-3 <b>Ch 9</b> Single Voice, Single Message, 9-4 <b>Ch 9</b> Ideal Ivan and Danielle Demographic, 9-6 <b>Ch 9</b> The Language of Web Project Management, 9-28 <b>Ch 9</b> Conclusion, 9-32	<b>3.1</b> Translating Requirements into Structure <b>9.1</b> Introduction to Project Planning <b>9.2</b> Discovering Client Goals <b>9.16</b> Wrapping Up Project Planning
<b>1.2</b> Summarize how designers make decisions about the type of content to include in a web authoring project, including considerations such as copyright, project fit, permissions, and licensing.	<b>Ch 3</b> Project 3.2: Modifying Page Properties, 71 <b>Ch 4</b> Project 4.2: Inserting and Editing Images, 112 <b>Ch 4</b> Project 4.3: Floating Images into Position, 121 <b>Ch 4</b> Project 4.4: Creating Structured Layouts with Images, 125 <b>Ch 4</b> Project 4.5: Styling the Header Area, 133 <b>Ch 9</b> Copyrights and Wrongs, 9-9 <b>Ch 9</b> The Language of Web Project Management, 9-28	<b>3.2</b> Modifying Page Properties <b>4.1</b> Prototyping and Designing with Background Images <b>4.2</b> Inserting and Editing Images <b>4.3</b> Floating Images into Position <b>4.4</b> Creating Structured Layouts with Images <b>4.5</b> Styling the Header Area <b>9.3</b> Finding the Target Audience <b>9.4</b> The Golden Rule for Client Work <b>9.5</b> About Copyright <b>9.6</b> Digital Tools for Tracking Copyright <b>9.7</b> Fair Use and Copyright <b>9.8</b> Licensing Strict and Free
<b>1.3</b> Demonstrate knowledge of project management tasks and responsibilities.	<b>Ch 3</b> Project 3.1: Translating Client Project Requirements, 66 <b>Ch 3</b> Project 3.4: Adding Content Containers and Attributes, 83 <b>Ch 5</b> Project 5.5: Finalizing the Prototype for Production, 166 <b>Ch 7</b> Project 7.1: Exploring Responsive Design with Media Queries, 208 <b>Ch 9</b> Who You're Talking FOR and Who You're Talking TO, 9-3 <b>Ch 9</b> Project Management, 9-16 <b>Ch 9</b> The Language of Web Project Management, 9-28 <b>Ch 9</b> Conclusion, 9-32	<b>3.1</b> Translating Requirements into Structure <b>3.4</b> Adding Content Containers and Attributes <b>5.5</b> Finalizing the Prototype for Production <b>7.2</b> Exploring Responsive Design with Media Queries <b>9.9</b> Project Management Intro <b>9.10</b> Project Management—Understand the Problem <b>9.11</b> Project Management—Think It Through <b>9.12</b> Project Management—Get It in Writing <b>9.13</b> Project Management—Avoid Creep <b>9.14</b> Project Management—Make It So <b>9.16</b> Wrapping Up Project Planning

DOMAIN OBJECTIVES	CHAPTER	VIDEO
1.4 Communicate with others (such as peers and clients) about design plans.	<p><b>Ch 5</b> Project 5.1: Styling the Navigation Sidebar, 141</p> <p><b>Ch 6</b> Project 6.3: Adding Optional Content with Library Items, 189</p> <p><b>Ch 9</b> Feedback Loop, 9-26</p> <p><b>Ch 9</b> The Language of Web Project Management, 9-28</p>	<p><b>5.1</b> Styling the Navigation Sidebar</p> <p><b>6.3</b> Adding Optional Content with Library Items</p> <p><b>9.15</b> The Advantages of Working at a Firm</p>
<b>DOMAIN 2.0 Understanding Web Authoring</b>		
2.1 Demonstrate knowledge of basic web authoring terminology.	<b>Ch 1</b> World Wide What?, 4	<b>1.1</b> How the World Wide Web Works
2.2 Demonstrate knowledge of basic design principles and best practices employed in the web authoring industry, such as usability, accessibility, consistency, and visual hierarchy.	<p><b>Ch 3</b> Project 3.3: Setting Font Formats and Links, 77</p> <p><b>Ch 3</b> Project 3.5: From Blueprint to Assembly: Positioning Page Elements with CSS, 89</p> <p><b>Ch 6</b> Project 6.2: Building Out the Site with Dreamweaver Templates, 183</p> <p><b>Ch 8</b> Creativity Is a Skill, 8-4</p> <p><b>Ch 8</b> The Design Hierarchy, 8-6</p> <p><b>Ch 8</b> The Elements of Art, 8-9</p> <p><b>Ch 8</b> The Elements of Color, 8-24</p>	<p><b>3.3</b> Setting Font Formats and Links</p> <p><b>3.5</b> From Blueprint to Assembly: Positioning Page Elements with CSS</p> <p><b>6.2</b> Building Out the Site with Dreamweaver Templates</p> <p><b>8.1</b> Design School: Introduction</p> <p><b>8.2</b> Design School: Creativity Is a Skill</p> <p><b>8.3</b> Design School: The Design Hierarchy</p> <p>8.4 Design School: The Elements of Art</p> <p><b>8.5</b> Design School: The Element of Space</p> <p><b>8.6</b> Design School: The Element of Line</p> <p><b>8.7</b> Design School: The Element of Shape</p> <p><b>8.8</b> Design School: The Element of Form</p> <p><b>8.9</b> Design School: The Elements of Texture and Pattern</p> <p><b>8.10</b> Design School: The Element of Value</p> <p><b>8.11</b> Design School: The Element of Color</p>

*continues on next page*

continued from previous page

DOMAIN OBJECTIVES	CHAPTER	VIDEO
<p><b>2.3</b> Demonstrate knowledge of Hypertext Markup Language (HTML) and the behavior of basic HTML tags.</p>	<p><b>Ch 1</b> Serving Up the Goodies, 7  <b>Ch 1</b> The Structure of Websites, 15  <b>Ch 1</b> Project 1.2: Under the Hood—Examining Code in a Web Page, 24  <b>Ch 1</b> Challenge! Customize the Starter Page and Make It Your Own, 32  <b>Ch 3</b> Project 3.3: Setting Font Formats and Links, 77  <b>Ch 7</b> Understanding HTML5, CSS3, and Responsive Design, 205  <b>Ch 7</b> Project 7.1: Exploring Responsive Design with Media Queries, 208  <b>Ch 7</b> Project 7.2: Exploring Responsive Design with jQuery and Bootstrap, 215  <b>Ch 7</b> Project 7.3: Inserting HTML5 Video, 225</p>	<p><b>1.2</b> Tag: You’re It  <b>1.3</b> Let’s Get Stylish  <b>1.4</b> Understanding the Structure of Websites  <b>1.6</b> Under the Hood—Examining Code in a Web Page  <b>1.7</b> Challenge! Customize the Starter Page and Make It Your Own  <b>3.3</b> Setting Font Formats and Links  <b>7.1</b> Exploring Responsive Design with HTML5  <b>7.2</b> Exploring Responsive Design with Media Queries  <b>7.3</b> Exploring Responsive Design with jQuery and Bootstrap  <b>7.4</b> Inserting HTML5 Video</p>
<p><b>2.4</b> Demonstrate knowledge of Cascading Style Sheets (CSS) and the behavior of basic CSS tags.</p>	<p><b>Ch 1</b> Let’s Get Stylish, 12  <b>Ch 2</b> Using the CSS Designer Panel, 54  <b>Ch 3</b> Project 3.2: Modifying Page Properties, 71  <b>Ch 4</b> Project 4.1: Prototyping and Designing with Background Images, 103  <b>Ch 4</b> Project 4.2: Inserting and Editing Images, 112  <b>Ch 5</b> Project 5.1: Styling the Navigation Sidebar, 141  <b>Ch 5</b> Project 5.2: Styling Navigation Elements, 147  <b>Ch 5</b> Project 5.3: Styling Structured Data, 147  <b>Ch 5</b> Project 5.4: Building and Designing Interactive Forms, 160  <b>Ch 6</b> Reviewing CSS Syntax, 175  <b>Ch 6</b> Project 6.1: Putting the Style Sheet into Production, 177  <b>Ch 6</b> Project 6.3: Adding Optional Content with Library Items, 189</p>	<p><b>2.5</b> The CSS Designer Panel  <b>3.2</b> Modifying Page Properties  <b>4.1</b> Prototyping and Designing with Background Images  <b>4.2</b> Inserting and Editing Images  <b>5.1</b> Styling the Navigation Sidebar  <b>5.2</b> Styling the Navigation Elements  <b>5.3</b> Styling Structured Data  <b>5.4</b> Building and Designing Interactive Forms  <b>6.1</b> Putting the Style Sheet into Production  <b>6.3</b> Adding Optional Content with Library Items</p>

DOMAIN OBJECTIVES	CHAPTER	VIDEO
<b>2.5</b> Demonstrate knowledge of how to author web content that works equally well in various operating systems, browser versions/configurations, and devices.	<b>Ch 7</b> Project 7.1: Exploring Responsive Design with Media Queries, 208 <b>Ch 7</b> Project 7.2: Exploring Responsive Design with jQuery and Bootstrap, 215 <b>Ch 7</b> Project 7.3: Inserting HTML5 Video, 225	<b>7.2</b> Exploring Responsive Design with Media Queries <b>7.3</b> Exploring Responsive Design with jQuery and Bootstrap <b>7.4</b> Inserting HTML5 Video
<b>2.6</b> Demonstrate knowledge of digital images, text, interactive elements, animation, and video and audio content as they pertain to web authoring.	<b>Ch 4</b> Image File Formats for the Web, 101 <b>Ch 4</b> Project 4.2: Inserting and Editing Images, 112 <b>Ch 4</b> Project 4.3: Floating Images into Position, 121 <b>Ch 4</b> Project 4.4: Creating Structured Layouts with Images, 125 <b>Ch 4</b> Project 4.5: Styling the Header Area, 133 <b>Ch 7</b> Project 7.4: Exploring CSS3 Animation, 229 <b>Ch 8</b> The Element of Type, 8-29	<b>4.2</b> Inserting and Editing Images <b>4.3</b> Floating Images into Position <b>4.4</b> Creating Structured Layouts with Images <b>4.5</b> Styling the Header Area <b>7.5</b> Exploring CSS3 Animation <b>8.12</b> Design School: The Element of Type <b>8.13</b> Design School: The Principles of Design <b>8.14</b> Design School: The Principle of Emphasis <b>8.15</b> Design School: The Principle of Contrast <b>8.16</b> Design School: The Principle of Unity <b>8.17</b> Design School: The Principle of Variety <b>8.18</b> Design School: The Principle of Balance <b>8.19</b> Design School: The Principle of Proportion or Scale <b>8.20</b> Design School: The Principles of Repetition and Pattern <b>8.21</b> Design School: The Principles of Movement and Rhythm <b>8.22</b> Design School: Wrapping Up Design School
<b>DOMAIN 3.0 Understanding Adobe Dreamweaver CC</b>		
<b>3.1</b> Identify elements of the Adobe Dreamweaver CC interface and demonstrate knowledge of their functions.	<b>Ch 2</b> Dreamweaver Workspace Overview, 40 <b>Ch 2</b> The Document Window, 42 <b>Ch 2</b> The Property Inspector, 48 <b>Ch 2</b> Panels and Panel Groups, 52 <b>Ch 2</b> Using the Assets Panel, 57 <b>Ch 2</b> Coding Tools in Dreamweaver CC, 59 <b>Ch 6</b> Project 6.2: Building Out the Site with Dreamweaver Templates, 183	<b>2.1</b> A Quick Tour of the Dreamweaver Workspace <b>2.2</b> The Dreamweaver Document Window <b>2.3</b> The Property Inspector <b>2.4</b> Panel Groups in Dreamweaver <b>2.7</b> The Assets Panel <b>2.8</b> Coding Tools in Dreamweaver CC <b>6.2</b> Building Out the Site with Dreamweaver Templates

*continues on next page*

continued from previous page

DOMAIN OBJECTIVES	CHAPTER	VIDEO
<b>3.2</b> Identify and define the functions of commonly used panels, including the Insert panel, Property inspector, Assets panel, and Files panel.	<b>Ch 2</b> Dreamweaver Workspace Overview, 40 <b>Ch 2</b> Dreamweaver Workspace Overview, 40 <b>Ch 2</b> The Property Inspector, 48 <b>Ch 2</b> Panels and Panel Groups, 52 <b>Ch 2</b> Using the CSS Designer Panel, 54 <b>Ch 2</b> Using the Files Panel, 56 <b>Ch 2</b> Using the Assets Panel, 57 <b>Ch 3</b> Project 3.4: Adding Content Containers and Attributes, 83 <b>Ch 3</b> Project 3.5: From Blueprint to Assembly: Positioning Page Elements with CSS, 89 <b>Ch 6</b> Project 6.3: Adding Optional Content with Library Items, 189	<b>2.1</b> A Quick Tour of the Dreamweaver Workspace <b>2.3</b> The Property Inspector <b>2.4</b> Panel Groups in Dreamweaver <b>2.5</b> The CSS Designer Panel <b>2.6</b> The Files Panel <b>2.7</b> The Assets Panel <b>3.4</b> Adding Content Containers and Attributes <b>3.5</b> From Blueprint to Assembly: Positioning Page Elements with CSS <b>6.3</b> Adding Optional Content with Library Items
<b>3.3</b> Navigate, organize, and customize the workspace.	<b>Ch 2</b> The Document Window, 42 <b>Ch 2</b> Panels and Panel Groups, 52	<b>2.2</b> The Dreamweaver Document Window <b>2.4</b> Panel Groups in Dreamweaver
<b>DOMAIN 4.0 Web Authoring Using Adobe Dreamweaver CC</b>		
<b>4.1</b> Define a new Dreamweaver site.	<b>Ch 1</b> Define a New Dreamweaver Site, 20	<b>1.5</b> Defining a Dreamweaver Website <b>3.1</b> Translating Requirements into Structure
<b>4.2</b> Create a web page and define the page layout using HTML, CSS, and div tags.	<b>Ch 1</b> Project 1.2, Under the Hood—Examining Code in a Web Page, 24 <b>Ch 1</b> Challenge! Customize the Starter Page and Make It Your Own, 32 <b>Ch 3</b> Project 3.1: Translating Client Project Requirements, 66 <b>Ch 3</b> Project 3.2: Modifying Page Properties, 71 <b>Ch 4</b> Project 4.1: Prototyping and Designing with Background Images, 103 <b>Ch 4</b> Project 4.2: Inserting and Editing Images, 112 <b>Ch 5</b> Project 5.2: Styling Navigation Elements, 147 <b>Ch 7</b> Project 7.1: Exploring Responsive Design with Media Queries, 208 <b>Ch 7</b> Project 7.2: Exploring Responsive Design with jQuery and Bootstrap, 215 <b>Ch 7</b> Project 7.3: Inserting HTML5 Video, 225	<b>1.6</b> Under the Hood—Examining Code in a Web Page <b>1.7</b> Challenge! Customize the Starter Page and Make It Your Own <b>3.1</b> Translating Requirements into Structure <b>3.2</b> Modifying Page Properties <b>4.1</b> Prototyping and Designing with Background Images <b>4.2</b> Inserting and Editing Images <b>5.2</b> Styling the Navigation Elements <b>7.2</b> Exploring Responsive Design with Media Queries <b>7.3</b> Exploring Responsive Design with jQuery and Bootstrap <b>7.4</b> Inserting HTML5 Video

DOMAIN OBJECTIVES	CHAPTER	VIDEO
4.3 Add and modify text.	<b>Ch 3</b> Project 3.3: Setting Font Formats and Links, 77	<b>3.3</b> Setting Font Formats and Links
4.4 Insert and manipulate images, video, sound, interactive elements, and animation.	<b>Ch 5</b> Project 5.1: Styling the Navigation Sidebar, 141 <b>Ch 6</b> Project 6.1: Putting the Style Sheet into Production, 177 <b>Ch 6</b> Project 6.2: Building Out the Site with Dreamweaver Templates, 183 <b>Ch 6</b> Project 6.3: Adding Optional Content with Library Items, 189 <b>Ch 7</b> Understanding HTML5, CSS3, and Responsive Design, 205 <b>Ch 7</b> Project 7.2: Exploring Responsive Design with jQuery and Bootstrap, 215 <b>Ch 7</b> Project 7.3: Inserting HTML5 Video, 225 <b>Ch 7</b> Project 7.4: Exploring CSS3 Animation, 229	<b>5.1</b> Styling the Navigation Sidebar <b>6.1</b> Putting the Style Sheet into Production <b>6.2</b> Building Out the Site with Dreamweaver Templates <b>6.3</b> Adding Optional Content with Library Items <b>7.1</b> Exploring Responsive Design with HTML5 <b>7.3</b> Exploring Responsive Design with jQuery and Bootstrap <b>7.4</b> Inserting HTML5 Video <b>7.5</b> Exploring CSS3 Animation
4.5 Link web content, using hyperlinks, email links, and named anchors.	<b>Ch 5</b> Project 5.1: Styling the Navigation Sidebar, 141 <b>Ch 5</b> Project 5.2: Styling Navigation Elements, 147 <b>Ch 6</b> Project 6.1: Putting the Style Sheet into Production, 177 <b>Ch 6</b> Project 6.2: Building Out the Site with Dreamweaver Templates, 183 <b>Ch 6</b> Project 6.3: Adding Optional Content with Library Items, 189	<b>5.1</b> Styling the Navigation Sidebar <b>5.2</b> Styling the Navigation Elements <b>6.1</b> Putting the Style Sheet into Production <b>6.2</b> Building Out the Site with Dreamweaver Templates <b>6.3</b> Adding Optional Content with Library Items
4.6 Create and manage special content types in a web authoring project, including tables and forms.	<b>Ch 5</b> Project 5.3: Styling Structured Data, 147 <b>Ch 5</b> Project 5.4: Building and Designing Interactive Forms, 160	<b>5.3</b> Styling Structured Data <b>5.4</b> Building and Designing Interactive Forms

*continues on next page*

continued from previous page

DOMAIN OBJECTIVES	CHAPTER	VIDEO
<b>DOMAIN 5.0 Testing, Publishing, and Maintaining Web Content Using Adobe Dreamweaver CC</b>		
<b>5.1</b> Conduct technical and usability tests.	<b>Ch 3</b> Project 3.4: Adding Content Containers and Attributes, 83	<b>3.4</b> Adding Content Containers and Attributes
	<b>Ch 3</b> Project 3.5: From Blueprint to Assembly: Positioning Page Elements with CSS, 89	<b>3.5</b> From Blueprint to Assembly: Positioning Page Elements with CSS
	<b>Ch 5</b> Project 5.5: Finalizing the Prototype for Production, 166	<b>5.5</b> Finalizing the Prototype for Production
	<b>Ch 6</b> Project 6.4: Publishing and Updating a Dreamweaver Site, 195	<b>6.4</b> Publishing and Updating a Dreamweaver Site
	<b>Ch 7</b> Project 7.1: Exploring Responsive Design with Media Queries, 208	<b>7.2</b> Exploring Responsive Design with Media Queries
<b>5.2</b> Manage assets, links, and files for a site.	<b>Ch 2</b> Using the Files Panel, 56	<b>2.6</b> The Files Panel
	<b>Ch 3</b> Project 3.1: Translating Client Project Requirements, 66	<b>3.1</b> Translating Requirements into Structure
	<b>Ch 6</b> Project 6.4: Publishing and Updating a Dreamweaver Site, 195	<b>6.4</b> Publishing and Updating a Dreamweaver Site
	<b>Ch 9</b> The Language of Web Project Management, 9-28	
<b>5.3</b> Publish and update site files to a remote server.	<b>Ch 6</b> Project 6.4: Publishing and Updating a Dreamweaver Site, 195	<b>6.4</b> Publishing and Updating a Dreamweaver Site

# Glossary

**absolute links** Links to a website outside the current site. For example, *http://adobe.com*.

**absolute positioning** Used to place objects on the page at precise X and Y coordinates. The X coordinates measure an element's position from the left side of the page, whereas Y coordinates measure an element's location from the top of the page.

**accessibility** The practice of removing barriers that prevent access to website interaction and information by people with disabilities.

**action** A response triggered by an event initiated by the viewer of a web page, such as rolling the mouse pointer across an image.

**active** A hyperlink that is currently selected by a viewer of a web page.

**additive color** Created by combining light.

**address** The unique location of an individual web page on the Internet. *See* URL.

**alignment** Indicates how the lines are aligned on the right and left edges, such as left, centered, and right.

**all caps** Uses only uppercase letterforms for each letter.

**anchors** Navigational elements of a web page that allow the viewer to jump to a particular point on the page.

**animated GIF** An image file that plays a series of frames, creating the illusion of movement.

**alternate text** Additional information that is displayed when an image is loading or when the viewer is using a screen-reading device.

**analogous** Colors that are side by side on the color wheel. They create gentle and relaxing color schemes.

**analytics** The process of collecting and analyzing data about a website's visitors for the purposes of improving site performance.

**asymmetrical** Achieves balance with different elements with different weights on each side (or the top and bottom) of an image.

**aspect ratio** The proportional relationship between the width and height of an image. Written as 16:9, 4:3, and so on.

**attribute** Named properties that assign additional information to an HTML tag.

**attribution** Written acknowledgment provided with the name of the original copy-right holder of the work. Creative Commons and other licenses feature different kinds of attribution requirements.

**balance** Evenly distributed, but not necessarily centered or mirrored.

**bandwidth** The rate of data transfer, bit rate, or throughput, measured in bits per second (bps).

**baseline** An imaginary line used to organize text along a horizontal plane.

**bitmaps** Images created on a computer screen using discrete bits of individual colors.

**blackletter fonts** Also known as old English, Gothic, or Textura. Feature an overly ornate style. Convey a feeling of rich and sophisticated gravitas.

**Bootstrap** A popular open source library of HTML, CSS, and JavaScript used to build web page layouts and components.

**box model** The CSS box model refers to the box that wraps around HTML elements. Each box consists of margins, borders, padding, and the actual content.

**browser** A computer program used to access and view web pages stored on the World Wide Web.

**Cascading Style Sheets (CSS)** Coded instructions to a web browser describing how elements of HTML are to be displayed.

**cast shadow** The shadow cast on the ground and on any objects that are in the shadow of the form. Shadows fade as they get farther from the form casting the shadow.

**chaotic lines** Look like scribbles and feel unpredictable and frantic. Convey a sense of urgency, fear, or explosive energy.

**check in** Marking a file as available for others to work on.

**check out** Locking a file to prevent others from working on it.

**child selector** A CSS attribute that immediately follows a “parent” selector, as in `tr:nth-child(even)`.

**class selectors** CSS styling rules targeted at class names assigned to elements in the page. May be used multiple times in a document.

**client** The software and computer device used to access information on a remote server.

**client-side programming** Programming that runs on a client computer and that is written in a language, such as JavaScript, that can be executed by the browser.

**client survey (interview)** The process used at the beginning of project planning to determine the customer's objectives for a website and the requirements and resources available.

**Code Navigator** The Dreamweaver pop-up display that allows you to view all the styling properties applied to an object on the page.

**color** The perceived hue, lightness, and saturation of an object or light.

**color harmonies** Color rules that are named for their relative locations on the color wheel.

**complementary** Colors that are opposite each other on the color wheel. They are high in contrast and vibrant.

**contrast** Creates visual interest and a focal point in a composition. It is what draws the eye to the focal point.

**Creative Commons** Ways that artists can release their works for limited use and still choose the way the works are used and shared: Public Domain, Attribution, ShareAlike, NoDerivs, and NonCommercial.

**CSS Design panel** The Dreamweaver panel that displays the source, media queries, selectors, and properties of a CSS style sheet.

**CSS box model** *See* box model.

**curved (line)** Expresses fluidity, beauty, and grace.

**database** Structured data stored on a remote server that allows for dynamic data on a client device.

**declaration block** The styling properties to be applied to a particular selector as determined by the CSS rule.

**decorative fonts** Also known as ornamental, novelty, or display fonts. They don't fall into any of the other categories of fonts. Convey a specific feeling.

**deliverables** A predetermined list of items that will be delivered to the customer.

**deprecated** A technique or code designated as obsolete.

**descendent selectors** The descendent selector matches all elements that are descendants of a specified element.

**design comp** A digital image that shows all the design details of a project. Typically used for client reviews.

**design elements** The building blocks of art defined by artists to provide a framework for creating art.

**design principles** The essential rules or assembly instructions for art.

**diagonal (lines)** Lines traveling neither on a vertical nor a horizontal path. Express growth or decline and imply movement or change.

**dingbat fonts** Also known as wingdings. They are a collection of objects and shapes instead of letters.

**direction** A common way to describe lines, such as vertical, horizontal, diagonal.

**Document Object Model (DOM)** Defines the logical structure of the HTML in a web page.

**Document toolbar** The Dreamweaver UI element that displays buttons to toggle between Live, Design, and Code views and access other inspectors for visualizing a page.

**document tree** A method for visualizing the relationship between computer files and folders.

**Document type <!doctype>** The first element of an HTML document. Used to declare to a web browser the version of HTML the page is written in.

**domain name server** An Internet service that translates domain names to IP addresses.

**domain name** The portion of the Internet registered to a particular company, organization, or individual, such as microsoft.com, adobe.com, and google.com.

**dynamic website** A website that uses a database and web programming, such as PHP, to provide live interactions with client computers.

**element selectors** CSS rules that target elements of the page by the tag used to define them.

**element styles** *See* element selectors.

**elements of art** The building blocks of creative works. They are the “nouns” of design, such as space, line, shape, form, texture, value, color, and type.

**emphasis** Describes the focal point to which the eye is naturally and initially drawn in a design.

**encoding** The process of converting a digital file from one format to another. Usually used when referring to the preparation of digital video files.

**external style sheets** A separate document containing CSS styling rules that is linked to an HTML page.

**external CSS file** *See* external style sheets.

**fair use** A set of rules that specify how and when copyrighted material can be used and that make sure copyright protection doesn't come at the cost of creativity and freedom.

**feedback loop** A system set up to continually encourage and require input and approval on a project's direction.

**file extension** The three- or four-letter designator that follows a filename and that specifies which program can use the file, such as .png, .jpg, and .html.

**File Transfer Protocol (FTP)** The networking standard that allows files to be transferred from a client to a remote server.

**file weight** The amount of data that must be downloaded from a web server in order to display a page. This includes the HTML, images, scripts, and all other assets required for the page to display.

**Files panel** The Dreamweaver panel that allows you to view, open, edit, and perform server interactions with the folders and files in a website.

**flow** A category related to the energy conveyed by lines and shapes.

**focal point** What the design is all about. The call to action or the primary message you are trying to get across.

**font family** A CSS styling rule that allows you to list a set of fonts to be used when styling text. Allows for a "fallback" if one or more fonts are not present on the client device.

**fonts** The whole collection of a typeface in each of its sizes and styles.

**font weight** CSS styling property that determines how thick or thin the text characters should be.

**form** Describes three-dimensional objects, such as spheres, cubes, and pyramids.

**framework** A set of computer rules, procedures, and standardized programming methods.

**geometric (lines)** Tend to be straight and have sharp angles. Look manmade and intentional. Communicate strength, power, and precision.

**geometric shapes** Predictable and consistent shapes, such as circles, squares, triangles, and stars. They are rarely found in nature and convey mechanical and manufactured impressions.

**get** The Dreamweaver term for copying a file from a remote server to a local computer.

**glyph** Each character of a font, whether it is a letter, number, symbol, or swash.

**glyphicons** Fonts that display as graphical elements when assigned as a CSS class to a `<span>` tag. Included as part of the Bootstrap CSS library.

**group selectors** A CSS selector rule that is written to assign the same properties to more than one tag, as in `h1`, `p`.

**hand-drawn lines** Appear as though created using traditional techniques, such as paints, charcoal, or chalk.

**handwritten fonts** Also known as hand fonts, they simulate handwriting.

**head (head of the document)** The area of a web page enclosed by the `<head></head>` tags. Used for storing instructions to the browser that are not visible in the page, such as metadata and links to external files.

**hexadecimal** The six-digit code used to define colors on a web page, such as `#FOFOFO`.

**highlight** The area of a form that is directly facing the light and that appears lightest.

**horizontal** Moving from left to right; for example, the horizontal line in an “H.” Expresses calmness and balance.

**horizontal scale** Describes the function of stretching letters and distorting the typeface geometry.

**HTML5** The latest version of the web programming language used to mark up and display documents on the World Wide Web. HTML5 contains new specifications for embedding page divisions, graphics, audio, video, and interactive documents.

**hyperlink** A word or picture in a web page that allows the viewer of the page to go to a different location when it is clicked or tapped.

**Hypertext Markup Language (HTML)** The web programming language used to mark up and display documents on the World Wide Web.

**Hypertext Transfer Protocol (HTTP)** The underlying set of communication rules, or protocols, that allow web servers to transfer information to and from client devices on the World Wide Web.

**hyphenation** Determines if and when words should be split with hyphens when wrapping to the next line.

**ID selectors** CSS styling rules targeted at an element that has a named identifier (ID).

**ideographs (ideograms)** Images that represent an idea, such as a heart representing love.

**image maps** An image in which some portions of the graphic contain a hotspot that contains a hyperlink.

**implied lines** Lines that don't really exist but are implied by shapes, such as dotted or dashed lines, people waiting in lines, or the margins of a block of text.

**indent** Settings that determine how far an entire paragraph is indented from the rest of the text on each side or in just its first line.

**information architecture** The methods used to structure and organize computer data.

**inline styles** A type of CSS styling that is contained directly inside a tag, such as `<span style="color: #AC1316">`.

**internal style sheet** CSS styles that are written inside the `<head>` of a document used for styling on that single page.

**Internet Protocol (IP)** The coding rules that govern how data is sent over the Internet or other networks.

**IP address** A unique string of numbers separated by periods that identifies each computer attached to the Internet.

**iterations** New versions of a design that successively become closer to the desired result.

**iterative work** Work that is shared as it is completed, allowing the customer to chime in with comments while it is still easy to make a change.

**JavaScript** A scripting language for adding interactive elements to a web page.

**JavaScript library** A set of pre-programmed JavaScript functions for adding interactions, animations, and other common tasks to a web page.

**jQuery Mobile** The JavaScript library that uses CSS selectors to access and manipulate HTML elements on a web page.

**justified** Aligns text to a straight edge on both the right and left edges of a paragraph.

**kerning** The space between specific letter pairs.

**leading** The amount of space between the baselines of two lines of text.

**library item** An element within a site that is available for reuse and automated updating using the Dreamweaver Assets panel.

**licensing** A way to legally use copyrighted material for a certain time and in a certain way, usually associated with paying a fee established by the copyright holder.

**ligatures** Special characters used to represent letter combinations, such as “fi.”

**light source** The perceived location of the lighting in relation to the form.

**line** A mark with a beginning and an end point.

**linting** The program that examines computer code for potential errors as it is being written.

**Live view** The Dreamweaver view of a document that mimics how the page will display when viewed in a web browser or mobile device.

**media queries** CSS styling rules that allow a page to adapt to different screen sizes and orientations. Also used for creating print versions of web pages.

**metadata** Information that is included in a document but is hidden, such as copyright, lens information, location via GPS, camera settings, and more.

**minified** Computer code that has been optimized by removing all unnecessary characters and spacing without changing its functionality.

**model releases** The permission that is required when a person’s face is identifiable in a photo and the image will be used to promote something—whether it’s a product or an idea.

**monochromatic** Different shades and tints of the same color. Communicates a relaxed and peaceful feeling.

**monospaced fonts** Fixed-width or non-proportional fonts that use the same amount of horizontal space for each letter.

**movement** Visual movement within an image, such as the natural tracking of the eye across an image as the eye moves from focal point to focal point.

**negative space** Blank areas in a design. Also known as white space.

**NoDerivs (ND)** Creative Commons licensing. Requires that you not change material when you incorporate it into your own work. It can be used freely, but you must pass it along without change.

**NonCommercial (NC)** Creative Commons licensing. Means you can use work in your own creative work as long as you don't charge for it.

**object shadow** The area of the form that is facing away from the light source and appears darkest.

**open source** Source code that is freely available for modification or enhancement by anyone.

**optimization** The process of choosing a file type and compression settings to provide the smallest file sizes possible while maintaining quality.

**organic lines** Lines that are usually irregular and imperfect. Found in nature.

**organic shapes** Are random or generated by something natural. They are usually asymmetric and convey natural, homemade, or relaxed feelings.

**page title** The name of the web page that is seen in the title bar of the browser and in web search results.

**paragraph settings** Affect an entire paragraph rather than selected words. These settings include alignment, hyphenation, and so on.

**paragraph spacing** Similar to leading, but applies to an entire paragraph instead of lines of type within them. Also includes the spacing above or below paragraphs.

**path** A sequence of symbols and letters that indicates the name and location of a computer file, as in `C:\MyDocuments\MyFile.doc`.

**pattern** A repetitive sequence of different colors, shapes, or values.

**pictograph (pictogram)** Graphic symbol that represents something in the real world. Computer icons are pictographs that suggest the function they represent, such as a trash can icon to delete a file.

**primary colors** Red, blue, and green. These can be combined to create every other color in the visible spectrum.

**project creep** Unplanned changes that increase the amount of work, or scope, that a project requires. When the project loses focus and spins out of control, eating up more and more time and effort.

**project deadlines** Dictates when work needs to be completed.

**project scope** Outlines the amount and type of work to be completed.

**Property inspector** The contextual Dreamweaver panel that is located below the Document window; used to display and edit the properties of a selected element on the page.

**proportion (scale)** Describes the relative size and scale of elements.

**protocol** Computer programming and communication rules.

**pseudo-class selectors** Keywords added to selectors; specify a special state of the element to be selected when an action occurs on the page, such as a mouse pointer rolling over a link.

**public domain** Creative Commons licensing. When copyright is expired or released and no longer applies to the content or when an artist releases their work. It can be used without worrying about infringement.

**put** The Dreamweaver term for copying a file from a local computer to a remote server.

**radial** Circular type of balance that radiates from the center instead of the middle of a design.

**reflected highlight** Area of a form that is lit by reflections from the ground or other objects in a scene.

**related files** Files that are linked to a document. Displayed in Dreamweaver's Related Files toolbar.

**repetition** Repeating an element in a design.

**representative shapes** Shapes used to represent information. They are helpful in communicating with multicultural and multilingual audiences.

**RGB** Red, green, blue color values used to write color specifications into CSS styling rules.

**rhythm** Creative and expressive, rather than a consistent pattern or repetition in a design.

**root folder** The highest folder in the structure of a website. All the files and folders for a site are located in this folder. Also called the site folder.

**round trip** The process by which an image can be associated with a source application, such as Photoshop, so that it can be opened and edited from within Dreamweaver.

**rule of thirds** A technique for laying out the space of your page to provide a focal point. Two vertical and two horizontal lines evenly divide the space into nine equal boxes, as in a tic-tac-toe board.

**sans serif fonts** Text without serifs. Often used for headlines and titles for their strong, stable, modern feel.

**scale** See proportion.

**script fonts** Mimic handwriting. They convey a feeling of beauty, grace, or feminine dignity.

**secondary colors** Created when you combine primary colors.

**serif fonts** Associated with fonts created by typewriters. They convey tradition, intelligence, and class.

**server** A computer with special software that allows data to be transferred back and forth to a client computer or device.

**shape** An area enclosed or defined by an outline, such as circles, squares, triangles, or even clouds.

**ShareAlike (SA)** Creative Commons licensing. Allows you to use an item (design) in any way you want as long as your creation is shared under the same license as the original work.

**sitemap** A model of a website's content that shows the logical structure of the folders and files within the site. Also used as a navigational guide for viewers as well as a roadmap for search engines to discover and index.

**sketches** Representative drawings of how to lay out a document or web page. These are sometimes one of the deliverables of a project.

**slab serif fonts** Squared-off versions of a typical serif font. Also known as Egyptian, block serif, or square serif. Convey a machine-built feel.

**small caps** Uses only uppercase letterforms for each letter and appears in a smaller size.

**space** The canvas, or working area. Its dimensions are determined by the resolution of the page you are creating.

**specifications** Detailed written goals and limits for a project. These are sometimes one of the deliverables of a project.

**Split view** The Dreamweaver view that splits the Document window into two panes where the design and code for the page may be viewed at the same time.

**static website** A website that does not have a connection to a database but relies on the manual updating of content by a web designer.

**stock photos** Images for which the author retains copyright but for which a license for use is available.

**storyboard** A sketch of the initial design for a web page; includes the basic layout blocks of the page as well as specific information about colors, fonts, and other styling properties.

**style (line)** An effect applied to a line, such as varying width, hand-drawn, and implied.

**subtractive color** Created by subtracting light.

**Subversion** An open source version control system that tracks revisions and makes backup copies of files in a shared work environment.

**swashes** Special characters with flowing and elegant endings for the ascenders and descenders.

**symmetrical** Occurs when you can divide an image along its middle, and the left side of the image is a mirror image of the right (or the top reflects the bottom). Conveys an intentional, formal, and mechanical feeling.

**syntax** Determines the rules to be followed when writing code. The spelling and grammar of a programming language.

**tags** The instructions written by the programmer (or generated by Dreamweaver or another web authoring program) that tell the browser what to do.

**templates** HTML pages that contain special code that allows certain portions of the page to remain the same on every page that uses the template, while other portions are available for editing.

**tertiary colors** Created by mixing primary and secondary colors.

**texture** Describes the actual tactile texture in real objects or the appearance of texture in a two-dimensional image.

**tracking** The overall space between all the letters in a block of text. It allows you to compress or expand the space between the letters as a whole rather than just between specific pairs, as you do with kerning.

**typeface** Specific letterform set, such as Helvetica, Arial, Garamond, and so on. It is the “look” of letters.

**type size** A font’s height from the highest ascender to the lowest descender.

**unity** Also known as harmony and sharing similar traits. Low contrast. Things that go together should look like they belong together. The opposite of variety.

**Universal Resource Locator (URL)** The unique web address of pages and files located on the World Wide Web.

**usability** Describes how easy it is for users of a website to find information and interact with the site in a positive way.

**value** Describes the lightness or darkness of an object. Together with color, value represents the visible spectrum, such as a gradient.

**variety** High contrast. The opposite of unity.

**varying width lines** Expresses flow and grace.

**vertical** Moving from top to bottom. Vertical lines tend to express power and elevation.

**vertical scale** Describes the function of stretching letters and distorting the typeface geometry.

**viewport** The visible area of a web page when viewed on a computer, tablet, or mobile device.

**web font** A library of fonts stored on a remote server that may be displayed in a web page by using the `@font-face` rule. Browsers download the font from its hosted location, then display the font as specified in the CSS styling rules.

**website** A collection of web pages, files, and links all associated with a unique name.

**weight (line)** The thickness of a line.

**wireframe** A schematic sketch of a website that shows how pages are connected and how users are expected to access information on the site.

**World Wide Web (web)** The portion of the Internet that supports HTML documents that features links to other documents, as well as graphics, audio, and video files.

**World Wide Web Consortium (W3C)** The international standards organization for the World Wide Web. In addition to developing standards for web documents, it engages in outreach and education and develops software.

**workspace layouts** The arrangement of panels and windows in the Dreamweaver work environment.

*This page intentionally left blank*

# Index

## SYMBOL

/ (forward slash) character, 8

## A

<a> tag, 11, 27–28  
absolute links, 247  
absolute positioning, 247  
accessibility  
    defined, 247  
    errors, 170–171  
active hyperlink, 247. *See also* hyperlinks  
additive color, 247. *See also* colors  
address, 247  
advertisement book, adding, 166–167  
alien planet, first steps on, 5  
alignment  
    defined, 247  
    of elements with guides, 130–131  
all caps, 247  
alt attribute, 30, 51, 124, 170–171, 247  
analogous, 247  
anchors, 247  
animated GIF, 247  
animation, 229–236  
application bar, 41–42  
argyle-bg.png file, 110  
art, elements of, 250  
<article> tag, 11  
aspect ratio, 114, 248  
Assets panel  
inserting images with, 113  
    using, 57–59, 145  
    asymmetrical, 247  
attributes  
    accessing, 30  
    adding, 11, 27, 83–88  
    defined, 248

## B

Back/Forward buttons, 46  
background color. *See also* colors  
    choosing from images, 151  
    setting, 91  
background images, using, 108–111. *See also* images  
backing up files, 106

balance, 248  
bandwidth, 248  
baseline, 248  
bitmaps, 102, 248  
blackletter fonts, 248  
<blockquote> tag, 49  
<body> tag, 9, 27, 76  
Bold settings, accessing with Property inspector, 49  
Bootstrap, 207. *See also* responsive design  
    components in Dreamweaver, 217–219  
    explained, 248  
    glyphicons, 218–219  
    media queries, 217  
    modifying properties, 219–220  
    overriding settings, 221–222  
    responsive design page, 216–217  
boss, thinking like, 66–68  
box model, 45, 90, 248  
Brightness and Contrast tool, 117  
Browse to File button, 35  
browser  
    capabilities, 14, 26  
    defined, 4, 248  
    previewing files in, 110  
    previewing pages in, 171  
    starting up page inspectors in, 224  
browsing to folders, 22, 40

## C

“Capes & Babes” online comic, 64  
card container, styling for CSS3 animation, 231  
cartooning, 64  
cast shadow, 248  
chaotic lines, 248  
character container, preparing content for, 126  
character content, inserting <div> for, 126–129  
check in and check out, 248  
Check In/Check Out button, 57  
child selectors, 158, 176, 248  
Chris the Cartoonist project. *See also* project files  
    client feedback and requirements, 103–105  
    described, 64  
    downloading files, 65  
    font formats, 77–81  
    home page, 69  
    links, 81–82

- page properties, 71–76
  - problem definition, 66–68
  - site setup, 65
  - site structure, 68–70
- class attribute, 11, 28
- class rules, 49
- class selectors, 176, 249
  - applying, 123
  - creating and applying, 121–122
  - duplicating, 123
- classes
  - applying to objects, 219–220
  - versus IDs, 87
- cleaning up code and design 169–170
- clear values, applying, 123–124
- client
  - defined, 249
  - versus server, 6
- client feedback and requirements, 103–105
- client project requirements, translating, 66–70
- client resources, 72
- client survey (interview), 249
- client-side programming, 249
- code. *See also* writing in code
  - cleaning up, 169–171
  - examining in web pages, 24–31
- code collapse, 60
- code coloring, 30
- code hints, 59
- Code Navigator, 60, 88, 249
- coding tools, 59–60
- color harmonies, 249
- Color selector, 91
- coloring code, 30, 59
- colors. *See also* additive color; background color; primary colors; secondary colors; subtractive color; tertiary colors
  - adding to text, 12
  - defined, 249
- column widths, setting, 157–158. *See also* widths
- Command key. *See* keyboard shortcuts
- commenting style sheets, 179–180
- company name, styling with Edge Fonts, 135–136
- company tagline, styling, 136
- comparing documents, 25
- complementary colors, 249
- computers, function of, 4–5
- connections
  - diagram of, 6
  - making, 5–7
  - to servers, 57
- #contact-form selector, creating, 165
- #container <div>, styling, 92
- container selector, choosing, 107
- containers
  - floating images in, 128
  - styling for CSS3 animation, 229–236
- content. *See also* optional content
  - adding to web pages, 73–76
  - converting to Library Items, 190–192
  - selecting, 86
- content containers
  - adding, 83–88
  - styling, 129–130
- content organization, viewing, 26. *See also* document tree
- contrast
  - adjusting, 117
  - defined, 249
- copying. *See also* get; put
  - folders, 39
  - pages, 32
  - plain text, 74
  - project files, 20–21
- Create tab, choosing, 21–22
- Creative Commons, 249
- Crop Image tool, 117
- CSS (Cascading Style Sheets), 7
  - commenting, 179–180
  - converting to external CSS files, 180–182
  - defined, 248
  - float property, 121–122
  - shorthand notations, 177–178
  - source order, 178–179
  - types of, 175
  - viewing disabled, 25
- CSS box model, 45, 90, 248
- CSS class rules, 49
- CSS class selectors, 176, 249
  - applying, 123
  - creating and applying, 121–122
  - duplicating, 123
- CSS Designer panel, 31, 54–55, 90–95
- CSS files. *See also* files
  - applying override settings, 222–224
  - customizing for Bootstrap settings, 221–222
- CSS float property
  - modifying, 123–124
  - using, 121–122
- CSS mode, using in Property inspector, 49–52
- CSS properties, adding, 75–76

CSS styling rules, 12–14, 28–30, 207  
  creating, 179  
  filtering, 214  
CSS3 animation, 229–236  
CTRL key. *See* keyboard shortcuts  
curved (line), 249  
cutting and pasting text, 73–74

## D

database, 15–16, 249  
declaration block, 249  
decorative fonts, 249  
default styles, removing from text links, 135.  
  *See also* styles  
deliverables, 249  
deprecated, 250  
descendant selectors, 176, 250  
design, cleaning up, 169  
design comp, 83–84, 250  
design elements, 250  
design principles, 250  
Design view, 27, 31, 44, 249  
Designer panel, 31, 54–55, 90–95  
Device Preview, 47  
diagonal (lines), 250  
diamond-bg.png file, 110  
dingbat fonts, 250  
direction, 250  
<div> tags, 85–88, 92–95  
<!doctype>, 9, 73, 250  
Document tabs, 42  
Document toolbar, 45, 250  
document tree, 250. *See also* content organization  
Document window, 41–48  
  File path, 46  
  media queries, 46  
  Refresh option, 46  
  related files, 45  
documents, comparing, 25  
DOM (Document Object Model), 8, 250  
DOM panel, locating, 26–27, 88  
domain names, 6, 250  
domain objects for ACA, 240–246  
dragging and dropping links, 81  
Dreamweaver, features of, 38  
Dreamweaver websites. *See also* websites  
  copying project files, 20–21  
  defining, 20–22  
  downloading sample files, 19–20

  naming, 22  
  site definition, 21–22  
  Workspace Layout, 23  
Dreamweaver workspace  
  customizing, 53  
  layout, 23, 259  
duplicating images, 115–116  
dynamic website, 16, 250

## E

Edge Fonts, styling company name with, 135–136.  
  *See also* fonts  
editable regions, 184–185, 189–190  
editing images, 115–116  
edits, round-tripping, 51  
element selectors, 76, 176, 250  
element styles, 28. *See also* styles  
elemental CSS properties, adding, 75–76  
elements  
  of art, 250  
  positioning with CSS, 90–95  
<em> tag, 49  
emphasis, 251  
encoding, 251  
error checker, 48  
error indicator, identifying, 170  
external style sheets, 175, 251

## F

fair use, 251  
faux column technique, using with sidebar,  
  106–108  
feedback loop, 251  
file extension, 16, 251  
file management, 47  
File panel, displaying, 23  
File path, displaying in Document window, 46  
file weight, 102, 251  
filename, locating, 32  
files. *See also* CSS files; folders; local and  
  remote files; project files; related files;  
  synchronizing files  
  backing up, 106  
  managing, 117  
  organization of, 16–18  
  refreshing, 57  
  saving, 32  
  synchronizing, 57, 199–200  
  updating on server, 198–199

Files panel  
  explained, 251  
  Mac versus Windows, 70  
  using, 56–57

Flick, Chris, 64

float property  
  modifying, 123–124  
  using, 121–122

flow, 251

fluid design, 207

focal point, 251

folder location  
  browsing to, 40  
  identifying, 21

folders. *See also* files  
  copying, 39  
  browsing to, 22  
  organization of, 16–18

font family, 251

font settings, displaying, 75

font weight, 251

fonts. *See also* Edge Fonts; sans serif fonts;  
  script fonts; serif fonts; web fonts  
  defined, 251  
  fixing, 168–169  
  handwritten, 252  
  monospaced, 254

footer, styling links in, 95

#footer <div>, styling, 94

Footer area, creating, 85

footer ID, creating, 88

form tag, inserting, 162–163

formatting text, 77–81

forms  
  adding function to, 164  
  defined, 251  
  form tags, 162–163  
  inserting text from Word, 161–162  
  list labels, 163  
  Select form, 163  
  styling, 165  
  text fields, 162–163

forward slash (/) character, 8

Forward/Back buttons, 46

frameworks, 207, 251

FTP (File Transfer Protocol), 251

## G

geometric lines and shapes, 252

get, 252. *See also* copying; put

GIF (Graphics Interchange Format), 101

glyphs and glyphicons, 218–219, 252

graphic design, 70

group selectors, 176, 252

guides, aligning elements with, 130–131

## H

<h1> tag, 10, 26

hand-drawn lines, 252

handwritten fonts, 252

head (of document), 252

head in web page, 9

<head> tag, 9, 26

header, adding links in, 134

#header <div>, styling, 93

header images, inserting and floating, 133–134.  
  *See also* images

heading properties, setting, 157–158

heads-up display, 220

hexadecimal, 252

hey.html web page, 9

hidden formation, locating, 26

highlight, 252

home page. *See also* web pages  
  creating, 69  
  replacing, 185–186

horizontal, 252

hover values, setting, 153–154

href value, writing, 82

HTML (Hypertext Markup Language), 7, 26, 252

HTML mode, using in Property inspector, 48–49

HTML Reference Page, 10

HTML standards, release of, 206–207

HTML5. *See also* tags  
  explained, 252  
  tags and media queries, 212–213

HTML5 video  
  assets, 225–226  
  inserting, 226–228

HTTP (Hypertext Transfer Protocol), 5, 253

hyperlinks, 4, 252. *See also* active hyperlink  
  adding for other pages, 147–148  
  adding in header, 134  
  creating, 81–82  
  dragging and dropping, 81  
  styling in footer, 95

hyphenation, 253

## I

- ID selectors, 176, 253
- ideographs (ideograms), 253
- IDs versus classes, 87
- image container, styling for CSS3 animation, 231
- image editing tools, 51–52, 116–117
- image file formats
  - GIF (Graphics Interchange Format), 101
  - JPEG (Joint Photographic Experts Group), 101–102
  - PNG (Portable Network Graphics), 102
- image maps, 52, 253
- images. *See also* background images; header images
  - adding alt text to, 124
  - choosing background colors from, 151
  - clarifying paths to, 188
  - creating structured layouts with, 125–132
  - duplicating, 115–116
  - editing, 115–116
  - floating inside containers, 128
  - floating into position, 121–124
  - inserting in sidebar, 144–145
  - inserting using Assets panel, 113
  - locating, 35, 145
  - positioning in sidebar, 145–146
  - Property inspector for, 50–52
  - replacing on pages, 34–35
  - resizing, 116
  - scaling and sizing, 114–115
  - using in page backgrounds, 108–111
  - width and height, 51
- `<img>` tag, 11, 27
- implied lines, 253
- indent settings, 49, 253
- `index.html` file
  - naming, 69
  - saving, 32
- information architecture, 68, 253
- information box container, styling, 232–236
- inline styles, 50, 175, 253
- Insert panel, displaying, 86
- Inspect mode, 45
- interactive forms
  - adding function to, 164
  - form tags, 162–163
  - inserting text from Word, 161–162
  - list labels, 163
  - Select form, 163
  - styling, 165
  - text fields, 162–163

- internal style sheets, 28, 175, 253

## Internet

- development of, 4
- explained, 13
- versus World Wide Web, 4

IP (Internet Protocol), 6, 253

IP address, 6, 253

iPhone, release of, 206

Italic settings, accessing with Property inspector, 49

iterations and iterative work, 253

## J

- JavaScript, 7–8, 207, 253
- JPEG (Joint Photographic Experts Group), 101–102
- jQuery Mobile, 207, 217–219, 254
- justified, 254

## K

- kerning, 254
- keyboard shortcuts
  - cutting and pasting text, 73–74
  - opening web pages, 9
  - selecting content, 86

## L

- layouts. *See* structured layouts
- leading, 254
- Learn-Dreamweaver folder, contents of, 39
- `<li>` tag, 80
- library items, 254
  - adding optional content with, 189–194
  - converting content to, 190–192
  - creating from scratch, 192–193
  - inserting and updating, 194
- licensing, 254
- ligatures, 254
- light source, 254
- lines
  - defined, 254
  - geometric, 252
  - hand-drawn, 252
  - implied, 253
  - organic, 255
  - styles, 258
  - varying width, 259
  - weight, 259
- link values, setting with pseudo-class selectors, 152–153
- links. *See* hyperlinks; navigation links

linting technology, 170, 254  
list items, assigning width and margins to, 150  
list labels, including in interactive forms, 163  
list properties, removing defaults for, 150  
lists  
    creating, 80  
    unordered, 80  
Live view, 31, 44, 46–47, 254  
    modifying Bootstrap properties, 219–220  
    of responsive design, 210–211  
local and remote files, synchronizing, 199–200.  
    *See also* files  
Love Hate/Love makes me happy, 153  
LVHA (link, visited, hover, active), 153

## M

Macromedia, 15  
mailto: link, 161  
margin settings  
    adjusting, 143–144  
    applying to list items, 150  
    displaying, 75  
    fixing, 94–95  
media queries, 46, 254  
    in Bootstrap, 217  
    in CSS Designer panel, 55  
    and HTML5 tags, 212–213  
    and responsive design, 208–214  
metadata, 254  
Microsoft Word, cleaning up tables from, 155–157  
minified, 254  
mission class, 11  
mobile design, 207  
model releases, 254  
monochromatic, 254  
monospaced fonts, 254  
movement, 254

## N

naming  
    files and folders, 18  
    websites, 22  
<nav> tag, inserting, 148–150  
navigation elements  
    adding links to other pages, 147–148  
    background colors for images, 151  
    hover values, 153–154  
    removing default list properties, 150  
    setting link values, 152–153  
    width and margins for list items, 150

navigation links, changing, 33–34  
navigation sidebar  
    adjusting widths, 143–144  
    checking work, 141–142  
    client requirements, 141  
    inserting images in, 144–145  
    modifying margins, 143–144  
    positioning images in, 145–146  
    styling, 106–108, 142–143  
navigation tags, inserting, 149–150  
NC (NonCommercial), 255  
ND (NoDerivs), 255  
negative space, 255

## O

object shadow, 255  
<ol> tag, 49  
open source, 255  
opening  
    page-starter.html file, 24, 54  
    prototype.html, 105  
    responsive-starter.html file, 53  
    Site Setup dialog window, 39  
    web pages, 9  
    website-specs.txt, 78  
optimization, 102, 255  
optional content, adding with library items,  
    189–194. *See also* content  
organic lines and shapes, 255  
organization of content, viewing, 26. *See also*  
    document tree  
OS X, system and folder names, 210  
override settings, applying in custom CSS files,  
    222–224

## P

<p> tag, 10  
page elements, positioning with CSS, 89–95  
page font settings, displaying, 75  
page inspector, starting in browser, 224  
page margin settings, displaying, 75. *See also*  
    margin settings  
page properties, modifying, 71–76  
page title, 73, 255  
pages. *See* web pages  
page-starter.html file, opening, 24, 54  
panel groups, 41–42  
panel options, accessing, 53  
panels and panel groups, 52–59

paragraph  
  settings and spacing, 255  
  styles, 12–13  
pasting text, 73–74  
path  
  defined, 255  
  determination of, 17  
  to images, 188  
pattern, 255  
phone app, building, 237  
photos. *See* images  
Photoshop, round-trip editing with, 118–120  
pictograph (pictogram), 255  
pictures. *See* images  
plain text, copying, 74  
PNG (Portable Network Graphics), 102  
preferences, setting, 112  
Preview in Browser, 47, 110, 171  
primary colors, 255. *See also* colors  
profile page, customizing, 214  
project creep, 208, 255  
project deadlines, 255  
project files, copying, 20–21. *See also* Chris  
  the Cartoonist project; files  
project scope, 256  
properties, modifying, 30  
Property inspector, 30, 41–42, 256  
  for images, 50–52  
  Inline styles, 50  
  switching views of, 48–49  
  views, 52  
Property inspector for text  
  in CSS mode, 49–52  
  in HTML mode, 48–49  
proportion (scale), 256  
protocol, 5, 256  
prototyping in web design, 105. *See also* responsive  
  prototype page  
pseudo-class selectors, 176, 256  
  listing, 153  
  setting hover values with, 153–154  
  setting link values with, 152–153  
public domain, 256  
publishing to remote server, 197–198  
put, 256. *See also* copying; get

## R

radial, 256  
Ray the hipster werewolf, 72  
reflected highlight, 256

refreshing files, 57  
related files, 45, 256. *See also* files  
remote and local files, synchronizing, 199–200.  
  *See also* files  
remote server. *See also* server  
  defining, 195–196  
  publishing to, 197–198  
  updating files on, 198–199  
repetition, 256  
replacing images, 34–35  
representative shapes, 256  
Resample Image tool, 117  
resizing images, 116  
responsive design, 207–214  
responsive-starter.html file, opening, 53–54  
RGB (red, green, blue), 256  
rhythm, 256  
root folder, 16–17, 256  
round-trip editing with Photoshop, 51,  
  118–120, 256  
rows in tables, styling, 158–159  
rule of thirds, 257

## S

SA (ShareAlike), 257  
sans serif fonts, 257. *See also* fonts  
saving files, 32  
scaling images, 114–115  
script fonts, 257. *See also* fonts  
secondary colors, 257. *See also* colors  
Select form, inserting, 163  
selecting tags, 27  
selector types, 176  
semantic structure, 77–81  
serif fonts, 257. *See also* fonts  
server. *See also* remote server  
  versus client, 6  
  connecting to, 57  
  defined, 257  
shapes, 255, 257  
Sharpen tool, 117  
shorthand notations, setting in CSS, 177–178  
sidebar  
  adjusting widths, 143–144  
  checking work, 141–142  
  client requirements, 141  
  inserting images in, 144–145  
  modifying margins, 143–144  
  positioning images in, 145–146  
  styling, 106–108, 142–143

site definition, creating, 21–22. *See also* websites  
 site reports, running, 170  
 Site Setup dialog window, opening, 39  
 site structure, creating, 68  
 sitemap
 

- creating, 68
- defined, 257

 size of window, 47  
 sizing images, 114–115  
 sketches, 257  
 slab serif fonts, 257  
 small caps, 257  
 Snippets panel, opening, 216  
 space, 257  
 specifications, 257  
 spelling, checking, 170  
 Split view mode, 25, 257  
 starter page, customizing, 32–35. *See also* web pages  
 static website, 15–16, 258  
 stock photos, 258  
 storyboard, 258  
 <strong> tag, 49  
 structured data, styling, 155–159  
 structured layouts, creating with images, 125–132  
 style (line), 258  
 style rules. *See* CSS styling rules  
 style sheets, 7
 

- commenting, 179–180
- converting to external CSS files, 180–182
- defined, 248
- float property, 121–122
- shorthand notations, 177–178
- source order, 178–179
- types of, 175
- viewing disabled, 25

 <style> tag, 26  
 styles, applying to paragraphs, 12–13. *See also*

- default styles; element styles

 styling
 

- card container, 231
- company name with Edge Fonts, 135–136
- <div> tags, 92–95
- hyperlinks in footer, 95
- image container for CSS3 animation, 232
- information box container, 232–236
- interactive forms, 165
- navigation sidebar, 106–108, 142–143
- rows in tables, 158–159
- structured data, 155–159
- tagline, 136
- text, 78

 subfolders, 17  
 subtractive color, 258. *See also* colors  
 Subversion, 258  
 <summary> tag, 11  
 swashes, 258  
 symmetrical, 258  
 synchronizing files, 57, 199–200. *See also* files  
 syntax
 

- coloring, 30, 59
- defined, 7, 258

## T

table and table cell properties, setting, 157  
 <table> tag, 11  
 tables
 

- cleaning up from Word, 155–157
- styling alternate rows in, 158–159

 tag selector, 48  
 tagline, styling, 136  
 tags. 8–10, 14, 258. *See also* HTML5
 

- <a>, 11, 27–28
- <article>, 11
- <blockquote>, 49
- <body>, 9, 27, 76
- <div>, 85–88, 92–95, 127–129
- <em>, 49
- <h1>, 10, 26
- <head>, 9, 26
- <img>, 11, 27
- <li>, 80
- <nav>, 148–150
- <ol>, 49
- <p>, 10
- selecting, 27
- <strong>, 49
- <style>, 26
- <summary>, 11
- <table>, 11
- <td>, 76
- <th>, 76
- <title>, 9, 26
- <ul>, 49

 team page, building, 187–188. *See also* web pages  
 templates
 

- creating, 183–184
- defined, 258

tertiary colors, 258. *See also* colors

text

- adding colors to, 12
- cutting and pasting, 73–74
- inserting into interactive forms, 161–162
- styling versus formatting, 78
- and text formats, 77–81

text fields, inserting into interactive forms, 162–163

text links, removing default styles from, 135

texture, 258

thumbnails of images, displaying, 51

title, changing for web page, 33

`<title>` tag, 9, 26

toolbars. *See* coding tools; Document toolbar

tracking, 258

type size, 258

typeface, 258

## U

`<u>` tag, 49

unity, 259

unordered list, creating, 80

updating files on server, 198–199

URL (Universal Resource Locator), 259

usability of site, 68, 259

## V

value, 259

variety, 259

varying width lines, 259

vertical and vertical scale, 259

videos

- for ACA objectives, 240–246

- delivering web pages, 7

View menu, 25

viewport, 259

views

- choosing, 43

- Property inspector, 52

- switching between, 31, 43

- switching for Property inspector, 48

## W

W3C (World Wide Web Consortium), 10, 206, 259

W3C HTML Reference Page, 10

web. *See* World Wide Web

web address, example of, 5

Web browser. *See* browser

web fonts, 168–169, 259. *See also* fonts

web pages. *See also* home page; starter page;

- team page

- adding to sites, 188

- copying, 32

- explained, 14

- glyph library, 218–219

- hidden formation, 26

- HTML (Hypertext Markup Language), 26

- making, 9–10

Webcomic Alliance podcast, 64

websites. *See also* Dreamweaver websites; site

- definition

- “Capes & Babes” online comic, 64

- defined, 259

- structure of, 15–16

- W3C HTML Reference Page, 10

website-specs.txt, opening, 73, 78

weight (line), 259

widths, adjusting. *See also* column widths

- adjusting, 143–144

- assigning to list items, 150

window size, 47

wireframes

- defined, 259

- working with, 83–85

Word, cleaning up tables from, 155–157

workspace

- customizing, 53

- layout, 23, 259

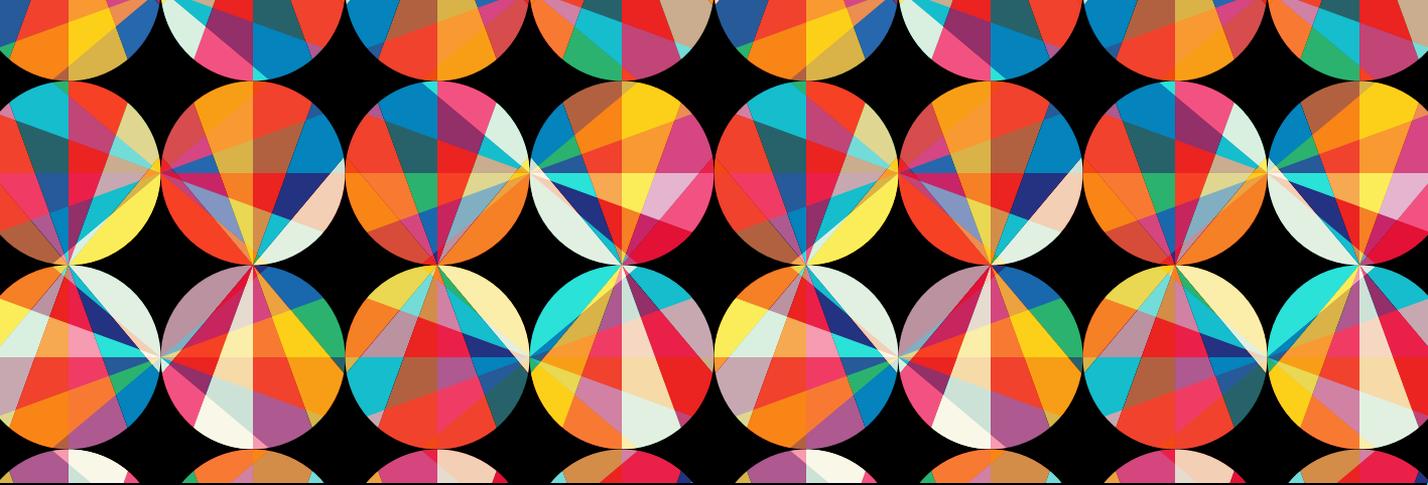
World Wide Web, 4, 13, 259

writing in code, learning, 7. *See also* code

WULVS (Werewolves UnLimited Valuables Security

- Services), 67

WYSIWYG (What You See Is What You Get), 15



## CHAPTER OBJECTIVES

### Chapter Learning Objectives

- Getting comfortable with creativity, and learning that creativity is an obtainable skill.
- Learn about the elements of art and how to use them to communicate visually.
- Learn how to use the principles of design to effectively direct the attention of viewers when viewing your design work.
- Learn about a design hierarchy and how the elements and principles can be understood to help design compelling visual compositions.
- Understand how the principles of design affect the design decisions made in web authoring projects.

### Chapter ACA Objectives

#### **DOMAIN 2.0**

#### UNDERSTANDING WEB AUTHORING

- 2.2** Demonstrate knowledge of basic design principles and best practices employed in the web authoring industry, such as usability, accessibility, consistency, and visual hierarchy.
- 2.6** Demonstrate knowledge of digital images, text, interactive elements, animation, and video and audio content as they pertain to web authoring.

## CHAPTER 8

# Leveling Up with Design

Throughout the course of this book you've learned how to use Dreamweaver, and there have been lots of hints about the design properties that lead to good solid design. Alignment, spacing, and even how to use fonts have all been woven into the lessons so far. Now that you have a good grasp of the Dreamweaver tools, it's time to level up and learn more about the best ways to use them. As with any skill, understanding how a tool works and becoming a master craftsman are two completely different levels of achievement. In many ways, they're really different ways of thinking about the tools you've learned to use.

As an example, think about carpenters. Their initial level of learning covers tools such as saws, hammers, and drills. They learn how to use the tools correctly and when to apply specific techniques, including cutting with or against the grain and how best to join wood together. Using the right techniques, they can effectively build anything.

You're now at the point when the only way to get better is to practice and to learn the thought processes that a real master craftsman goes through to create unique and creative work. The beauty of this stage is that it's when you start to become an artist. Being good at any tool is not just about knowing how it works and what it does, but about knowing *when* to use it and what techniques to apply to create something new.

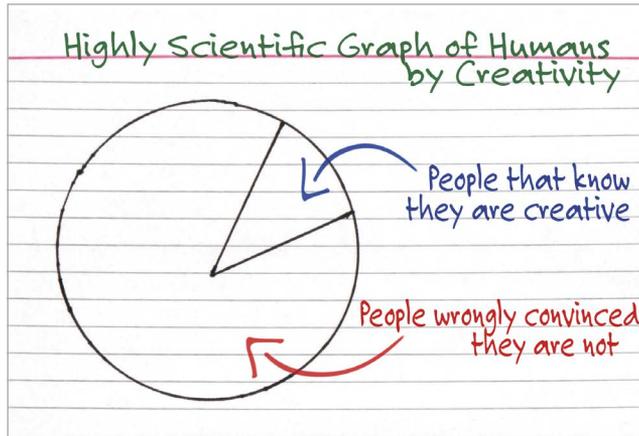
★ *ACA Objective 2.2*

▶ *Video 8.1 Design  
School: Introduction*

# Creativity Is a Skill

▶ **Video 8.2** Design School: Creativity Is a Skill

Let's start by talking about creativity. **Figure 8.1** is a highly scientific and statistically accurate diagram that I've created.



**Figure 8.1** Human creativity

Okay, so perhaps it isn't *exactly* statistically accurate and maybe my methods weren't really scientific, but it's still true. Creativity is a skill that you can learn and improve with practice. But you have only one way to guarantee that you'll *never* be more creative than you are now:

Give up.

The biggest creative problem most people have is that they give up. Some people give up before even making their first effort. Others give up after their fifth effort or after 15 minutes of not creating a masterpiece. Nobody creates a 15-minute masterpiece. A big difference between a great artist and a terrible artist is that a great artist has tried and failed, made some adjustments, and tried again. The next try may also lead to failure, but the artist pushes forward. The artist isn't deterred by bumping into a problem because it's all part of the learning and creative process.

Of course, you can't know how far you can take your creative skills until you start practicing and expanding those skills. But if you never use them, they'll only grow weaker. Remember that every effort, every new attempt, and every goof-up builds strength.

## Getting a Creative Workout

This chapter is about developing your creative skills and flexing those muscles. It's about turning the craft of being a digital designer into a natural ability. Even if you take the big step of becoming an Adobe Certified Associate, you must realize that passing the test is just getting through the tryouts. When you get that credential, it's like being picked to be on a professional football team. It's certainly something to be proud of, but the real goal is the *Super Bowl*. You have a lot more work ahead, but it is fun and rewarding work.

This book obviously can't directly respond to the skills you are developing, but it has introduced you to some exercises to help you explore and enhance those skills. Getting beyond the basics of creativity and design is just a matter of applying your basic skills in ways you've never tried. That's really all there is to creativity.

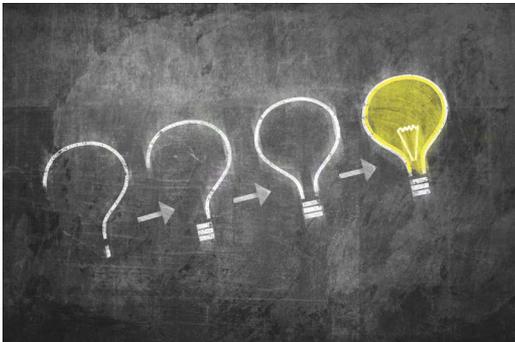
## Prepping Your Mindset

This is the most important part of increasing your creative skills. And for most of us, it's the hardest because today's culture is so focused on instant success and efficiency that people learn to fear the essence of creativity:

Failure.

Failure—and more specifically, the ability to comfortably accept failure—is the key ingredient in developing your artistic skillset. This is especially true if you are a beginner and aren't entirely happy with your current abilities. Just face that you're a design baby. And start acting like one.

Yes, I just encouraged you to act like a baby. But I mean an actual infant, not a grown person who is throwing a fit. Real babies are fearless and resilient when they fail. They keep trying. They don't give up. Most of the time, they don't even realize that they're failing, because they haven't yet learned that concept (**Figure 8.2**).



**Figure 8.2** Failure is the path to success.

That's what you need to tap into. In art, there are no failures, there are only quitters. Tap into your inner infant. Try and fail, and try and fail, and try and fail. Eventually, you'll take your first step. You may quickly fail again on your second step, but don't give up. Babies don't know that they've failed. They just know that they got *a little bit closer*. They know what every true artist knows, and what the rest of us need to remember but have forgotten along the way: *Failure is the path*.

★ ACA Objective 2.2

## The Design Hierarchy

▶ **Video 8.3** *Design School: The Design Hierarchy*

Almost everyone recognizes art when they see it, but nobody can agree on how to create it. In an attempt to provide a framework, artists have developed a list of **design elements**, the building blocks of art, and **design principles**, the essential rules or assembly instructions of art. Although every artist understands the importance of the elements and principles of art, there is no official list of those elements and principles that all artists agree on. This can be incredibly frustrating. How can you study and learn about something that nobody can fully identify?

Actually, you should learn to celebrate this fact. The lack of an “official” list means that you really can't be wrong. Different artists in different media look at the elements and principles of art differently. A sculptor and a painter see and approach their art in different ways. A movie producer approaches her art unlike a costume designer approaches his. But all these artistic people still study and embrace some elements and principles to guide their art.

## Applying the Design Hierarchy

There's a helpful way to understand and think about artistic elements and principles: the “design hierarchy.” This isn't the *ultimate* approach to understanding and interacting with the artistic elements and principles. (Frankly, an “ultimate” approach probably doesn't exist.) It is more of an organized starting place that might help you focus your design skills (**Figure 8.3**).

Design, unlike art, has a purpose. It's often about creating or accomplishing something specific, rather than simply enjoying or exploring an artistic impulse. A design task might require that you advertise a product, communicate an idea, or promote a cause or specific issue. For most people it's actually easier to be artistic when the result is something that has a purpose.

Thinking through these building blocks can also serve as a great little exercise when you're looking at a piece of work that you are unhappy with but can't quite figure out why. Sometimes, just rolling through the following elements and principles in your mind can be an excellent creative checklist.

## START WITH A FOCAL POINT

The **focal point** is what your design is all about. In an advertisement, it's your "call to action" phrase that motivates the consumer. For a cause, it's the primary message that your cause is trying to get across. It's the primary idea that you want people to take with them, like the title of a book or a chapter. It's the "sales pitch" of your design. When you're working with a client, they'll often be the one to determine what the focal point should be. Your job as a designer is to translate their big idea into something visual and beautiful.

Because the focal point is the "focus" of a design, it should be the most memorable aspect of the design. In a symmetrical or radial design, center the focal point for the most emphasis. In an asymmetrical design, let it fall on one of the natural focal points, which you'll learn about shortly when you study space.

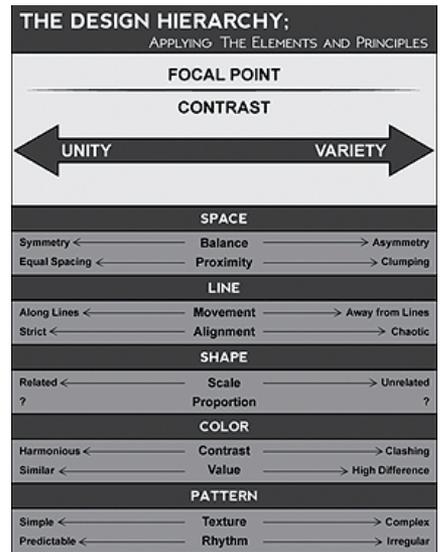
**Critical question:** *Do people know where to look in my design?*

## CREATE FOCAL POINTS USING CONTRAST

**Contrast** creates a focal point by generating "tension" in the image. Place one red golf ball in a pile of 10,000 white golf balls, and you'll notice it immediately. The most dramatic way to create contrast is to vary the most common characteristic of the elements in the design. If you have a bunch of multicolored circles that are the same size, then varying the size of one circle, regardless of its color, will create a contrast that draws the eye (**Figure 8.4**). Keep the shape the same size, but make it a star instead of a circle to generate the same tension.



**Figure 8.4** Focal point is created with contrast: Make one thing different and it stands out.



**Figure 8.3** Design hierarchy

Establishing unity in a design is critical to allowing you to create tension. Without a degree of unity, you cannot create a focal point and the eye struggles to find something to focus on. Design is about forcing a focal point to be where you want it. When too many different, unrelated elements are present in your design, it makes chaos, and the viewer will not know where to look.

*Critical question: Does my design lack sufficient contrast to see the important features clearly?*



**Figure 8.5** Keep the contrast in your image low toward the unity side unless you're trying to make something stand out.

## CONTRAST HAS A RANGE OF TENSION FROM UNITY TO VARIETY

Imagine that contrast can be described as the “tension” in the image. And that contrast can be imagined as a range from low contrast, which we call **unity**, to high contrast, which we call **variety** (Figure 8.5). Compositions with very low contrast have very little tension, and these designs are generally perceived as peaceful and calm but sometimes can feel emotionless, cold, and lifeless. Compositions with very high contrast have a lot of tension; they are generally perceived as energetic, lively, active, and hot but sometimes feel unpredictable or emotional.

## A FRAMEWORK FOR CONNECTING THE DOTS

The following framework connects artistic elements and principles to help you at the onset of the creative journey. They're good starting points. Remember that with design, you're not trying to *define* the artistic elements and principles as much as trying to *embrace* and *apply* them. As with every study of the artistic elements and principles, you won't arrive at a destination. Rather, you'll embark on a journey. These connections should help you make sense of the elements and principles long enough to help you find your own way to understand and apply them.

- **Create balance and proximity by arranging the elements in your compositional space.** Tension develops among the different areas of your composition. Unify the design with symmetry and equal spacing, or vary it using asymmetry and grouping elements.

*Critical question: Does my design use space to help communicate relationships?*

- **Create movement and alignment by arranging your elements along lines.** Tension is created with the direction or flow of your composition. Unify the design with strict alignment or movement along straight or flowing lines, or vary it with random, chaotic movements away from any single line or flow.

*Critical question: Do my design feature elements, such as lines and borders, help communicate order and flow?*

- **Create scale and proportion by carefully designing shapes in your composition.** Tension is created among the sizes of elements in your composition. Unify the design with similar or related sizes of elements, or vary it with a mix of unrelated sizes. Keep in mind that a paragraph has a shape, and a group of elements has a shape. Look at *areas* as well as objects in your design.  
*Critical question:* Does my design consider the message sent by the size of shapes I create?
- **Create themes and feeling by using typography, colors, and values in your composition.** Tension is created among elements of different type, value, or color. Unify the design with related type, colors, and values, or vary it with clashing colors or extreme difference in values or type.  
*Critical question:* Does my design consider the message sent by the typography, color, and value I used?
- **Use repetition and rhythm to create patterns and texture in your compositions.** Tension is created among elements with different texture or patterns. Unify the design with simple repetition to produce predictable patterns and rhythms, or vary it using complex, irregular, or chaotic patterns and rhythms.  
*Critical question:* Does my design consider the message sent by the patterns and textures I designed?

## Wrapping Up the Design Hierarchy

If you're basically familiar with the artistic elements and principles, this hierarchy might be a framework you can use. But the goal of art is to continually explore the elements and principles. The next section will cover them all, and I'll share some challenges that may help you dig a little deeper into learning to use them well.

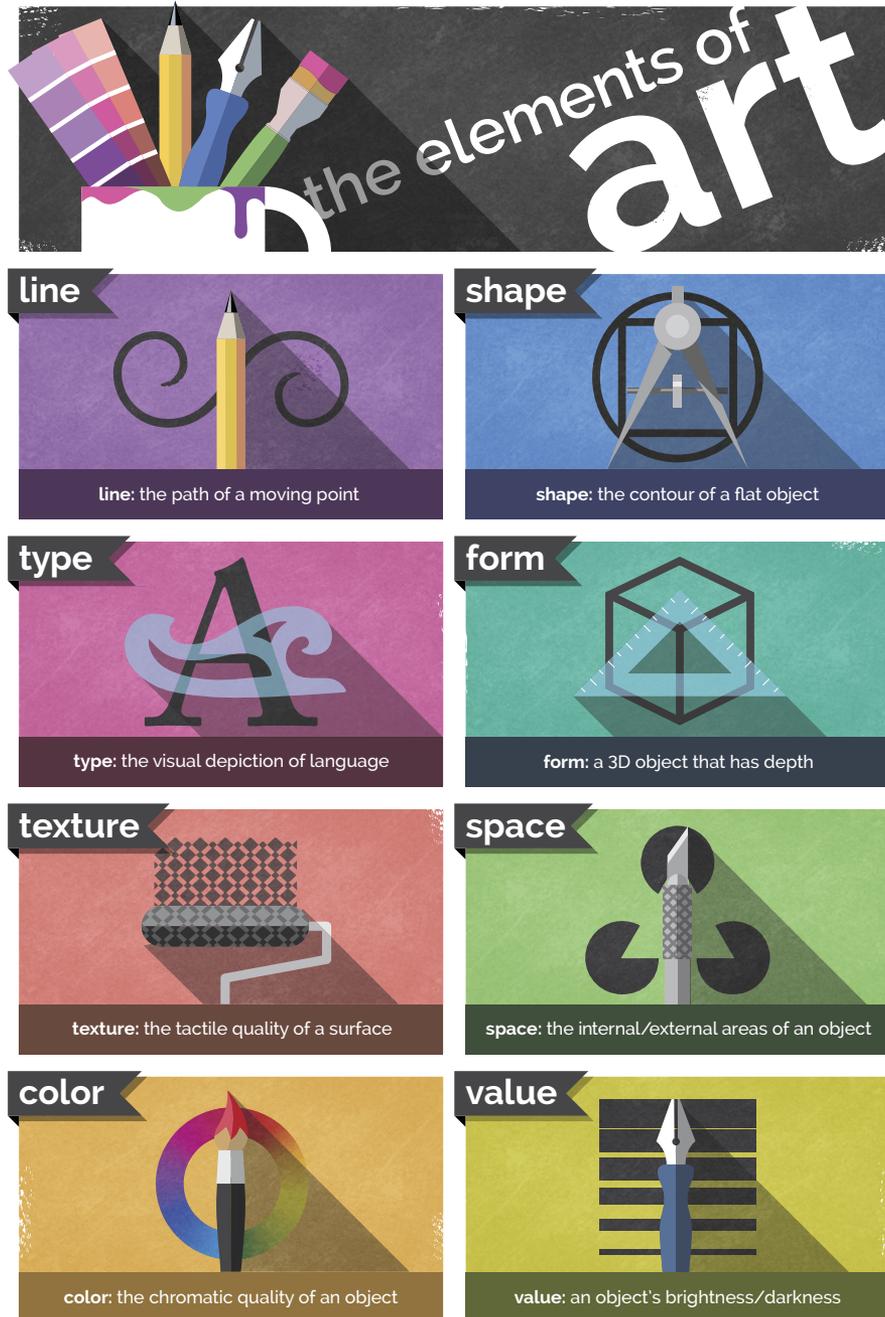
## The Elements of Art

The **elements of art** are the building blocks of creative works. Think of them as the “nouns” of design. The elements are space, line, shape, form, texture, value, color, and type (**Figure 8.6**). Many traditional artists leave type off the list, but for graphic designers, it is a critical part of how we look at design (and besides that, type is really fun to play with).

★ ACA Objective 2.2

▶ **Video 8.4** Design School: The Elements of Art

**Figure 8.6** The nouns of design



## The Element of Space

Space is the first element of art to consider (Figure 8.7). It also happens to be one of the most abused, overlooked, neglected, and underrated elements in design. You can look at and consider space in multiple ways.

▶ **Video 8.5** *Design School: The Element of Space*



Figure 8.7 Space

### SPACE AS YOUR CANVAS

The most basic way to look at **space** is as your canvas or working area. In web design, a document's dimensions are determined by the resolution of the page you're creating. You may be designing for a full-screen presentation on a computer or tablet, or you might be designing for a screen that displays on a phone. These dimensions define the space.

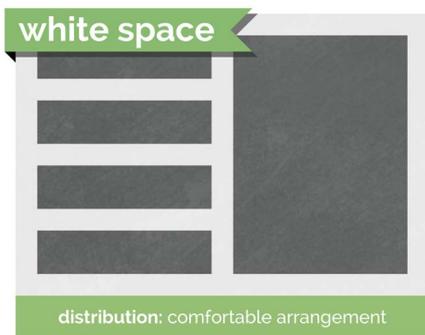
#### LEVEL UP: A PENNY IN SPACE

Here's a fun challenge:

- 1 Grab a piece of blank paper, and simply place a coin on it. Don't pick a random place to put it; *design* the spot where it should go.
- 2 Look at the paper with the coin on it and get a sense for the "feeling" it creates.
- 3 Fold the paper any way you want, except directly in half, so that you have a differently shaped space, and repeat the process.

Can you see how simply using different sizes and dimensions of space—with the exact same content—can change the way your art "feels"?

And did it enter your mind to fold the paper in a way that wasn't rectangular? Not all spaces have 90-degree corners. Think outside the box...*literally*.



**Figure 8.8** Use space to create comfortable arrangements of elements for your design.

## SPACE AS A CREATIVE TOOL

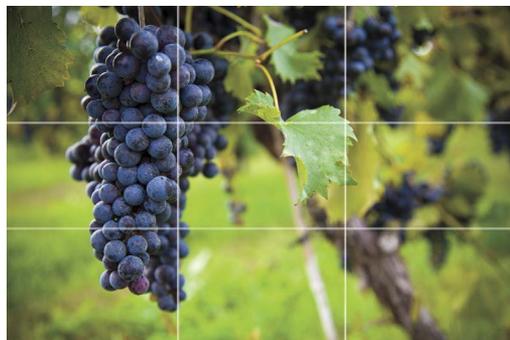
Another important way to look at space is as a design element. The inability to use space well is an obvious sign of a new designer. A crowded design practically *screams* “newbie.” It takes a while to use space well, and it requires some practice. Start learning to use space as a creative element and you’ll see a drastic increase in the artistic feel of your designs (**Figure 8.8**).

When used properly, space can provide the following benefits to your designs:

- **Creating emphasis or focus.** When you have space around something, it tends to give it emphasis. Crowding things makes them seem less important.
- **Creating feeling.** Space can be used to create a feeling of loneliness, isolation, or exclusivity. It can also be used to create a feeling of seriousness or gravitas.
- **Creating visual rest.** Sometimes space is needed to simply create some visual “breathing room” so that the other elements in your design can speak as intended.

## THE RULE OF THIRDS

The **rule of thirds** provides a great way to think about space. This rule is critical in photography and video and also applies to all visual arts. To visualize it, think of the rule of thirds as a tic-tac-toe board overlaid onto your design. Two horizontal and two vertical divisions create nine equal boxes on your design. The basic rule is that major elements of your design should fall on the dividing lines, and the areas of emphasis for the design should fall on the intersections. Using this rule in your designs creates compositions that have much more interest, tension, and visual strength (**Figure 8.9**).



**Figure 8.9** Use the rule of thirds to determine where major elements should fall to create a visually appealing layout.

### TIP

*Beginning designers center everything, which can contribute to a static look and a feeling of boredom. A great way to get started with the rule of thirds is to simply stop centering things. You’ll notice that you tend to move elements quite naturally to a focal point suggested by the rule of thirds.*

## NEGATIVE SPACE

In the design industry, people use the term **negative space** or “white space” to refer to blank areas in the design, even if you’re working on a colored background (**Figure 8.10**). Negative space is without a doubt one of my favorite creative uses of the element of space. It can be fun to explore using negative space to create logos or designs that are clever.



**Figure 8.10** Negative space is the space that surrounds an object and helps to define and separate the objects around it.

### TIP

Check out some awesome, updated examples of negative space online at [www.brainbuffet.com/design/negative-space](http://www.brainbuffet.com/design/negative-space).

Sometimes you can place a really boring idea in negative space to get moving in a new, creative direction. It can be tough for beginners to even “see” negative space, but once you start to look for it, you’ll start to see creative uses of it everywhere. Begin to use it, and your design skills will jump up a level.

The following elements are much easier to understand because you can think of them as things. (Whereas we tend to think of space as the absence of things, or “nothing.”) Learn to use space well. The ability to do so is the mark of an experienced and talented artist.

### LEVEL UP: SKETCHING NOTHING

This simple exercise will help you start to see negative space and learn how to create it.

Place a chair on a table, and then simply sketch everything you see *except* the chair. Don’t worry if your sketch is messy—this is not about your artistic skill; it’s about learning to see. Just focus on learning to see the space around things rather than the things themselves. When you’re finished, you’ll have a chair-shaped hole in your drawing. Notice that the space where you drew nothing has the most visual impact. Space is powerful.

### NOTE

For many people, the terms *negative space* and *white space* are interchangeable. If you’re unclear about what someone is asking for, ask for clarification. Art and design are not entirely technical, accurate, and organized processes, so you’ll need to get used to lots of “mushy” terms that are used in multiple ways.

## The Element of Line

▶ **Video 8.6** *Design School: The Element of Line*

The meaning of **line** is pretty obvious (**Figure 8.11**). Although technical definitions exist, such as “a point moving through a space,” we are all aware of what a line is. If you’re not sure, ask a kindergartener. A line is exactly what you think it is: a mark with a beginning and an end. Don’t overthink the basics. We’re going to dig deeper than that, but start with the basic idea and then build the new understanding upon it.

**Figure 8.11** Element of line



### LEVEL UP: LINE 'EM UP

A fun side quest is to simply experiment with lines. See how many different ways you can think about the concept of a line. You’ll be exploring more of these ideas in just a minute, but first see how many different kinds of lines you can draw.

Here are some quick ideas to get you started:

- **Level 1:** Short, long, straight, wavy, zigzag, geometric, organic
- **Level 2:** Angry, lonely, worried, excited, overjoyed
- **Level 3:** Opposition, contrast, politics, infinity

Try to create as many different kinds of lines as you can, and find a word to describe each one. There are no wrong answers here; this is art.

Remember that you’re just drawing lines, not *pictures*. So to indicate sadness, don’t draw an upside down “u.” The goal is to draw a line that, in itself, represents sadness. It’s a little challenging as you move away from descriptions and closer to abstract ideas, but that’s the point. The best artists learn to “hint” at feelings and concepts in their art. Explore.

## COMMON LINE DESCRIPTORS

By thinking about the different ways that lines are used or drawn, you can more fully understand the meaning that your lines are giving to your design.

The following adjectives are often associated with lines:

- **Direction:** A common way to describe lines is to describe the direction in which they travel (**Figure 8.12**). Lines can be horizontal, vertical, or diagonal. Sometimes the way a line is drawn can even express movement in a particular direction. Think about this and make sure that the lines of your art are moving in the directions you want. **Vertical** lines tend to express power and elevation, and **horizontal** lines tend to express calm and balance. **Diagonal** lines often express growth or decline, and they imply movement or change.
- **Weight:** Another common descriptor is the weight of a line, which describes its thickness (**Figure 8.13**). Heavy or thick lines generally represent importance and strength, and they tend to feel more masculine. Light or thin lines generally communicate delicacy and elegance, and they tend to represent femininity. Using a variable line width implies natural, artistic, flowing feelings and often suggests an intentional grace or natural beauty.



**Figure 8.12** The direction of a line can convey meaning and feeling.



**Figure 8.13** Weight can be used to convey importance and strength.

- **Flow:** This word is the category that is related to the energy conveyed by lines and shapes (**Figure 8.14**). **Geometric** lines tend to be straight and have sharp angles; they look manmade and intentional. Geometric lines communicate strength, power, and precision when used in design. **Curved** lines express

### TIP

*Having a hard time remembering horizontal and vertical directions? “Horizontal” (like the horizon) has a cross-bar in the capital “H” that is a horizontal line itself. Vertical starts with the letter “V,” which is an arrow pointing down.*

fluidity, beauty, and grace. **Organic** lines are usually irregular and imperfect—the kind of lines you find in nature or as the result of random processes. Organic lines represent nature, movement, and elegance. **Chaotic** lines look like scribbles and feel very unpredictable and frantic. They convey a sense of urgency, fear, or explosive energy.

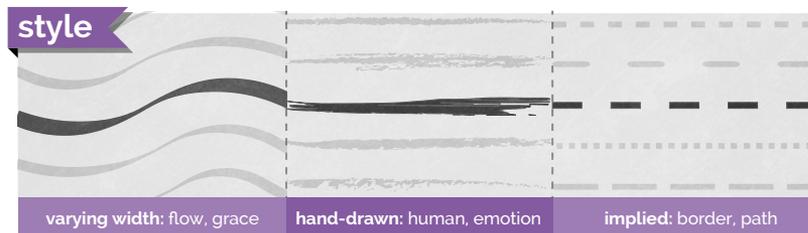
- **Style:** A line style is an effect applied to the line, such as a double line or dotted line (Figure 8.15). (In graphics programs such as Photoshop, you can select from preset line styles or create your own.) Words used to describe popular line effects include *varying width*, *hand-drawn*, and *implied*. **Varying width lines** are useful for expressing flow and grace. **Hand-drawn lines** look as if they were created with traditional media, such as paints, charcoal, or chalk. **Implied lines** are lines that don't really exist—like dotted or dashed lines, or the lines you create when you queue up for service at the grocery store. These implied lines are powerful tools for designers because individual things can feel unified or grouped together when they are aligned.

If you look at the leftmost line of the paragraphs on this page, you can see a “line” formed by the beginning of each line of type. Pay attention to the implied lines you create using the design elements in your artwork. Think of creative ways to use and suggest lines, and pay attention to what you might be saying with them. Make sure that the message all your lines send matches the intent of your work.

**Figure 8.14** The flow of lines conveys meaning based on their regularity.



**Figure 8.15** Style



## The Element of Shape

The next element needs little introduction: Shapes are boundaries created by closed lines. **Shape** can be defined as an area enclosed or defined by an outline. You're familiar with shapes such as circles, squares, and triangles, but there's a ton more to them than that (**Figure 8.16**). Those specific shapes are created in geometry, but what about the shape of a hand or cloud? These are shapes too, and you often use the same descriptors for shapes as you do for lines.

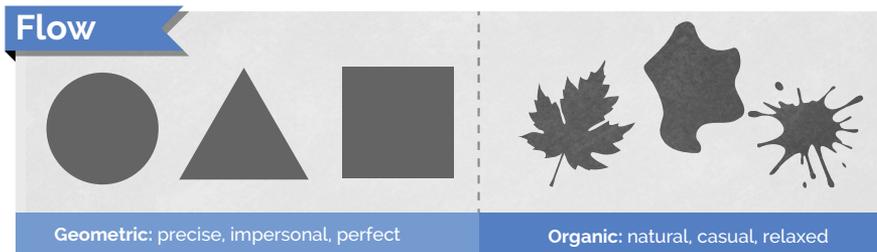
**Video 8.7** *Design School: The Element of Shape*



**Figure 8.16** Shape

### COMMON SHAPE DESCRIPTORS

The word “flow” describes how intentional a shape looks (**Figure 8.17**). **Geometric** shapes are more predictable and consistent, and they tend to feel more perfect, precise, calculated, and impersonal. Squares, circles, triangles, and stars are rarely found in nature and therefore convey more mechanical and manufactured impressions. **Organic** shapes often look random or generated by something natural. Organic shapes are usually asymmetric and generally convey a sense of natural, homemade, or relaxed feelings.

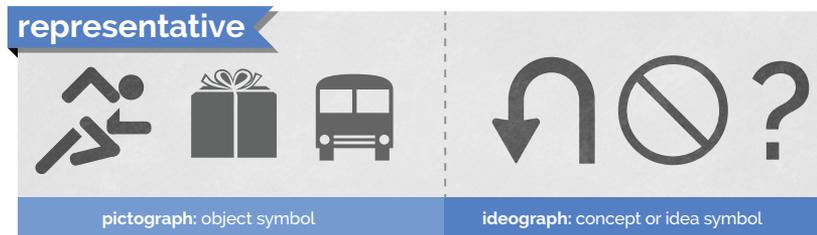


**Figure 8.17** How intentional is a shape?

## REPRESENTATIVE SHAPES

A **pictograph** (or pictogram) is a graphic symbol that represents something in the real world. Computer icons are pictographs that often suggest the function they represent (like a trash can icon to delete a file). Other examples of pictographs are the human silhouettes often used to indicate men's and women's restrooms. They're not *accurate* representations of the real objects, but they are *clear* representations of them. **Ideographs** (or ideograms) are images that represent an idea. A heart shape represents love, a lightning bolt represents electricity, and a question mark represents being puzzled. **Representative shapes** are very helpful in communicating across language barriers and can assist when designing for multicultural and multilingual audiences (**Figure 8.18**).

**Figure 8.18** Representative shapes



### LEVEL UP: SHAPING UP

This quest is designed to get you experimenting with shapes. I've provided the same list of words I used to inspire line drawings. (I'm not being lazy; I'm being intentional.)

- **Level 1:** Short, long, straight, wavy, zigzag, geometric, organic
- **Level 2:** Angry, lonely, worried, excited, overjoyed
- **Level 3:** Opposition, contrast, politics, infinity

Draw shapes that represent these words. Try to avoid any common representative graphics; that's too easy. Develop new shapes or just try to create shapes that seem to represent emotions or ideas that aren't normally communicated using shapes.

This is also a fun exercise to do with a friend. Have your friend pick a word for you to draw, or draw something and see how your friend describes it. You'll learn a lot about the different ways that people understand and express ideas. Remember that there are no wrong answers. Just explore, see what happens, and have fun with it.

Pictographs and ideographs are particularly important in web design because they can convey actions that may be taken or information that may be found with just a quick glance. Bootstrap CSS includes an entire library of these elements known as glyphs.

## The Element of Form

**Form** describes three-dimensional objects or, at least, objects that look three-dimensional (**Figure 8.19**). The best way to visualize form clearly is to consider that circles, squares, and triangles are shapes, whereas spheres, cubes, and pyramids are forms. Like shapes, forms are basically divided into geometric and organic types. Geometric forms, such as a cube, are pretty common. You are also familiar with organic forms such as the human form or the form of a peanut. When you work with 3D in applications, these forms are often referred to as “solids.”

**Video 8.8** *Design School: The Element of Form*



**Figure 8.19** Form

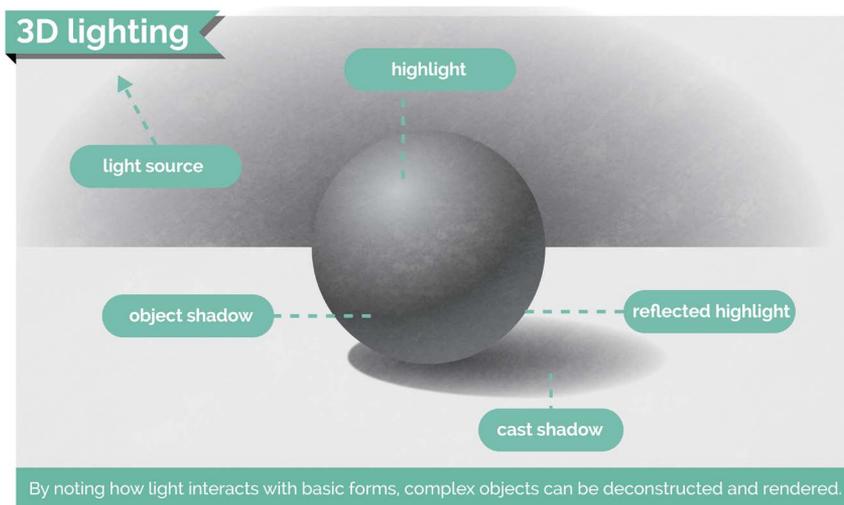
### 3D LIGHTING

In art and design, there's a special focus on the techniques that make images appear 3D in a 2D work of art. You have some standard elements to consider when you want to create a feeling of depth and form. **Figure 8.20** explains the standard elements of a 3D drawing that you need to consider.

- **Highlight** is the area of a form directly facing the light and appears lightest.
- **Object shadow** is the area of the form that is facing away from the light source and appears darkest.
- **Cast shadow** is the shadow cast on the ground and on any objects that are in the shadow of the form. One thing to remember is that shadows fade as they get farther from the form casting the shadow. Be sure to consider this as you're creating shadow effects in your art.

- **Light source** is the perceived location of the lighting in relation to the form.
- **Reflected highlight** is the area of the form that is lit by reflections from the ground or other objects in the scene. This particular element of drawing a 3D object is most often ignored, but it provides believable lighting on the object.

**Figure 8.20** Elements of 3D are created with light and shadows.



### LEVEL UP: FORMING UP

This is a standard art exercise. You might ask a friend in an art class or an art teacher to help with this one, or find a tutorial online. Draw a sphere with lighting and shadow on a flat plane. Imagine a cue ball on a white table. Be sure to include the elements of 3D mentioned previously and shown in Figure 8.20.

Your sketch might not be great. In fact, it may be horrible. That's *great!* If it's your first time and it looks better than you could do in kindergarten, you're doing amazing. Although you've got digital tools to help with this stuff, learning to draw and doodle activates a different part of your brain, and you want to wake up that part. I'm more concerned that you learn the ideas than that you master three-dimensional drawing on paper.

Focus on learning how things *look* and how to represent them in art and design. And look at some real 3D objects in the light. Put a tennis ball on the sidewalk and just stare at it. It's amazing how much of the real world you never pay attention to. Art is more about learning to see than learning to draw or create.

# The Elements of Texture and Pattern

**Texture** can describe an actual, tactile texture in real objects or the appearance of texture in a two-dimensional image (**Figure 8.21**). It is important to use texture to communicate feeling and authenticity in your art or designs. If you want to depict something elegant, soft, or comfortable, you could use a texture that resembles fabric or clouds. To represent strength or power, you might choose textures that represent stone or metal. And you could represent casual, informal, or nostalgic feelings using a texture that represents weathered wood or worn paint.

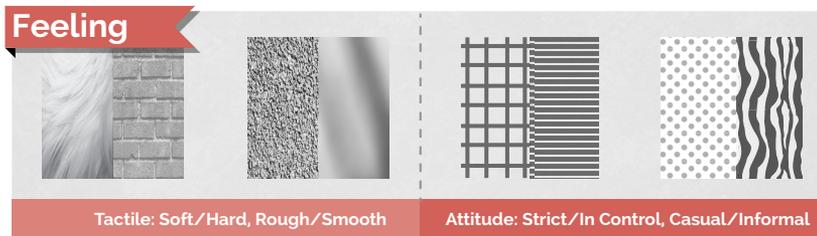
**Video 8.9**  
*Design School:  
The Elements of  
Texture and Pattern*



**Figure 8.21** Texture

**Pattern** can be defined as a repetitive sequence of colors, shapes, or values (**Figure 8.22**). Pattern is technically a different concept from texture, but in graphic design it's often regarded as the same thing. Let's face it: All the "textures" you will use when you create images to use in a web design project are just pixels. Any repetitive texture might also be considered a pattern. In the designs you created for the sample web design project in this book, you used patterns to define the look of the sidebar and header.

As you work more as a designer, you'll start to notice the nuances and subtle differences between textures and patterns. For now, just think of them as the visual qualities of your shapes and forms that can't be described by color or value alone.



**Figure 8.22** What feelings do textures convey?

## LEVEL UP: TEXTURE TANGLE

Texture is a valuable design tool. The goal of this exercise is to experiment with texture and patterns. Remember that there is no wrong answer or failure. The only way to miss a perfect score is to quit or not even make an attempt. It takes practice to create things the way you visualize them in your head, but that's the life of an artist. It's never perfect; it's just art.

- 1 Grab a sheet of paper and a fine-point pen.
- 2 Draw a line that crosses over itself at least once, forming a loop that goes from one edge of the paper to another. That loop is your first "texture pit."
- 3 In that loop, create a texture by just repeating a pattern.
- 4 Create another area and fill it with a different pattern or texture.

That's it. When your paper is full, you have leveled up.

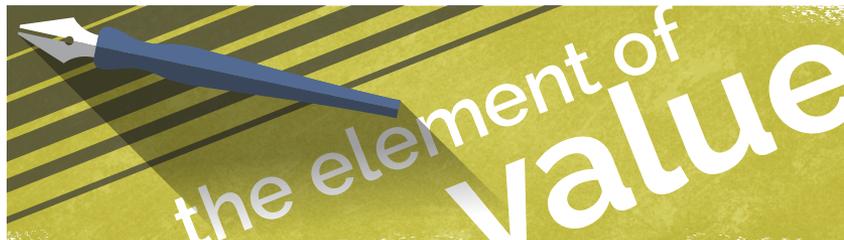
- **Level 1:** Use only geometric textures and patterns.
- **Level 2:** Attempt to create areas that look like textures of materials, plants, or animals. You might need to switch to pencil to be able to shade the areas well.
- **Level 3:** Introduce color. Try adding color to your designs, and see how that works.

## The Element of Value

▶ **Video 8.10** *Design School: The Element of Value*

**Value** describes the lightness or darkness of an object (**Figure 8.23**). Together with color, value represents the visible spectrum. Value is easily understood by thinking of a gradient that goes from black to white. But it is important to remember that value applies to color as well, and you can have a spectrum that fades from black, through a color, and then to white.

**Figure 8.23** Value of the visible spectrum



## LEVEL UP: VALUE VISION

Value is an interesting element to work with, and you've already experimented with it. Remember the "Forming Up" sidebar? That was a great exercise in controlling your pencil to draw a believable sphere. Honestly, there's no better exercise for learning to control your pencil and properly use value than to create 3D forms. But this experiment is going to take you a little further.

Pencils afford you the ability to create different values pretty easily, but what about a pen?

Create a set of values while experimenting with monochromatic (no gray) textures. Learn about hatching, cross-hatching, and stippling, all created using a fine-point marker. Use the thin-line Sharpies or an ultra-fine roller ball (.05 or less). Experiment with making different values using purely black and white duotone (no gray) methods.

- **Level 1:** Create three levels of gray across a spectrum.
- **Level 2:** Generate seven layers of gray across a gradient using mixed methods.
- **Level 3:** Generate a 3D sphere using purely duotone methods.

Professionals in the art and design industry often use the term *value*, but clients rarely use it. Clients will just ask you to lighten or darken a graphic or text, or sometimes use the words "tint," "shade," or "tone" in place of value (Figure 8.24). Technically, they're different, but many people use the terms tone, shade, tint, and value interchangeably. As always, when clients use a term and you're not exactly sure what they mean, ask for clarification. Sometimes they don't even know exactly what they mean, so asking ensures that everyone is on the same page.

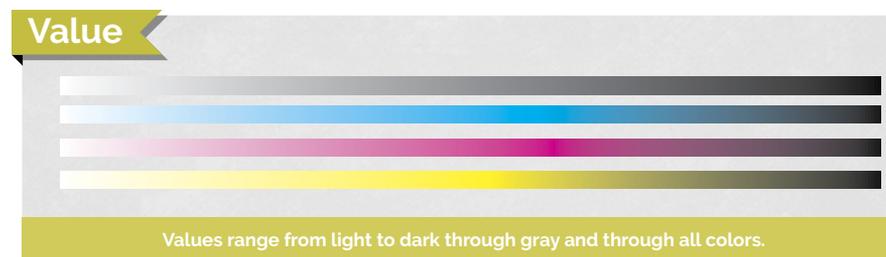
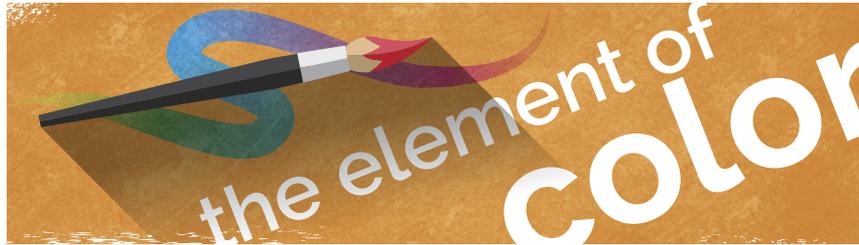


Figure 8.24 Value of the visible spectrum

## The Element of Color

Color is really a study in itself (**Figure 8.25**). And when you think about it, it's actually hard to define. How would you define color without using examples of colors? Check out its definition in a dictionary and you'll find that defining color really doesn't help you understand it. Color is best *experienced* and *explored*.

**Figure 8.25** Color



We have so many ways to think about color and so many concepts to dig into that exploring color is a lifelong pursuit for most artists and designers. Color theory is a deep and complex study. We'll explore the basics here, but remember that this is just the on-ramp toward understanding color. Grasp these concepts and you'll still have a ton more to learn.

Color psychology is a relatively new discipline and an interesting study in itself: understanding the emotional and behavioral effects that colors can have on people. The colors you choose really do matter. For the purposes in this book, **color** is the perceived hue, lightness, and saturation of an object or light.

### HOW COLOR WORKS

Color is created in two different ways: additive color and subtractive color (**Figure 8.26**). When you work in an image editor, such as Photoshop, where some projects might be intended for print while others need to be published to the web, you will need to choose the correct color mode.

- **Additive color** is created by combining light. This is how your monitor works, and the most common color mode for additive color is RGB. The letters RGB stand for red, green, blue, which are the three colors used to create digital images. Monitors and electronic devices are dark when turned off, and you create colors by adding light to the screen. In the images you insert in Dreamweaver and the color values used in HTML and CSS, colors are indicated using RGB values and may be referenced by either their hexadecimal value—#FOFOFO—or in the RGBa format—`rgba(240,240,240,1.00)`.



**Figure 8.26** Subtractive and additive color

- **Subtractive color** is created by subtracting light. This is the color system you learned in early art classes. In it, red, blue, and yellow are the primary colors. For print, you use CMYK, which is cyan, magenta, yellow, and black. (The black is added due to impurities in ink and is required for the theoretical combination of cyan, magenta, and yellow to create all the colors in reality.) You start with a white surface (the paper) that reflects all the light back to us. Then you subtract color by using paints or inks that limit the light that is reflected back to the viewer.

## THE COLOR WHEEL

Sir Isaac Newton invented the color wheel in the mid-17th century (**Figure 8.27** on the next page). The color wheel offers a way to display and build all the colors possible using paint. It's a common exercise in beginning art classes to create a color wheel when learning to mix and experiment with colors. In digital imaging, it's not as important to go through this exercise, but if you have a chance, give it a try. It's kind of interesting to see how all the colors can mix to obtain the colors and shades you can perceive.

The color wheel is important because many color theories are named after their relative positions on the color wheel. They're a lot easier to remember using the color wheel to illustrate.

The first thing to realize is that some colors are classified as **primary colors**. These colors can be combined to create every other color in the visible spectrum (**Figure 8.28** on the next page). For subtractive color, used in traditional art, the primary colors are red, blue, and yellow. For additive colors, the primary colors are red, blue, and green.

The colors that are created when you combine primary colors are known as **secondary colors** because you can create them only by applying a second step of mixing the colors. When you mix secondary and primary colors, you get another set of colors, known as **tertiary colors**.

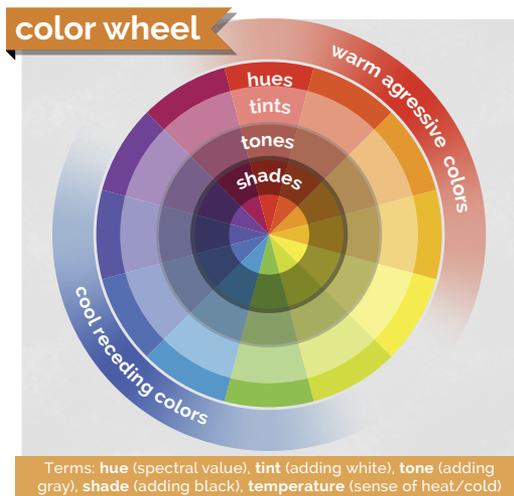


Figure 8.27 The color wheel

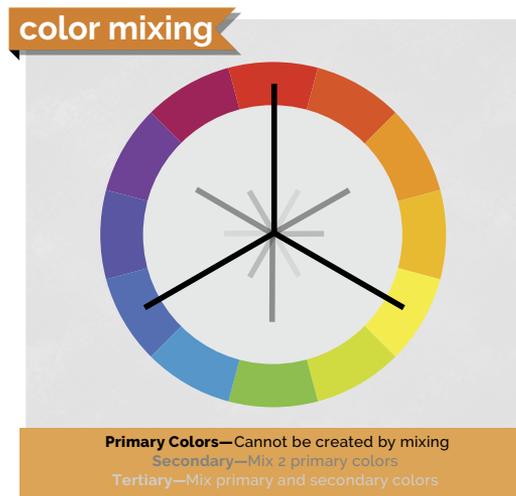


Figure 8.28 Color mixing

### LEVEL UP: ROLLIN' WITH COLOR

Creating a color wheel is a staple project in art classes. Any experienced traditional artist or art teacher can help you with this task. You can find a ton of information about this online, but this is a valuable project to help you get in touch with color and understand how colors are related.

- **Level 1:** Create a traditional color wheel using paint on paper.
- **Level 2:** Use Adobe's free Color CC tool to create color wheels of your own. You'll see many of the terms used here in this online tool, found at [color.adobe.com/create/color-wheel](http://color.adobe.com/create/color-wheel).

## BASIC COLOR RULES

Color rules (also known as **color harmonies**) are a great way to start picking colors for your projects in Dreamweaver and are prominently featured if you use the Adobe Color CC tool. The color rules are named for their relative locations on the color wheel. When you're choosing colors and exploring other ways to join colors to create contrast or harmony, these color rules are where you should begin. Let's cover some of the basic color rules and the impressions they tend to communicate.

The three most common ways of thinking about colors (and, really, the basis of all color rules) are monochromatic, analogous, and complementary color schemes (Figure 8.29).

- **Monochromatic** colors, as you’ve probably guessed, are based on different shades and tints of the same color. They tend to communicate a very relaxed and peaceful feeling, and you’ll create very little contrast or energy in art using these colors.
- **Analogous** colors are side by side on the color wheel, and they tend to create gentle and relaxing color schemes. If you want to add a little more variation while maintaining a calm feeling, consider analogous colors. Analogous colors don’t usually stand out from each other; they seem to work together and can almost disappear together when overlaid.
- **Complementary** colors are opposite each other on the color wheel. Complementary color combinations are high in contrast and normally very vibrant, so use with caution. When overused, complementary color can be very “loud” and can easily cross over into visually obnoxious if you’re not careful. You can remember this rule with the alliterative phrase “Complementary colors create contrast.”



Figure 8.29 Basic color rules

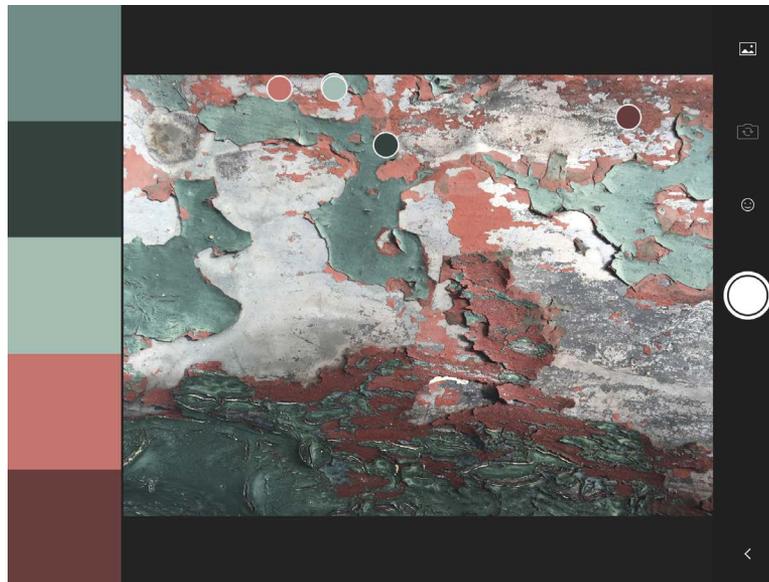
You can explore a bunch of color combinations on the Adobe Color CC website (Figure 8.30 on the next page). It’s an amazing way to browse other people’s color collections, or grab colors from an image you like and create a color set to use for your projects. When you register with your free Adobe ID, you can save the color themes you find or create and bring them into your Adobe apps to use and share with others. Adobe Color CC is also part of a free app for iOS and Android. Open the Adobe Capture CC app, and choose COLORS. Then open an image or point your phone camera at something to create a color theme (Figure 8.31). Experiment with creating custom themes from your favorite blanket, a sunset, your goldfish, or your crazy uncle’s tie-dyed concert shirt from Woodstock.

Creating color swatches for use in a web design project is an important way that you can communicate your design intentions with your client and ensure that the colors throughout the site are harmonious and consistent.

**Figure 8.30** Use the Adobe Color CC website to explore color harmonies and save combinations to your Creative Cloud account.



**Figure 8.31** The Adobe Capture CC app lets you capture colors from photos you've taken or grab them live using the camera.



## COLOR ASSOCIATIONS

We tend to associate different things with different colors. Whether this is learned behavior or instinct can be debated, but as a designer it's important to recognize and properly use the right color for the right message. Again, this isn't a science, and you'll need to consider more than the chart in **Figure 8.32** when developing your color-picking strategies. But understanding colors and their associations can be useful to produce the right “feel” in your images. You can evoke interesting feelings and contrasts by capitalizing on these associations in your design work. You saw this theory in action when you selected the colors that were used in the design for the *wulvs.com* website.

## The Element of Type

Type is generally not considered a traditional element of art, but it can be a critical part of designers' work. Typefaces carry a lot of emotional meaning, and choosing the proper typefaces for a job is a critical skill that every designer needs (yet many don't have). Like color, this area of design is so deep and has so many aspects that entire books and college courses are focused only on typography (**Figure 8.33**). You can't reduce it to simple rules such as “Use three typefaces, maximum,” “When in doubt, use Helvetica,” or “Never use Chiller.” (Even though that's not a bad starting place.)

Working with type is often like being a marriage counselor or a matchmaker. Every typeface has a personality, and you need to carefully match the type in your design work with the overall feeling of the piece. The typefaces you use also need to work together and not clash with one another. It takes time and experience to master this balance, but over time you can become a skillful typeface matchmaker. Most artists have a few fonts they tend to lean on heavily, and that's okay—especially at the beginning.



| associations  |                         |
|---|-------------------------|
|  | sky, water, peace, cold |
|  | nature, growth, money   |
|  | sunlight, joy, energy   |
|  | fire, warmth, attention |
|  | blood, passion, danger  |
|  | flowers, creativity     |

Color associations can depend on: culture, profession, or experience.

**Figure 8.32** What messages are your color choices sending?

★ *ACA Objective 2.6*

▶ **Video 8.12** *Design School: The Element of Type*

**Figure 8.33** What are your typeface choices communicating?

**TIP**

Check online at [brainbuffet.com/design/typography](http://brainbuffet.com/design/typography) for a list of fonts and typography-based resources to help you dig a little deeper with this concept.

**LEVEL UP: PUZZLING TYPOGRAPHY**

A popular jigsaw puzzle style right now (often available in bookstores) has a strong typographic focus that lists “family rules” or “life lessons” in various fonts. However, any puzzle with lots of larger text in multiple typefaces will work great. Get one of these puzzles, and while you’re putting it together, notice the subtle differences between fonts and the letterforms of each font. It’s a great way to learn to pay attention to the subtle differences between the various styles of type.

- **Level 1:** Do a puzzle like this with a group of friends and discuss your observations.
- **Level 2:** Do a puzzle like this by yourself.

## TYPOGRAPHY

Typography is the art of using letterforms and type arrangement to help the language communicate a message. As mentioned earlier, type can be its own topic of study and can easily become overwhelming. I’ll skim the surface here, but I hope you’ll dig in a little more as you move forward in your design. Let’s get some vocabulary out of the way first (**Figure 8.34**).

Technically, a **typeface** is the specific letterform set that makes up the type. Helvetica, Arial, Garamond, and Chiller are examples of typefaces. In brief, it’s the “look” of the letters. **Fonts** are the whole collection of the typeface in each of its sizes and styles. So 12 pt Arial Narrow and 12 pt Arial Bold are different *fonts* of the same *typeface*. The same is true for two different sizes of the same typeface.



**Figure 8.34** Many of today’s typography terms were created in the days of movable type and have changed as publishing technology evolved.

From now on, let's use the term *font*, because that's also the name of the file you'll be installing when you add typefaces to your computer, and it's the term that is used in Dreamweaver, HTML, and CSS.

## TYPE CLASSIFICATIONS

Typefaces can be classified in many ways, but this chapter uses the Adobe Typekit classifications (Figure 8.35). Generally, people divide fonts into two main categories that should be used for large areas of type: serif and sans serif.

These two basic classifications of fonts are used in describing fonts in CSS and HTML. Recall from the projects that you completed that for web pages a font family is used to define a series of fonts displayed based on whether viewers have a particular font installed on their computer.

- **Serif fonts** are often associated with typewritten documents and most printed books. Generally, serif fonts are considered to be easier to read in larger paragraphs of text. Because so many books use serif fonts and early typewriters produced them, serif text often feels a bit more traditional, intelligent, and classy.
- **Sans serif fonts** do not include serifs. “Sans” is a French word that migrated to English and simply means “without.” Sans serif fonts are often used for headlines and titles for their strong, stable, modern feel. Sans serif fonts are also preferred for large areas of text for reading on websites and screens.

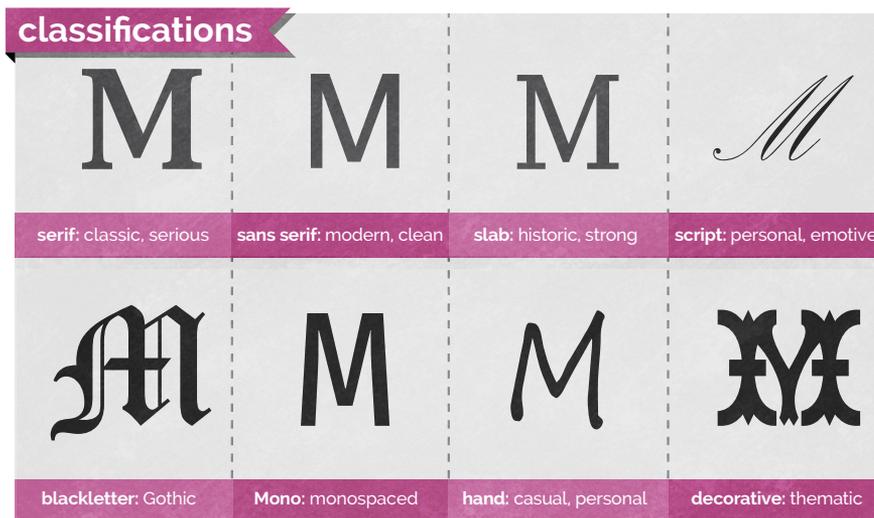


Figure 8.35 Adobe Typekit classifications

Beyond these basic types of text, designers do use other typefaces that are not appropriate for large areas of type because they are not easy to read in a long paragraph. Most designers consider the following fonts to be “decorative” for that reason.

These kinds of fonts are available for use in web design only when a web font is used and are typically used only for labels, section headers, and logos. Dreamweaver uses similar font categories as these in the Manage Fonts window where the Adobe Edge Web Fonts are listed.

- **Slab serif fonts** (also called Egyptian, block serif, or square serif) are a more squared-off version of a typical serif font. These fonts bridge the gap between serif and sans serif fonts and generally feel a bit more machine-built. The very simple design tends to make them feel a bit rougher than their serif counterparts.
- **Script fonts** (often known as formal) have a very elegant feeling. These fonts are great to use for invitations to formal events, such as weddings, and in designs where you want to convey a feeling of beauty, grace, or feminine dignity. If you were designing for a spa, a beauty shop, or for products or services, script fonts would carry a feeling of relaxed and elegant beauty.
- **Blackletter fonts** (also known as old English, Gothic, or Textura) feature an overly ornate style and are often used to title formal documents, such as certificates, diplomas, or degrees, as well as old German Bibles and heavy metal bands. It conveys a feeling of rich and sophisticated gravitas, often hinting at a long history of tradition and reliability.
- **Monospaced fonts** (also known as fixed-width or non-proportional) use the same amount of horizontal space for each letter. Typically, fonts use a variable spacing technique known as **kerning**. (You’ll learn more about kerning later.) Monospaced fonts, in contrast, use the same width for every character. Monospaced fonts are good to use when you’re trying to make something look impersonal, machine-generated, or retro-geeky, because typewriters and early computers used monospaced fonts.
- **Handwritten fonts** (also known as hand fonts) simulate handwriting. They are popular for adding a personalized, casual, or human touch to your designs and are often used on junk mail to try to trick you into opening that “special limited-time offer just for you!” (Don’t fall for it.) Handwritten fonts are great for communicating casual and friendly feelings, but they can be tough to read in a larger block of text.

**TIP**

Want to create a font based on your own handwriting? Go to [www.myscriptfont.com](http://www.myscriptfont.com) to do so for free.

- **Decorative fonts** (also known as ornamental, novelty, or display) don't fall into any of the other categories. They also tend to convey really specific feelings. Decorative fonts should be used sparingly and very intentionally. Never use a novelty font just because you think it looks cool. That's a typical newbie move. Make sure that you are striving to convey something very specific when using decorative fonts.
- **Dingbat fonts** (also known as wingdings) are a special type of font that doesn't have an alphabet; they have a collection of shapes or objects instead of letters.

**NOTE**

Each character of a font, whether it's a letter, number, symbol, or swash, is called a **glyph**.

## TYPE TALK

You'll need to learn a bunch of jargon concerning type when working in the design industry. Some of it is commonly used in discussion about design, and some of it will help you discuss fonts when trying to find the perfect typeface for your design. I'll just cover the basics here, mentioning the easily accessible features in the Photoshop interface as well as mentioning ideas you should know to help improve the typography in your designs.

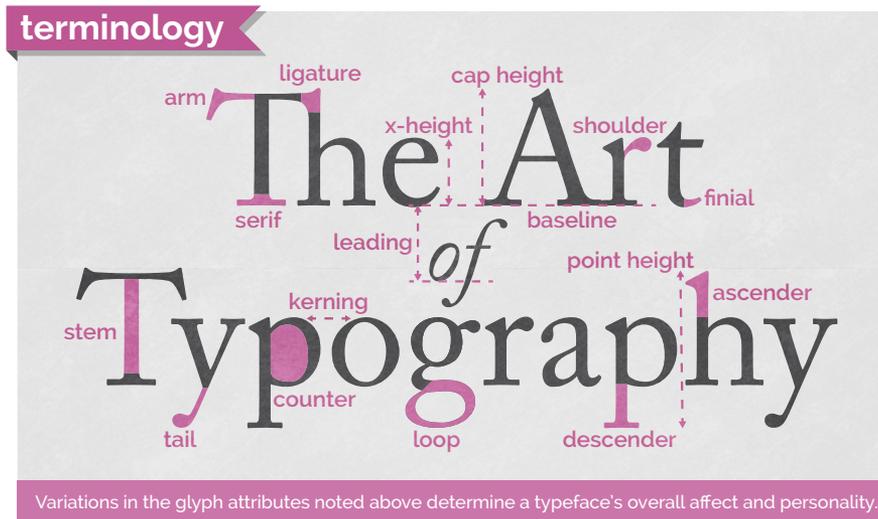
**NOTE**

You can find additional resources at the growing collection of typographical resources at [brainbuffet.com/design/typography](http://brainbuffet.com/design/typography).

## TYPOGRAPHIC ANATOMY

Figure 8.36, on the next page, illustrates many of the anatomical terms of typography. Software has no settings for these options, so I won't go into detail about them. But when you start to study typography and learn these terms, you will more easily discern the differences between typefaces. It's easiest to understand these terms by simply looking at Figure 8.36. You will hear these terms in the industry, and when you're looking for a specific font, they'll allow you to more easily describe what you're looking for. Let's face it, you can't get too far professionally if you're always using words like "doohickey" and "little hangy-downy thingy." You need to use descriptive terms like ascenders and descenders; anthropomorphic terms like arms, shoulders, and tails; or architectural terms like counters and finials.

**Figure 8.36** The terms of typography



## THE HOLY TRINITY OF TYPOGRAPHY

Mike Skocko is an inspiring teacher and a brilliant designer with a very informative website, at [maclab.guhsd.net](http://maclab.guhsd.net), that features some amazing tutorials, designs, inspirations, and student art.

One video Mike shares explains three concepts of typography: kerning, leading, and tracking, the “holy trinity of design.” As a designer, you really need to master them. The following terms (**Figure 8.37**) affect the ways that letters are spaced from each other vertically or horizontally.

- **Kerning** is the space between specific letter pairs. For example, the first two letters in the word “Too” are closer than the first two letters in the word “The” because the letter “o” can kind of tuck under the crossbar of the “T.” A quality font file will have a good set of kerning pairs for specific letter combinations, but for some professional work (and with poorly designed fonts) you might need to get in there and tweak the kerning. Adjusting the kerning between specific letters can help you perfect your type presentation in logos and headlines.
- **Tracking** is the overall space between all the letters in a block of text. It allows you to compress or expand the space between the letters as a whole, rather than just between specific pairs, as you do with kerning. Adjusting tracking can greatly affect the feeling that text conveys. Experiment with tracking to help create various feelings in headlines and titles.



Figure 8.37 The Holy Trinity of Design

- **Leading** is the amount of space between the **baselines** of two lines of text. The baseline is the imaginary line that text sits on. Unlike with word processing applications, which let you choose single- or double-spaced, you can actually set the specific distance between the lines of paragraph text. Doing so can create great-looking space within a design or even overlap with your paragraph text. Experiment with and explore this option in your work. You can really change the mood of text by adjusting the leading of a paragraph.

These values are all important when working in a purely graphical design context, but on the web things become more complicated. Because of the fluid nature of web pages and the ability that the viewer has to zoom in or out on the contents of the page—or view them at all sorts of different dimensions—some of these settings simply don't apply. You can, for instance, set a letter spacing quality in CSS that would create a set pixel value to space between characters, but what happens when the viewer increases the magnification of the page? That value becomes irrelevant.

The one CSS property that does directly relate to these values is line height, which is the functional equivalent of leading. By setting a specific line height value, you can create the same kind of spacing between lines of text that you would by adjusting the leading in a graphics program.

## SIZE, SCALE, AND SMALL CAPS

The following terms are available in the character panel of the interface for most Adobe applications and may be used when creating logos and other artwork for use in your web design projects (Figure 8.38).

- **Type size** is traditionally the height from the highest ascender to the lowest descender in a font, expressed in points (1/72 of an inch). Today, it's more a guideline than a firm definition, so most designers set the appropriate type size by eye. Different fonts of the same point size can appear to be much different in physical size if the ascenders and descenders are different from each

other. Web designers and others who design for screen presentations typically use pixels, as opposed to points, to define type size.

- **Vertical and horizontal scale** describe the function of stretching letters and distorting the typeface geometry. Because they distort the typeface, use them with caution. They should be used only when you are trying to express a very specific feeling and should not be used in blocks of type because readability can suffer with either of these adjustments.

**TIP**

Higher-end typefaces may provide a small-caps face that is much nicer than the setting in software. If small caps are critical for your design, look for a font that supports this feature.

- **All caps and small caps** are similar in that they both use only the uppercase letterforms for each letter, but ALL CAPS makes all the letters the same size, whereas SMALL CAPS sets the letters that would normally be capitalized at a larger size. Small caps tend to increase readability over all caps, but both cap formats should be avoided in large blocks of text because they are more difficult to read than standard text.

You can use font-variant and text-transform settings in CSS to achieve similar results in HTML text.

- **Ligatures and swashes** are special alternative settings offered with some fonts to combine letters or add special stylized touches to specific letter combinations or letters. For example, when the “Th” combination touches in a headline, you can replace it with a single ligature that looks much better. Swashes add flowing and elegant endings to letters with ascenders and descenders. Both of these are normally reserved for type that expresses an especially elegant or artistic feel.

**NOTE** The settings for ligatures and swashes are available not in the Character panel but in the OpenType panel of most Adobe graphics apps. These rarely used features can really give some creative expression with fonts that support it. There is no equivalent to these settings in HTML or CSS.

Figure 8.38 Character-level specifics



## LEVEL UP: GAMES OF A CERTAIN TYPE

You can find a lot of great online resources for exploring typography, and you can also find some that are not that great but are worth visiting at least once or twice. A pretty good list of typographic resources is maintained online at [brainbuffet.com/design/typography](http://brainbuffet.com/design/typography). Take the time to explore some of them to dig a little deeper into your understanding of type and typography.

- **Level 1:** Experiment with at least three typographic games online.
- **Level 2:** Achieve a high score on at least one typographic game online.
- **Level 3:** Achieve a high score on five or more typographic games online.

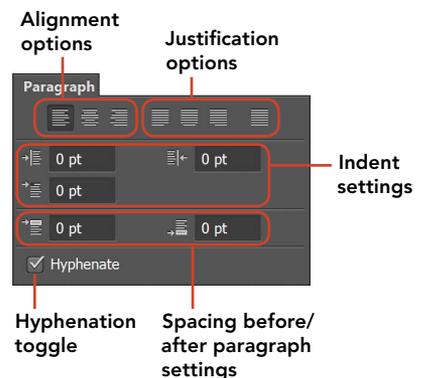
## PARAGRAPH SETTINGS

**Paragraph settings** affect an entire paragraph rather than selected words. These options adjust the **alignment** of the paragraph: left, centered, or right. **Justified** text aligns a straight edge on both edges of the paragraph, with the ability to dictate how the last line is aligned. **Indent** settings let you choose how far the entire paragraph is indented on each side or in just its first line. **Paragraph spacing** settings are similar to leading but apply to paragraphs—instead of lines of type within them—and you can also set the space above or below paragraphs. **Hyphenation** allows you to determine if and when words should be split with hyphenation.

The settings you see in **Figure 8.39** are those used in Photoshop, but similar kinds of alignments can be achieved in Dreamweaver using CSS text properties for `text-align` and `text-indent`.

## Wrapping Up the Elements

As you've seen in this chapter, the elements are the building blocks or the raw materials of design. But what turns these elements into art is applying the principles to the way you arrange these elements on your workspace. In the next section, you'll explore the principles, which are a framework that help you arrange your work in an artistic way.



**Figure 8.39** The icons for the paragraph settings are quite helpful in illustrating the function of each setting.

# The Principles of Design

▶ **Video 8.13**  
*Design School:  
The Principles of  
Design*

Much like the elements of design, different artists and schools of thought will generate different ideas about what makes up the principles of design (Figure 8.40). As a young artist, this can be so frustrating. Sometimes you just want someone to tell you the “answer.”

**Figure 8.40** The principles of design



But after really understanding the principles, you'll appreciate that no one can point to a universal list of artistic principles. If there's no correct answer, there's also no *wrong* answer either. Creatively, you always have another way to approach your art, and becoming an artist doesn't mean that you learn to see any *one* approach. Becoming an artist means that you learn to *see*.

Until you really explore design principles, you will not be ready to understand them. Therefore, definitions are not that helpful as hands-on work. But truly grasping the principles and experiencing them is the only way to grow as an artist. They're not something you understand; they're something you *undertake*. This is the beginning of a lifelong exploration of beauty and creativity.

The bottom line is this: Don't get hung up on names or descriptions. Try to engage with each idea as a loosely formed concept that remains fluid and flexible rather than "defining boundaries." The fact is, by studying principles you're trying to do just the opposite—moving in directions that have no boundaries. The goal of all good art and design is to explore new ways to use the elements and principles, and not to repeat what's been done in the past. You examine these principles to start your understanding of something, not to limit it.

## The Principle of Emphasis or Focal Point

**Emphasis** describes the **focal point** to which the eye is naturally and initially drawn in a design (**Figure 8.41**). Some art has a focus that's really obvious. Most marketing and advertising is that way. Other art invites you to step in and explore. It might encourage an exploration of color or texture, but it has no specific point other than the color or texture.

▶ **Video 8.14**  
Design School:  
The Principle of  
Emphasis



**Figure 8.41** Emphasis creates the focal point—a place where the eye is naturally drawn when encountering the image. Use this to your advantage when designing.

Let's think of emphasis as the main point or primary idea in a piece of art. You can move the viewer's eye to a specific point in the design by making something *different*. You can easily find examples right on the pages of this book. Chapter headings are larger than the rest of the type. Glossary terms are a different color. Even the simple process of *italicizing* words or putting them in **boldface** make text stand out. The human eye is naturally drawn to unique things. So if you want to make something stand out, make it different.

Careful use of contrast is critical to master because a typical design newbie tries to make everything special and unique. As a result, nothing stands out. The design looks *random*, not cool. Give a piece some unity, and it will feel right. You can then emphasize what's truly important. Which brings us to our next principle...

## The Principle of Contrast

▶ **Video 8.15**  
Design School:  
The Principle of  
Contrast

**Contrast** generally creates visual interest and a focal point in a composition. If you think about it, a blank canvas is a canvas with no contrast. As soon as you begin to alter the surface and create contrast, you also start to create a focus. The principle of contrast can be defined as a difference in the qualities of the elements within an image. To use contrast is to create something different from the surrounding pieces of the composition.

Many artists limit their understanding of contrast by simply confining it to color or value. But contrast is much more than that. Any difference between one thing and another is a contrast. You can have contrast in size, texture, value, color, or any of the defining characteristics you learned about in exploring the elements.

The most important thing to remember about contrast is that all contrast creates some emphasis, and if you have too much emphasis you will have no focal point.

## The Principle of Unity

▶ **Video 8.16**  
Design School:  
The Principle of  
Unity

**Unity** generally communicates calm, peaceful, or cool feelings within your art. The principle of unity (also known as harmony) requires that the things that go together should *look* like they belong together. The elements in your design should feel like a family. This doesn't necessarily mean that everything needs to be the same, just that they should share some similar traits. When you have no unity, you can have no focal point and, therefore, no emphasis (**Figure 8.42**).



**Figure 8.42** The similar lines, colors, and even values of this image give it a very unified feeling.

Note the headings of the different sections in this chapter. They make it really easy to find the content that you are looking for when skimming through the book. Even the breaks between paragraphs help distinguish one concept from another. But all the words in this paragraph belong together because they share a unity of typeface, spacing, color, and so on.

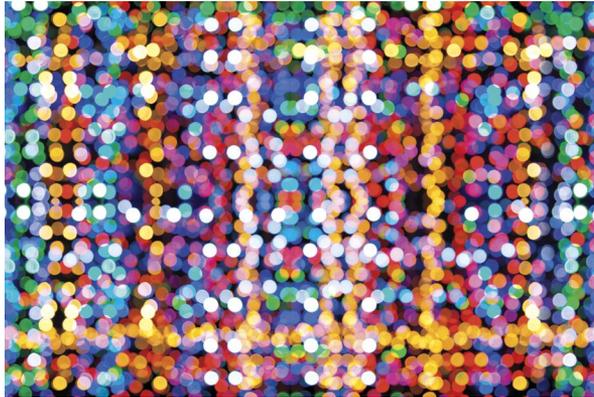
Unity is very important in your compositions so that when you create contrast it draws the viewer's eye where you want it to go. In design, more than in art, you are interested in guiding a viewer's perceptions. Design generally tends to be much more intentional than art. Both are very important, however, and beginning with design can make your experimentation with artistic elements and principles much more effective and productive.

Unity is particularly important in web design, as viewers of a website will routinely move from one page to another, and having a predictable and consistent look to the elements of a site, such as the navigation, helps maintain the viewer's focus.

## The Principle of Variety

**Variety** tends to communicate energy, heat, and high emotion in a design. When applying the principle of variety, you use different elements in an image to create visual interest. In many ways, it is the exact opposite of unity. You can think of unity and variety as being at the opposite ends of contrast. Unity is establishing a low degree of contrast in a composition (**Figure 8.43**). Variety does the opposite and brings a higher amount of contrast to the composition.

 **Video 8.17**  
*Design School: The Principle of Variety*



**Figure 8.43** When everything is different in an image, like these colored lights are, nothing stands out.

Variety is a principle that normally needs to be used sparingly. Too much variety quickly moves your art from interesting to chaotic and disorganized. Beginning artists and designers sometimes have a hard time properly using variety. They get a little bit carried away and lose all sense of focus or unity. Beware of that tendency. Nothing shouts “newbie” like chaos. Here’s another Cavanaugh-ism to help you tone down the newbie impulse to go crazy with variety:

“Just because you can, doesn’t mean you should.”

## The Principle of Balance

**Video 8.18**  
*Design School:  
The Principle of  
Balance*

### TIP

*Many artists and designers characterize pieces that are out of balance as “leaning to the left,” “leaning to the right,” or “top heavy.” Over time, you’ll develop these feelings too, and you’ll be able to spot it when art seems like it might physically tip over.*

**Balance** suggests that the arrangement of elements in an image should be evenly distributed. This is not to say that everything should be centered or that placing something in the top right means you should mirror it with something similar in the top left. That’s how a non-designer lays out a composition. An experienced artist learns to properly balance all the elements—including space—in their compositions.

- **Symmetrical** balance is what most students latch on to at first. It occurs when you can divide an image along its middle and the left side of the image is a mirror image of the right (or the top reflects the bottom). Using a seesaw analogy, a symmetrical balance would have two equally sized people equally distant from the fulcrum. This is the easiest balance to execute, but it conveys a very intentional, formal, and mechanical feeling.
- **Asymmetrical** balance achieves balance with different elements on each side (or the top and bottom) of an image. Imagine an adult on a seesaw with a child. They can balance, but only if the adult is closer to the fulcrum and the child is farther away. To achieve asymmetrical balance, you need to use space to counterbalance the different weights on each side (**Figure 8.44**).

- **Radial** balance is a circular type of balance that radiates from the center instead of the middle of a design (**Figure 8.45**). Many artists get the feeling that they're viewing a radially balanced image from above. This kind of balance is almost always circular. An excellent example of radial balance is a kaleidoscopic image, which can feel very balanced and unified but also typically feels more static than the other types of balance.



**Figure 8.44** This image, though asymmetrical, is well balanced. The bright sun is offset by the visual weight of the rocks in the lower right.



**Figure 8.45** This image has a radial balance.

## The Principle of Proportion or Scale

▶ **Video 8.19**  
*Design School:  
The Principle of  
Proportion or Scale*

**Proportion**, sometimes called **scale**, describes the relative sizes and scale of things. If you've ever seen a drawing and observed that "the head is too small" or "the body is too fat," you're evaluating the proportion. It's simply the sense that things seem to be the proper size relative to each other.

You can manipulate proportion and scale to create emphasis. Things that are larger than they should be appear stronger, more important, or more powerful. Take the chapter headings in this book. By making the headings disproportionately large, it indicates importance and emphasis. You can also reduce the perceived value, strength, or importance of something by reducing its scale.

## The Principles of Repetition and Pattern

▶ **Video 8.20**  
*Design School:  
The Principles of  
Repetition and  
Pattern*

The principles of repetition and pattern—along with movement and rhythm—seem to be the most confusing and difficult to grasp. As a matter of fact, some artists and writers more readily connect pattern and repetition with rhythm than with movement. Let's think about the simplest and most concrete uses of them here. As always, you are encouraged to explore these areas much more deeply on your own.

- **Repetition** is a principle that is pretty easy to grasp: repeating an element in your design. Repetition can convey many things, but it often represents importance, movement, or energy. Think of a cartoon illustration of a ball moving or a cartoon of a bird flapping its wings. Just repeating a few so-called "motion" lines can convey a sense of movement.

Another important way to think about repetition is to share certain traits of the elements in your design. Doing so brings some unity to those elements and lets the viewer know they're related. For example, you could repeat the colors of the logo in headers and bold type or repeat the same colors from a photo into the header text of an associated article in a magazine. Repetition of this kind conveys a sense of unity in the design.

- **Pattern** happens when different objects repeat in a sequence. The best way to think about the difference between repetition and pattern is that repetition is done with a single element (such as pinstripes on a suit), whereas pattern happens with a collection of elements (such as a floral pattern on fabric). Pattern also happens when elements are repeated consistently and, at some point, eventually move past repetition and become a pattern.

## The Principles of Movement and Rhythm

Movement and rhythm are similar, much like repetition and pattern. They modify an image much the way an adverb modifies an adjective. They explain a feeling that the elements create, rather than making specific changes to the elements.

- **Movement** refers to the visual movement within an image. Depending on the context, it can refer to the movement the eye naturally follows across an image as it moves from focal point to focal point, or to the perceived movement or flow of the elements in the image.

Movement can also refer to the “flow” of an image, and it is a critical principle to consider in your work. It’s more of a feeling than a concrete visual aspect of your design. When something “doesn’t feel right,” it may be that the flow is uncertain or contradictory. If you haven’t considered the flow of your design, analyze it and create a flow that guides the viewer through your design. The more linear your flow, the more your design will convey a sense of reliability, consistency, and calm. A more complex flow can convey a sense of creativity, freedom, or even chaos.

- **Rhythm** refers to the visual “beat” within the design, a sense of an irregular but predictable pattern. Just like a rhythm laid down by a drummer, a design’s rhythm is creative and expressive rather than a consistent pattern or repetition. Depending on the design you’re working on, rhythm may or may not be a critical principle to consider. A predictable rhythm can convey a sense of calm and consistency, whereas an erratic or complex rhythm can convey a sense of urgency or energy.

These two principles tend to be the most subjective, so be sure to clarify anything you’re unsure about when discussing these principles with a client. Because they’re more of a feeling than a specific element or technique, it takes a little experience to get a handle on them. But these are just like all the principles: Once you start looking for them, you’ll start to see them.

 **Video 8.21**  
*Design School:  
The Principles of  
Movement and  
Rhythm*

## Conclusion

▶ **Video 8.22** *Design School: Wrapping Up Design School*

It's critical that you as a designer understand the elements and principles so that your work can communicate clearly with your audience. Learning the technical basics of web design is fairly easy, and they can be mastered by people who will never "make it" in the industry. It's not enough that you know (technically) how to compose things in a web design if you don't (artistically) know when and where to design them.

Take the time to develop your design skillset and to remember that it *is* a skillset. You've got to practice and continue to develop your craft to level up those skills. Using Dreamweaver without having a design sense is like starting up a jet engine without the jet. It's powerful, but you won't really be able to fly it to a destination. The artistic elements and principles are the wings and controls that let you harness that power and get your art to go where you want it.

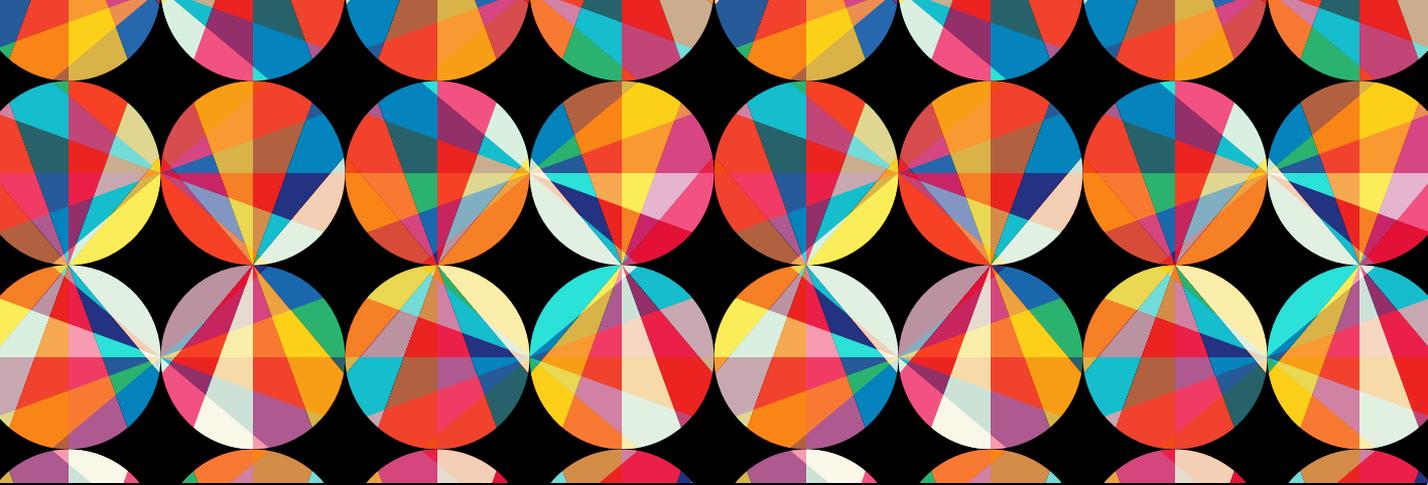
### TIP

Connect at [brainbuffet.com/design](http://brainbuffet.com/design) to see an ever-growing collection of design-related resources, or follow [@brainbuffet](https://twitter.com/brainbuffet) on Twitter for weekly resources, inspiration, and freebies to use with your design projects.

Invest the time and effort to practice and refine your design sense. You'll find not only that your personal art grows more impressive over time, but that the entire world opens up and becomes much more interesting, beautiful, and detailed.

Dreamweaver is about learning to create visual designs with exceptionally high quality and detail, but becoming an artist and refining your skills is about learning to *see*.

Open your eyes... and enjoy the beauty you've been missing that's all around you.



## CHAPTER OBJECTIVES

### Chapter Learning Objectives

- Learn how to create design work for others and understand their needs and purposes.
- Understand copyright concerns related to the design industry.
- Understand project management concepts and issues related to working as part of a design team.
- Be aware of common problems and pitfalls in project design and in the design industry.
- Learn the common elements that are included in a typical web authoring project plan.

### Chapter ACA Objectives

#### **DOMAIN 1.0**

#### **SETTING PROJECT REQUIREMENTS**

- 1.1** Identify the purpose, audience, and audience needs for a web authoring project.
- 1.2** Summarize how designers make decisions about the type of content to include in a web authoring project, including considerations such as copyright, project fit, permissions, and licensing.
- 1.3** Demonstrate knowledge of project management tasks and responsibilities.
- 1.4** Communicate with others (such as peers and clients) about design plans.

#### **DOMAIN 5.0**

#### **TESTING, PUBLISHING, AND MAINTAINING WEB CONTENT USING ADOBE DREAMWEAVER CC**

- 5.1** Conduct technical and usability tests.
- 5.2** Manage assets, links, and files for a site.

## CHAPTER 9

# Working with Outsiders

As a designer, you need to work with others. Being a designer and being an artist are two different careers, but most people who do creative work for a living need to do both. If you love to shoot old, crumbly walls and rusty farmhouse doors, you will find that you simply can't pay the bills with your "rusty hinge" photo collection. So a lot of photographers shoot weddings to pay the bills. It may not be their favorite type of photography, but it pays well and makes people happy.

Aside from the money, you will probably learn to develop an immense passion for creating designs for others. The secret to this part of the job is twofold: really listen and really *care*.

 **Video 9.1**  
*Introduction to Project Planning*

## Who You're Talking FOR and Who You're Talking TO

The first step in designing for a particular project is to understand the **client's needs**. This is critical because, among other things, the client pays the bills. Most of all, the client is hiring you to speak for them. For a designer, that's a weighty responsibility. You are being trusted to communicate for an entire company or cause. Therefore, you first and foremost need to address that client's needs and goals for the project. This will be the guiding principle when answering design questions. You must constantly remember your goals and focus narrowly on them to streamline your workflow and minimize distractions.

- ★ ACA Objective 1.1
- ★ ACA Objective 1.3

Let's look at a few example scenarios:

- Rockin' Zombies is a metal-opera-jazz-bluegrass fusion band that wants to promote an upcoming free concert at the local farmer's market.
- Hoi Polloi Church needs a brochure for its project to solicit volunteers and donations to help orphaned children recover from Hurricane Sally.
- OfficeHome Custodial Services wants a flier to share with businesses about its upscale, environmentally friendly office cleaning services.
- Zak's Bulldozer wants to promote its tree-removal service for residential homes, which uses low-impact tools.
- The city's health department wants a campaign to promote healthy eating and active lifestyle changes and to inform and warn people about common bad habits.
- Pop-Lite is a new, collapsible photo light, and the inventors need a logo and image for their Kickstarter campaign.

Each project has different goals, right? Some want to give away something. Some want to make money. Some want to solicit help from others. It's important to help the client pin down their project goals—and that can be difficult to do.

★ ACA Objective 1.1

▶ Video 9.2  
Discovering Client  
Goals

## Single Voice, Single Message

Here's a brainteaser. You have 20 close friends, and you can no longer understand what they say. Why? They are close enough to talk to one another. They are all speaking your language. They are all speaking loudly enough for you to hear clearly. They have no health or physical impairments. So what's happening?

They're all talking at once!

If a design says too much, it says nothing. It becomes “noisy” and makes it difficult for the viewer to focus on the main idea. This reality is similar to the design concept of *focal point*. Design projects have a kind of focal point too. It's important to clearly define and pin down the most important goals of a project. Sometimes, clients are trying to clearly define their company purpose, vision, or dreams for their organization. Hearing the overall goals and dreams for the business are *very* helpful in the design process, and necessary so that you understand your client. But to get a project done efficiently—and create a project that communicates well—you must work with the client to establish and narrow the goals for *this particular project* (Figure 9.1).



**Figure 9.1** Be sure you have a clear and concise understanding of the client's goals for the project.

The short version of a campaign's goals is often called the “elevator pitch” because it summarizes the project in the time that it would take for an elevator ride. It's communicating your purpose in a short, simple sentence. I normally push a client to shoot for seven words or fewer. The goal is to clearly define the goals for this particular design project.

Here are some elevator pitches related to the previous scenarios:

- Come to our free concert.
- Help child victims of a disaster.
- Subscribe to our news service.
- Remove trees without damaging your yard.
- Get healthy and avoid hidden dangers.
- Buy fun and functional little photo lights.

Admittedly, these pitches are not elegant or enticing. There's no “pop” to the message. But they *are* the very core of what you're trying to communicate. It is the reason your client is paying you to design the art. You'll need more detail than this to produce the *best* design, but focusing on this core goal can help you rein in the insidious forces of project creep. But first, let this sink in:

**The client's goals are the #1 priority.**

If the goal is unclear, the finished product will be unclear. Figure out the goal and you can always come back to it as a “home base” when the project starts to grow or lose its focus. Sometimes the goal isn't obvious or turns out to be different than it first appeared. But it's always critical, and one of your first jobs is to help the client focus on the primary goal of the project. Nonetheless, at the end of the day, you

work for the client, so the client calls the shots, has the final say, and makes the decisions—even if you disagree.

Now let's talk about the second most important person on the project—the one who doesn't really exist. I'm talking about the ideal audience.

★ ACA Objective 1.1

## Ideal Ivan and Danielle Demographic

In real life, you shouldn't judge people, make assumptions, and lump them together. But we do. And as designers, we must. This is called developing a “demographic” for the project and “identifying the target audience.” It's a critical step in helping your clients bridge the gap to the audience they want to reach. You do this by identifying the common characteristics of that audience and creating an image in everyone's minds of the typical customer: Ideal Ivan and Danielle Demographic. Some clients will say “everybody needs my product,” but those clients will still need to focus on a specific target demographic for *this project*. As the old saying goes, “Aim at nothing, and you'll hit it every time.” Those are wise words, especially when identifying a target audience.

▶ **Video 9.3**  
Finding the  
Target Audience

Identifying a target demographic (**Figure 9.2**) for your project is a critical step, second only to defining the client goals. And generally, it's also a part of those client goals. For example, when you want to create a new fishing pole, you can easily picture your target audience: people who fish. So you're probably not going to use the same graphics, words, images, and *feel* as you would to reach a punk rock audience. And at the same time, expectant mothers probably wouldn't be drawn in by images that would reach your fish or punk target.

Creating a demographic helps you focus on *who* should get your message. It's understanding the goals of your listener/viewer as well as the speaker, your client. You need to make sure you share information in a way that will connect to or resonate with that audience. And if you understand what your audience needs and feels, you can show how what you're offering meets those needs.

The easiest way to do this is to create imaginary “perfect fits” for your client's project. Here are some points to consider:

- **Income:** Determine whether you want to focus on quality, exclusivity, or price.
- **Education:** Establish the vocabulary and complexity of the design.
- **Age:** Dictate the general appeal, attitude, and vocabulary.

- **Hobbies:** Help choose images, attitudes, and terms and phrases unique to their interests.
- **Concerns, cares, and passions:** Identify core beliefs, trigger points, and so on.



**Figure 9.2** Once you have identified your target audience, you'll know how to speak their language.

It's pretty easy to see how different audiences will need different images. You don't want images of extreme sports in an ad aimed at expectant mothers. You wouldn't use a crowded nightclub image in a design for a camping and canoe outfitter company. It's a pretty obvious concept, but inexperienced designers sometimes design to please themselves, and that isn't always what pleases the target audience.

What makes your audience unique? Who has the problems that this product solves? Have those pictures in your mind. Work these ideas over with your client and help them envision their typical customer. Then look for images that will appeal to that ideal customer, this project's target demographic.

Think of yourself as a matchmaker. You're trying to introduce your client to the perfect customer or consumer. Speak in the language that the ideal clients would want to hear, and use images that will bring their lifestyle and outlook together with your client.

## The Golden Rule for Client Projects

▶ **Video 9.4** *The Golden Rule for Client Work*

Design is about helping others with their vision. It's about communicating a message. So when it comes down to it, use the business version of the golden rule: He who has the gold makes the rules.

Ultimately, you work for your clients. Help them see what you regard as the most effective way to reach the audience and identify the right questions to ask, but don't fight with them. Your clients have insight or perspective about their target audience that you are unaware of. When you disagree with clients about a design decision, help them realize their vision for their project. If you don't like it, you don't have to put the final piece in your portfolio, but you'll still get to put their check in your bank. If it comes down to what the client wants versus what you think the identified audience will respond to, do what the client asks. It's the client's project, audience, and money (**Figure 9.3**).

**Figure 9.3** He who has the gold makes the rules. Help your client achieve their goal, even if you don't completely agree with it.



This rule has one exception that you need to follow at all times. When your client asks you to skirt copyright law, you are still responsible to respect the law and your fellow designers. Often your client is just confused, and you can help them understand that you can't copy other designs or use copyrighted materials without authorization. Along those lines, let's take a moment to talk about copyright.



## Copyright Happens

### ▶ Video 9.5 About Copyright

The first thing to know is that copyright just happens. If something can be copied, then it's copyrighted. You needn't fill out any special forms, report to a government office, or do anything extra to put it in place. The law is written so that copyright happens as soon as something original and creative is recorded in a "fixed form." This means that as soon as you write something down, sketch it out, code a chunk of elegant programming, or click the shutter on your camera, whatever you just recorded is copyrighted. The only reason you might take any additional steps is to establish verifiable proof of *when* a creation was copyrighted, because the person who can prove that he or she recorded it first owns the copyright.

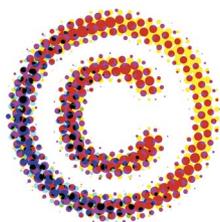
Imagine that you're in a restaurant talking with a friend. During this conversation you make up a song on the spot. A famous singer in the booth behind yours hears you and writes it down. He claims the copyright to those lyrics and makes a million bucks with your spontaneous song, and there's little you can do about it. However, if you recorded it on your phone when you sang it, you recorded it in fixed form *first*. So, *you* own the copyright to the lyrics, and the artist now owes *you* a truck full of money. Kind of weird, right?

Why does the law have this quirky little rule? Because the courts have to decide who actually owns copyright when its ownership is contested. And courts rely on tangible proof. So the law makes it simple by stating that he or she who first records something in fixed form wins. That way, if you have no proof that you were first, you're not going to try to sue someone. But even if you were first, if you can't prove it you're out of luck.

So why are there copyright notices on music, DVDs, and one of the first pages of this book? If we don't need it, why do we display it?

Simply put, it reminds people who owns the copyrighted material. If the date and copyright symbol were not displayed on this book, people might think they could legally make photocopies of it for their friends. Most people assume that if no copyright notice appears, then no copyright exists. (They're totally wrong.) The presence of a visible copyright statement discourages this conclusion and behavior.

So it certainly won't cause any harm to scratch a copyright symbol (©) on your art when you're finished with it (**Figure 9.5**), but it's more to remind *the public* than to protect *you*. You're already protected by law. Adding the copyright symbol to your work is like putting one of those security system signs in your front yard. It won't stop a really determined thief, but it can deter less-committed offenders. If you think it destroys the look of your front yard, you don't need it for protection.



**Figure 9.5** You don't need this mark or any sort of label, but it does show others that your work is copyrighted.

## Placing a Copyright Notice in Digital Content

The beauty of digital files is that they have the ability to contain hidden information that never compromises the enjoyment of the document itself. As a result, you can add copyright information to digital content without having a visually distracting copyright notice on the artwork. You do this by adding information known as **metadata** to your digital files.

Metadata is information that doesn't show up on the document itself, but is hidden inside the file. This is a perfect way to store copyright information, contact information, and so on. On some digital cameras, metadata can even record the lens information, the location via GPS, whether a flash was attached and fired, the camera settings, and more. In digital files, metadata can share the computer on which it was created, the time, and even the name of the creator. Be sure to use metadata when you're sending your work out over the web, and always check files that your clients give you to make sure that you're not violating another designer's means of making money when you're trying to make some yourself.

In web pages, metadata is easily stored in the `<head>` of the document using the `<meta>` tag, accompanied by named attributes that list the original author of the work and provide copyright information.

```
<META NAME="author" CONTENT="The name of the author and  
→ contact info or a link to their website.">  
<META NAME="copyright" CONTENT="Copyright © 2015 by Me">
```

"But I'm not trying to make any money with their art, so it's okay, right?" Well, that's a tough question with a few interesting rules attached.

## Playing Fair with Copyrighted Material

Can you use copyrighted material when you're learning to use Dreamweaver? Can you make a funny image using a movie poster by replacing the faces of the actors with the faces of your friends? What about using cool images from your favorite videogame that you include on a web page you're making in a web design class?

All these uses are completely legit. The people who came up with the copyright laws were very careful to make sure that the laws don't limit—but rather promote—creativity. They did this with a set of ideas known as **fair use**.

Fair use policy is a set of rules that make sure copyright protection doesn't come at the cost of creativity and freedom. Copyright can't be used to limit someone's

 **Video 9.6** *Digital Tools for Tracking Copyright*

### NOTE

*Metadata is how good technology teachers catch cheaters.*

 **Video 9.7** *Fair Use and Copyright*

personal growth or learning, freedom of speech, or artistic expression and creative exploration. Those ideas are more important than copyright, so copyright doesn't apply when it gets in the way of those higher ideals. You are free to use copyrighted materials in the pursuit of those higher goals. Some people (mistakenly) believe that fair use doesn't apply to copyrighted materials; but in fact, it applies *only* to copyrighted materials. Here is a list of issues that a court would consider when making a decision about fair use:

- **Purpose: If you used the work to teach, learn, voice your own opinion, inspire yourself to create a new piece of art, or report news, you're probably safe.** Protected reasons include educational purposes (teaching, research), freedom of speech (commentary or criticism), news reporting, and transformative work (parody or using the work in a new way). It isn't considered fair use if you're making money from the use, using it just for entertainment, or trying to pass it off as your own work.
- **Nature: If the nature of the original work was factual, published information, or the work was critical to favoring education, you're probably safe.** Was the content already published, fact based, and important for society to know? Then you're pretty safe to use the work. But if it was unpublished, creative (such as art, music, film, novels), and fictional, you're probably *not* cleared to use it.
- **Amount: If you use a small amount of a copyrighted work, it's more likely that your use of the work was fair use.** If you used only a small quantity—not the main idea or focus of the work, or just enough to teach or learn from—you're probably safe. If you used a large portion of the work or basically ripped off the central idea or the "heart of the work," it isn't fair use. For example, if you see a web design composition that includes a nice roll-over button effect that you want to emulate by writing your own code that achieves the same result, you're on solid ground. Taking the entire composition and simply changing a few names and labels and publishing it as your own work would be a violation of copyright.
- **Effect: If nobody was harmed because of the action you took, then your action was probably fair use.** If you used a legitimate copy of the original work, it didn't affect the sale of another copy, and you had no other way to get a copy, you're in pretty good shape. But if your copy made it less likely that someone would buy the original or you made a large number of copies, you're probably hurting the original creator, and that's not fair use.

As mentioned, copyright law really tries to address a simple question: How can we promote more freedom and creativity in the world? This is the question that copyright laws seek to answer. Fair use makes sure that beginning artists can experiment using anything they want. Just be sure not to share anything that might be another artist's copyrighted work.

But as a beginning designer, how can you get quality assets to use in real-world projects? Happily, you have access to more free resources than ever before via the Internet and free stock photo sites. Let's look at some.

## Uncopyrighting

You have a couple of ways in which to undo copyright. One is voluntary. An author can choose to release the copyright to their material. Believe it or not, this can be more difficult to do than you'd expect. Copyright law really tries to protect creators, and it can be difficult to *not* be protected by copyright law.

The second way is to let the copyright expire. Copyrights normally expire from 50 to 100 years after the death of the original author, but exceptions to this rule and extensions can be requested. It's beyond the scope of this book to discuss copyright at length, but it is important to realize that some materials have expired copyrights. When copyright is expired or released, the work is said to be in the **public domain**. This means that copyright no longer applies to the content, and you can use the material without worrying about infringement.

## Licensing

**Licensing** is another way that you can legally use copyrighted material. For designers and artists, licensing is fairly common because it allows you to use copyrighted material, for a certain time and in a certain way, by paying a fee established by the copyright holder according to the use of the material.

**Stock photos** are popular items licensed by Photoshop designers, and you can find them from many sources at varying prices. Stock photos are images for which the author retains copyright, but you can purchase a license to use these images in your designs. For almost everyone, this is a much less expensive solution than hiring a photographer to go to a location and shoot, process, and sell you the rights to an image.

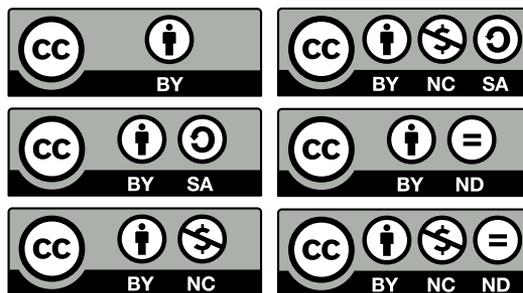
 **Video 9.8** Licensing  
Strict and Free

## CREATIVE COMMONS

In the last decade or so, a lot of exploration has been done in finding creative ways to license creative works. **Creative Commons** licensing is built upon copyright law, but it offers ways that artists can release their works for limited use and still choose the way the works are used and shared.

Creative Commons licenses include many different combinations of the following attributes and use many different symbols to identify the type of license (Figure 9.6), so you'll need to do some research when using Creative Commons–licensed materials and when releasing assets with Creative Commons licensing.

- **Public Domain** licenses, or “CC0” licenses, allow artists to release their works to the public domain. It’s a bit difficult to give your materials away to the public domain, but CC0 is generally recognized as a way to do so and is respected in most parts of the world.
- **Attribution** (marked as “BY”) requires that you credit the original author when using their work. You can do whatever you want with the work as long as you give that credit.
- **ShareAlike** (marked as “SA”) allows you to republish a creative work as your own creation, but you must provide the same license setting as the original author. This allows others to continue to share the work.
- **NoDerivs** (marked as “ND”) requires that you not change the material when you incorporate it into your own work. You can use NoDerivs material freely, but you must pass it along without change.
- **NonCommercial** (marked as “NC”) means that people can use your work in their own creative works as long as they don’t charge for it. You’re getting it free, so if you want to use it, then you have to be generous and also give away your work for free.



**Figure 9.6** Creative Commons licenses allow for a variety of easy-to-understand licensing options.

Creative Commons licenses are widely accepted and used, and you can find a ton of amazing resources that use this licensing. If you have any questions about Creative Commons licenses, you can find out everything, from a general overview to very detailed legal descriptions, at [creativecommons.org](http://creativecommons.org).

## PEOPLE PERMISSIONS

If you use pictures of people in a web design project, you'll need a special kind of license known as a **model release**. This type of release is required when a person's face is identifiable in the photo and the image will be used to promote something—whether it's a product or idea. Any work you do for a client is by definition a commercial use and would require a model release for every identifiable face.

## Think Like a Boss

Some might say that the only thing you need to know to succeed in life is how to solve a problem (**Figure 9.7**). That's not how to solve a *specific* problem—that would be only an exercise in memory. A well-trained monkey can mimic a person's actions and get a similar result, but he can't think with the depth of a human. My personal preference for solving problems is to work to understand things deeply and explore all the nuances of a potential solution. But for others, this process is a little outside their comfort zone and they find it faster and easier to copy someone else's solution.



**Figure 9.7** Make sure you stop to plan before working on your idea—it saves more time than it takes.

In the web design world, copying another solution might mean following tutorials or using designs that have been posted on sites such as CodePen (*codepen.io*), where designers display and freely share their work.

But what do you do when you're the leader? What happens when you need to do something new and fresh for your job? What if the boss doesn't know how to do it and that's why they hired you? How do you respond when the client's main request is "Do something that hasn't been done before."

That's where the problem-solving process comes in—the only skill you really need to be successful. If you can do that, you can figure out anything.

★ ACA Objective 1.3

▶ **Video 9.9** Project Management Intro

## Project Management

Project management is really just the problem-solving process in action—geared toward supervising resources, people, and team-based projects. The DNA of project management is problem solving and organizing the process so that you tackle the right issues at the right time using the right tools. Project-management systems take on a million forms (as do problem-solving systems), but if you truly grasp the ideas behind the labels, you can translate them into whatever management strategy your client, team, or boss is using.

The problem-solving process is creative in itself. A good solution to a problem can be artistic in its elegance and efficient grace. A problem solved well can be *beautiful*. This is the most important section of this whole book, because if you can grasp problem solving, you can learn whatever you need to learn now and in the future. Here is where we shove you out of the nest.

The following process will help when you need to solve a technical problem on hardware or equipment, handle a design problem within Dreamweaver, or figure out how to get your hand out of that jar you got it stuck in. It will help you translate a tutorial written for Dreamweaver CC into your current version (things change fast these days). And it boils down to three simple steps: **learn, think, and do.**

## Learn

▶ **Video 9.10** Project Management—Understand the Problem

The first step of a problem-solving process is to learn. It includes two important steps: learning what the problem is, and learning how others have solved similar problems (research and investigate). It seems simple, but the process can be confusing. Most projects with major problems get stuck at this initial step because they didn't learn well—or at all. Let's talk about them.

## UNDERSTAND THE PROBLEM

I talked about understanding the problem earlier in the chapter; the first step of every project is to understand the problem. For most design projects, you must figure out how to most effectively help your client share goals with the target audience. If the client is yourself, then it's about getting to the essence of what you want to communicate and to communicate it so that your target audience can act on it.

Mess it up here, and by definition you are not solving the problem. You haven't properly identified it, so how can you solve it? You might be solving *a* problem, but you're not solving *the* problem. Sometimes you can actually make the problem worse by implementing a plan that creates a new problem without solving the real one.

You can avoid trouble down the road by clearly understanding and defining the problem at the start. "I want to sell a million widgets" is not a problem you can solve; it's a desire the client has. So what's the problem you can help with? They haven't sold as many as they want to? That's not it either. How do you get to the bottom of the problem?

Start with good questions (**Figure 9.8**): Do people need a widget? If so, do people know that widgets are a thing? If so, do they choose a competing widget? If so, why do they choose this other widget? Why do you think they should use your widget instead? Who would be most likely to buy your widget? What is your budget for widget advertising? What do you want to spend on this particular project? What are your expectations?



**Figure 9.8** Nailing down the client's goals and preferences at the beginning of the project will help you develop an efficient plan.

Many clients become frustrated with this set of questions. They just want to see some action so that they can feel like they're doing *something*. But let me repeat that old saying: "Aim at nothing, and you'll hit it every time." This is when you sharpen your axe so you don't have to bang at the tree all day.

This part of the project can be fairly informal on smaller projects, but it can be huge on large projects. Here's a list of critical questions to answer:

- **Purpose:** Why are you doing this design project? What result would you consider a success?
- **Target audience:** Who needs this message or product? Describe your typical customer.
- **Limits:** What are the limits for the project? Budget and time are most necessary to nail down.
- **Preferences:** Aside from the results already discussed, are there any other results you'd like or expect from this project?
- **Resources:** Does the client already have the artwork, video, or other resources that need to be included? Are they in a format that's ready for the web?

These examples are intended to show how quickly you can determine a client's expectations. The answers to these questions define the size of the job and how you'll best be able to work with the client.

Sketches and written notes from this initial step will help. Gather as much information as you can to make the rest of the project go smoothly. The more you find out now, the less you'll have to redesign later because the client hates the color, the layout, or the general direction you took the project. Invest the time now, or pay it back with interest later. With a clear idea of what the problem is, you'll get the information you need to solve it in the next step.

## RESEARCH AND INVESTIGATE

After you understand exactly what your client is expecting, you can start doing the research to arrive at the answers you need. Let's take a quick look at that word: *re-search*. It literally means "search again." Lots of people fail to *re-search*; they just search. They look at the most obvious places and approaches, and if things don't immediately click, they settle for a poor but quick and easy solution.

Depending on the job, research can be a relatively quick process. Find out about the competitive products, learn about the problem you're trying to solve, and

understand the demographic you're going to target. Sometimes this can be done in your initial conversations with the client, but for larger projects you may need to spend more time and go as far as creating proofs of concept to help you define possible solutions. The main thing is to be sure to analyze the problem, and look for ways that others have solved similar problems. Again, the more research you do, the better information you'll have about the problem you're trying to solve, which will help you with the next step.

## Think

The next couple of steps represent the “thinking” phase. You can do this quickly using a pen and napkin, or you can do it in depth and generate tons of documentation along the way, particularly on large projects. But thinking is the part that most of us often mistake as the beginning. Remember that if the learning step isn't done well, your thinking step might be headed in the wrong direction.

As a designer, you'll often do much of your thinking on paper (**Figure 9.9**). Sketch it out first and have that reference to show the customer, and come back to it for clarity.

 **Video 9.11** Project Management—Think It Through



**Figure 9.9** Sketch first and establish clarity.

## BRAINSTORM

As with “re-search,” you need to really grasp the meaning of brainstorm. It’s a *brainstorm*. Not a brain *drizzle*. A full-on typhoon of ideas (**Figure 9.10**). It’s critical—*absolutely critical*—that you get out of the “analyze” mode of thinking and into your “creative” mode. They are two different mind-sets. Actually, they’re opposite mind-sets. If you start thinking critically instead of creatively, you’ve changed direction and you will lose ground on your brainstorming task. If you start moving in the critical direction, that’s the opposite of creative. Stop that! Don’t try to work hard on brainstorming. Work relaxed instead.

**Figure 9.10** Brainstorming is an inherently disorganized and messy process—clean up your ideas later.



But sometimes analysis happens. You start analyzing how to complete your ideas when you should be creating them. Here are some of the things that trigger the critical mode of thinking. Don’t do them when brainstorming.

- **The No-No’s:** Judging your ideas. Trying to finish an idea when you should still be brainstorming. Getting stuck on a particular idea. Planning out the project. Thinking about how much time you have. Thinking about the budget. Thinking about numbers. Grouping or sorting your ideas. Developing the idea that you think is best. Choosing a solution because the technology is cool.

Here's what you should be doing:

- **The Go-Go's:** Listen to music. Look at cool art online. Call a friend. Doodle something. Read a poem. Take a break. Go for a walk. Listen to some comedy radio. Write a haiku. Meditate for five minutes. Exercise. Sleep on it.

When you're in brainstorming mode, don't edit your ideas. Let them flow. If a crummy idea pops into your head, put it on paper. If you don't, it will keep popping up until it's been given a little respect. Give the weak ideas respect. They open doors for the great ones. Brainstorming is completely a matter of creating ideas.

## PICK AND PLAN

After brainstorming, you need to pick a solution that you generated in your brainstorming session and plan things out. You'll find that the plan you go with is rarely your first idea. Through the process of brainstorming, the idea will go through several **iterations**. A common mistake for beginners is to fall in love with an early idea—beware of this pitfall (**Figure 9.11**). Your best idea is lurking in the background of your mind, and you have to get rid of all the simple ideas that pop up first. For a small project or a one-person team, you might quickly hammer out a contract and get to work, but in larger projects, the planning needs to be detailed and focused.

 **Video 9.12** *Project Management—Get It in Writing*



**Figure 9.11** Resist the urge to fall in love with early ideas—the best ideas come near the end.

The larger the project, the more formal this process will be. Small projects with just one person working on them will need little planning for moving forward. But larger projects will need a project plan to set the project requirements for the team.

## SETTING PROJECT REQUIREMENTS

This is where the action happens. Look through the ideas you've generated, pick the one that seems best, and plan how to make it happen. This is where you determine exactly what needs to be done, establish some direction, and identify a clear target. Did you catch that last sentence? This planning stage (which most creative types naturally tend to resist, myself included) is where you clarify what needs to be done, establish your direction, and identify a clear target. We resist it because it seems to limit us. It ropes in our creative freedom and gives us a checklist—things that many creatives hate. These things are creative kryptonite—or at least we *think* they are. But let's really consider this for a moment.

If you don't perform this admittedly tedious step, what won't you have? You won't have a definition of what needs to be done, a direction to head in, or a target to hit. Everyone will be in the dark. Although this step doesn't seem creative in itself, creativity isn't the priority at this particular juncture. You're at a journey moment versus a destination moment. Creativity without limits is a journey, which is great for your own work but a disaster for a client-driven job. A client-driven project requires clearly defined goals—a destination. You need to arrive somewhere specific (Figure 9.12).

**Figure 9.12** A well-developed plan with sketches will ensure a smooth working relationship with the client.



Two critical points that need to be a part of every project plan are the project scope and the project deadline. Every contract needs to have these critical components defined to focus the project and make it clear to everyone involved the “what” and “when” of the project.

- **Project scope** is the amount of work to be done. On the designer’s side, this is the most important thing to establish. If the scope isn’t clear, you’re subject to the Achilles heel of design work: project creep. This is a pervasive problem in our industry (and I’ll address it later in detail), but simply writing down a defined scope can prevent the problem. Get in writing *exactly* what you need to do, and make sure specific numbers are attached.
- **Project deadlines** dictate when the work needs to be done. This is the client’s most important element. The deadline often affects the price. If the client needs ten JPEG images in six months, you can probably offer a discount. If they need art by tomorrow morning, then they’ll have to pay an additional “rush” fee. Deadlines on large projects also can be broken down into phases, each with its own fee. This division of tasks helps you pay the bills by generating cash flow during a large and lengthy project. It also limits the impact—for you—of payment delay.

These items, when shared and discussed with your client, will save time, money, and disagreements. These additional **deliverables** are the raw materials of project planning and help convey the exact target of the project. It’s my opinion, and that of experienced designers, that the following two deliverables are critical for every design project.

- **Sketches** are really helpful to show the client your design direction. Even better is when the client has an idea of what they’re looking for and can give you their own sketch. Does this limit your creative freedom? Yes! It also saves you a ton of time. The goal of a client job is to get a project done to their satisfaction. If they are very particular and know what they want, you’re not going to convince them otherwise. Sketches save time because they limit your direction to a direction that the client will accept, and they help you get to that acceptance faster. That means you finish and get paid sooner. The better the sketches are before you get into actual design, the fewer changes and redesigns you’ll need. You don’t need to be a master sketch artist. Just convey the idea. Sometimes sketches may just be **wireframes**, very rough representative sketches of how to lay out the document.

- **Specifications** are detailed, clear written goals and limits for a project. Many times, the specifications themselves will be referred to as the “project plan” and become part of the contract. This is admittedly the least creative and most boring document you will create, but it is also the most critical. All project plans should include two critical pieces of information: the scope of the project and the deadlines that need to be met. Be sure to *always* include *both* of these items in your project specifications.

## AVOIDING PROJECT CREEP

▶ **Video 9.13** Project Management—Avoiding Creep

No, the **project creep** is not the person on your project who stares at you from around the corner. That *is* creepy, but that’s not the meaning. Project creep is when the project starts to lose its focus and spin out of control, eating up more and more time and effort (**Figure 9.13**). It is really important to be aware of this phenomenon. It happens all the time. And the main culprit in every case is a poorly designed project plan that lacked clear specifications and deadlines.



**Figure 9.13** A project can appear small but have lots of issues under the surface. A solid contract will avoid this becoming your problem.

Here's how it happens: Joe Client creates a product and wants to sell it. He comes to you for marketing materials. You determine that he wants a logo, a flyer, and a three-page informational website. A price of \$2500 is settled on, and you've got a month to get it done. You go to work.

Then Joe realizes that he also wants some images for social media. Could you just make a few? He also realizes that he needs to put his new logo on new business cards. Could you just design a card with the logo on it? Oh, and he can't figure out how to get his product onto his favorite online marketplace. Could you just help him set that up? And he changed his launch date. He doesn't need it next month; he needs it next week because he just reserved a booth at a large convention. By the way, do you know anything about designing booths? Can you see how critical it is that you create a detailed project plan with task definitions and deadlines attached? Sometimes the client asks for something and it takes you 30 seconds. It's a good idea to always happily deliver on these little items. A favor is any job that takes you five minutes or less. After that, the favor turns into work. And your only defense is a contract defining a clearly stated scope.

Just make sure that the project's scope is clearly stated. If the contract says that you'll provide *any* images for the company's web presence, then you're in trouble. On the other hand, if it specifically says that you'll provide up to nine images for a three-page website, then you're in great shape. Taking the hour it requires to specify your project and its deadlines in detail will save you from many hours of work and contract revisions.

If the client approved a sketch with a basic layout and then asks for something different, you need to charge them for the change—if it's going to take more than 5 to 10 minutes. Sticking to this policy helps the client think about changes before sending them to you. If you fail to charge when addressing impromptu changes, the client has no reason to think about the requests in advance. Charging your customer for additional changes focuses them on what they really want.

Of course, if the client asks for something that actually makes the job easier and faster, then make the change and do it for free. The bottom line is this: Establish good will whenever it's good for both you and your client. But when an eleventh-hour alteration serves only one side of the relationship, the requesting side has to pay for the service. In the end, this arrangement ensures that everyone ends up winning and that the client regards you as the professional that you have become.

## Do

▶ **Video 9.14** Project Management—Make It So

The last phase of the project plan is to knock it out. This is the “two snaps and a twist” phase because it generally happens pretty quickly when you have a good plan—unless there’s a hitch. (I’ll get to that soon.) But at this point, on most design projects, you’re pretty much wrapping things up.

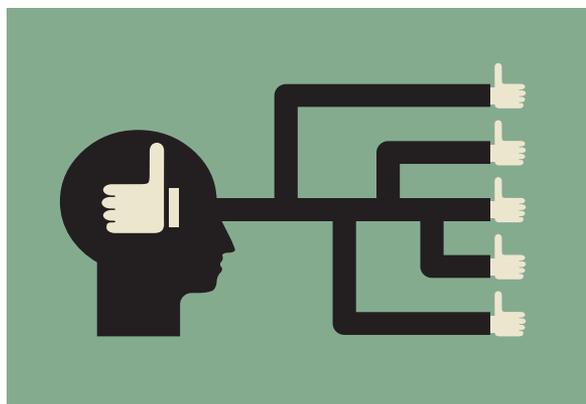
### BUILD IT

This step is pretty obvious: Make it happen. This phase is where most people think all the action is. But honestly, if you’ve done the prior steps well, this can be the fastest part of the process. You already know what to do; now just do it. The design decisions have been made and you get to work. Of course, when doing this step, it’s best to regularly refer to the specifications and keep the client informed. The best way to do so is to have a very intentional feedback loop in place.

### FEEDBACK LOOP

★ ACA Objective 1.4

A **feedback loop** is a system you set up to constantly encourage and require input and approvals on the project direction. Keeping your client informed is the best way to speed through the process (**Figure 9.14**). For a Dreamweaver project, **iterative work** establishes effective guideposts to send to the client for review and input. Iterative work is work you’re sharing as it’s done. Doing so performs a couple of critical functions. First, it lets the client see that work is being done and reassures them that the process has momentum. Second, it lets the client chime in on anything that they don’t like while it’s still easy to make a change.



**Figure 9.14** A good feedback loop helps ensure that what you are thinking is what the client will approve.

Establishing this open communication channel encourages and enforces a healthy exchange of opinions and can enable you to most efficiently adjust and fine-tune your project to suit your client.

## TEST AND EVALUATE

This very last step can also be fast if you've had a good feedback loop in place. For Dreamweaver projects, it's essentially checking the work against your project plan and making sure that you met all the specifications to satisfy you and your client (Figure 9.15). If not, you should essentially start the problem-solving process again to understand the current problem. Find out exactly what, according to the client, doesn't meet the requirements.



**Figure 9.15** If the project meets all the specifications in your project plan, you're done!

Assuming a good project plan with sketches and a good feedback loop, the test and evaluation phase should require just minor tweaks—no different than any other iterative work resolution. If you didn't have a good feedback loop and the first time the client sees your work is upon delivery, that client could become really unhappy and demand innumerable changes. Avoid this migraine headache with an effective and well-defined feedback loop as part of your well-developed plan. Those two tools are your weapon against project creep and unreasonable clients.

## WORKING FOR "THE MAN"

Many web design artists begin their careers working at larger firms, where they use Dreamweaver to maintain and add to existing websites. This can actually be a much easier way to get started than freelancing with your own business. If you're exclusively a designer, this type of job may require you to do the only tasks you're best

 **Video 9.15** *The Advantages of Working at a Firm*

at doing. In a large firm, someone else does the sales, manages client relationships and projects, does the backend programming for the website, and creates technical specifications. Often, you'll use the same print house or web templates, you'll know exactly what size images you need, and you may even have existing designs that you'll add to as new content is needed.

Working within an experienced company can also be an amazing education, as you can develop your strengths, learn about the industry, and slowly increase your involvement in the other aspects of this career beyond Dreamweaver proficiency.

- ★ ACA Objective 1.1
- ★ ACA Objective 1.2
- ★ ACA Objective 1.3
- ★ ACA Objective 1.4
- ★ ACA Objective 5.1
- ★ ACA Objective 5.2

## The Language of Web Project Management

The terms and processes that you've learned in this chapter will serve you well no matter what kind of design work you're doing. Whether you create logos using Photoshop, videos using Adobe Premiere Pro, or interactive games using Adobe Flash Pro or the new Adobe Animate product, the principles of the project management process remain constant. And you have made your client, Chris (**Figure 9.16**), very happy... and that is the ultimate goal, right?

**Chris Sez:**

*Hey, guys! Thanks for being such great pros when it came to putting my projects together!*

*I really appreciate all the care and time you took in making a plan, getting everything together, and hitting our objectives for price and delivery. I know there were times when I didn't understand the process and some of the terms you used were a bit confusing—like that wireframe thing—but in the end the wulvs.com website was on time and on budget and exactly what I wanted, so that's what really matters.*

*Chris*

Different industries develop their own languages and jargon, and you'll need to be able to speak the language when you're dealing with the natives of those worlds. In the web design world that Dreamweaver inhabits, there are certain phrases that you'll need to know. You've already encountered many of these in your work for Chris the Cartoonist, so let's review.



**Figure 9.16** Chris the Cartoonist

## Gathering Requirements

At the beginning of this chapter you learned that the client's goals are your number-one priority. When you are developing websites, the client's goals will have specific aspects that are unique to delivering content online. And think about the many ways that might be done! In the projects you completed for your fictional client Chris, you already created a traditional website, two mobile versions, and even a version that could be converted to a phone application. That's part of the excitement of learning web design! On any given day you might be working for a company that needs a simple online brochure while you also work on an e-commerce site for another.

Because website production has so many roads you might travel in meeting your client's requirements, it's important that you gather as much information as possible during an initial **client survey**.

You can review the list of questions that were posed in Project 3.1 as an example of the kinds of questions that you'll need to answer during the requirements phase for a brand new website. For an existing site, you'll also want to get the answer to these questions:

- Describe the experience a potential visitor to your site will have when they interact with your new site. What kinds of improvements and changes do you want to make from the existing site?
- Describe what's good and bad about the current site and why you want to make a change.
- Have you conducted **usability testing** of your site to find out how long visitors spend there and how they find information?
- Do you have **analytics** set up for your current site to measure how many visitors your site gets and information such as their location and what devices they are viewing the site on?

## Information Architecture

Recall from your early interactions with Chris that on his list of requirements were the pages needed to display information about his fictional company. During the information architecture phase, you considered how the pages of the site would be organized by developing a **site map**, or visual representation of how the documents

in the site would be placed into their respective folders. Information architecture is the process of translating these concepts from paper sketches, **storyboards**, or **wireframe** documents to the actual files and folders within your local computer and, ultimately, to a live website.

There are many ways of documenting these steps in the web design planning process, but they all depend on a few common elements, such as URLs for pages within the site, colors, fonts, and design elements required for the pages to be functional. This stage often requires a good deal of brainstorming and the use of proof-of-concept ideas that you try out on your own or with a team.

## Site Design

If you think back to Chapter 4, where images, colors, and other resources were provided by the client, it was at this stage that the site began to take on its look and feel. In a traditional web design process, you would now produce **page comps**, or compositions, that are intended to show the client a final prototype of both the design and functional aspects of the proposed site. These graphical representations of how the site will look are typically created in Photoshop to make it easier to respond to client changes and clarification of their design requirements.

Common tasks for a web design team at this stage follow many of those you used in the projects you completed in Chapters 5 and 6, including receiving text copy from the client, establishing the navigation to be used in the site, and creating a prototype for the client's approval.

## Site Construction

This is the production phase that takes place after the client has given their final approval to the prototype and supporting designs you've created. You took these steps during Chapter 6 when you moved the style sheet to an external file, added comments to your style sheet, and created Dreamweaver templates and library items to use as you produced the actual pages for the site.

It's also during this stage that testing moves from simply previewing your designs in a web browser to sharing information with others as the construction process continues. This will involve frequent contact with the client, but may also require that you bring in others to provide their feedback on the site and how interactions

take place and information is found. Typical questions asked during the testing phase include:

- How do the graphical design elements of balance, emphasis, color, tone, and the arrangement of information affect the design?
- Do the images used in the design add to the user experience?
- How quickly do pages load?
- Do all the pages appear the same on different computers and devices?
- Are pages consistent when you navigate within the site? Are common elements in the same location?
- How consistent is the use of color, fonts, and other styling elements within the site?
- Does the design feature good visual contrast, alternative text for images, and other features that ensure **accessibility**?
- Are the page titles, page descriptions, and other meta content in place that will allow search engines to discover the site?
- Does the text in the visible area of the page feature strong keyword correlations that match the goals of the site?

## Publish and Maintain

With all the individual web pages complete and final technical and usability testing behind you, the final step to take is to publish the site to a live web server, as you simulated in Chapter 6. It was at this point that you saw the way that Dreamweaver allows you to get files from your computer and put them to a remote location, as well as how you can use features such as Check-In/Check-Out and Subversion services to collaborate on a project with others.

Depending on the role that you played with the development of the site, you may have an ongoing relationship with the client in which you update and maintain changes as they come along, analyze the traffic to the site, and suggest changes to make it more effective. Or you might simply be handing things off to others to complete while you do more specialized tasks, such as creating advertising banners, design comps, or photo processing or graphic design support.

No matter the kind of work you're doing, what you now know about programming for the web and using Dreamweaver will be invaluable as you find the job niche that's right for you.

## WHAT IS SUBVERSION?

Subversion is a version control system—sort of a time machine for data—that allows files to be copied, moved, renamed, and deleted, all while maintaining backups through the entire process. Subversion is a very common collaborative system that allows multiple people to share the work of editing and publishing website files.

Dreamweaver provides support for Subversion (SVN) systems to function, but you must have access to an SVN server and an SVN repository—a network location where you and other users can get and commit files. The typical Dreamweaver workflow is to get and commit files back and forth between the SVN server and put them to the live server when ready. The remote server setup is completely separate from the SVN setup.

★ ACA Objective 1.1

★ ACA Objective 1.3

▶ **Video 9.16**  
Wrapping Up  
Project Planning

## Conclusion

Much of this chapter digressed from the hands-on Dreamweaver stuff that most books cover. But the lack of this information can present a problem for many beginning artists. You need to master a lot of industry information, creative knowledge, and business skills to be successful. Many creative people have a hard time with the business side of the career, but it's best to understand these ideas and concepts now before a lack of understanding becomes a problem. The tips and techniques that you've read in this chapter will truly help eliminate a lot of the inherent frustration in the complexity of working with and for other people. Dreamweaver is a creative application and tends to attract creative types. The qualities that make us great at thinking outside the box and designing new and beautiful images are the same qualities that may make us less skilled at the organized detail work of business and client management.

## Onward and Upward!

I hope you've enjoyed this book, come to appreciate just how fascinating the world of web design is, and learned the amazing opportunities that are available to you if you pursue a career in this field. Web design is without a doubt the fastest-moving digital design career available today, and if you love the challenge of learning new things and the simple joy of creating work that has value, you can't go wrong in thinking about the many, *many* ways you can get into the web design field.

The work that is needed in the web design field is as varied as the people who work to make it happen. From the hardcore programming whiz who loves building databases, to the interactive user experience geek who likes making animations and coding for visual effects, to the seriously artistic designer who can make a page comp sing in Photoshop—all of those kinds of careers are available.

But let's be honest here. Dreamweaver is far from the only way that web content can be published online. The idea that we could all be carrying around our own little web publishing devices in our pockets hadn't even been thoroughly considered when Dreamweaver came into being in the late 1990s, but these days we can all push videos, messages, likes, and comments instantly and from anywhere. You no longer need to be a web “master” to publish online.

What you do need to master is a thorough understanding of the interplay that takes place between HTML content, the styling rules of CSS, and technologies like JavaScript and HTML5 animations. To develop an understanding of the principles of web programming, you can't go wrong by continuing to work with Dreamweaver and honing your coding and design skills.

It's my hope that you're now ready to take the next step to a great career as a designer, programmer, game designer, or animator (or some job title that none of us have even thought of yet) and that this book has helped prepare you to successfully complete the Adobe Certified Associate exam for Dreamweaver.



JOIN THE  
**PEACHPIT**  
AFFILIATE TEAM!

You love our books and you love to share them with your colleagues and friends...why not earn some \$\$ doing it!

If you have a website, blog or even a Facebook page, you can start earning money by putting a Peachpit link on your page.

If a visitor clicks on that link and purchases something on peachpit.com, you earn commissions\* on all sales!

Every sale you bring to our site will earn you a commission. All you have to do is post an ad and we'll take care of the rest.

## APPLY AND GET STARTED!

It's quick and easy to apply.

To learn more go to:

**<http://www.peachpit.com/affiliates/>**

\*Valid for all books, eBooks and video sales at [www.Peachpit.com](http://www.Peachpit.com)



**Peachpit**