

ADOBE® DREAMWEAVER® CS4

CLASSROOM IN A BOOK®

Instructor Notes



Adobe® Dreamweaver® CS4 Classroom in a Book

© 2009 Adobe Systems Incorporated and its licensors. All rights reserved.

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

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

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

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

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

Apple, Mac, and Macintosh are trademarks of Apple, registered in the U.S. and other countries. Microsoft, Windows, and Windows NT are trademarks of Microsoft Corporation registered in the U.S. and/or other countries. All other trademarks are the property of their respective owners.

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

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

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

Writer: Virginia DeBolt  
Project Editor: Nancy Peterson  
Production Editor: Cory Borman  
Development Editor: Bob Lindstrom  
Technical Editors: Jim Maivald, Judith Stern  
Copyeditors: Elissa Rabellino, Dan Foster  
Compositor: Owen Wolfson  
Indexer: Karin Arrigoni  
Media Producer: Eric Geoffroy  
Cover Design: Eddie Yuen  
Interior Design: Mimi Heft

Printed in the USA

Book:  
ISBN-13: 978-0-321-57381-0  
ISBN-10: 0-321-57381-1

Instructor notes:  
ISBN-13: 978-0-321-61928-0  
ISBN-10: 0-321-61928-5

# INSTRUCTOR NOTES

## Introduction

The Adobe® Dreamweaver® CS4 Classroom in a Book™ course provides students with techniques, tips, and solutions for using the Adobe Dreamweaver CS4 software. These instructor notes are intended to complement the information in the Adobe Dreamweaver CS4 Classroom in a Book. The information in these notes is organized to follow the sequence of instruction in each lesson.

## About the workbooks

It is recommended that each student in the class have an individual copy of the Adobe Dreamweaver CS4 Classroom in a Book. Students will use this book as you guide them through lessons. You can buy more copies of the Adobe Dreamweaver CS4 Classroom in a Book for your students, or you can refer them to a local bookseller to purchase the book.

For additional information, call Pearson Education at 800-922-0579.

## Course strategy

This book consists of 11 lessons. If you cover one lesson per week, students will have time at the end of the semester to work on their own, constructing a new site using the Dreamweaver tools.

The lesson exercises are designed so that students are introduced to the important Dreamweaver tools and functions. All the new or enhanced features added in Dreamweaver CS4 are included in the exercises.

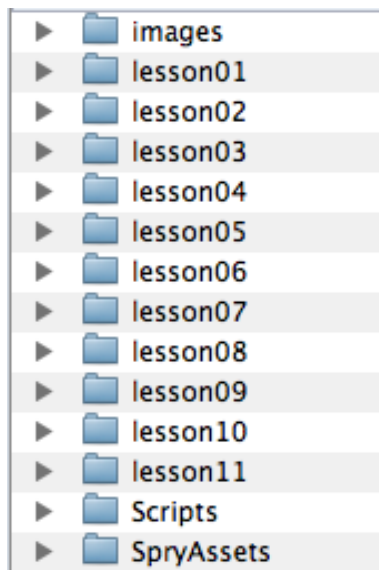
While completing the exercises, students create a complete website for a fictitious travel company. They also create a few pages unrelated to the travel site simply as practice exercises. Students primarily use the built-in CSS layouts; however, they also build their own CSS layout in Lesson 7.

You can evaluate each student based on his weekly completion of the lesson exercises. Requiring a student to design and build her own original website is a good culmination activity and could serve as a final exam.

The following lessons discuss enhanced and new features in Dreamweaver CS4:

- Lesson 1 introduces Live view. Live view is a full browser rendering inside Dreamweaver. Scripts run, links work, and movies play. Students can toggle between Design, Code, and Live views with the click of a mouse button.

- Lesson 1 familiarizes the student with the vertical Code and Design view and Split Code view.
- Lesson 1 introduces the HTML and CSS buttons on the Properties panel, which allow the student to toggle between HTML and CSS properties for any element on a page.
- Lesson 1 introduces changes in the way Dreamweaver identifies and creates CSS selectors in the New CSS Rule dialog box.
- Lesson 3 teaches students how to use the Spry Tooltip widget, which creates a styled tooltip controlled by JavaScript.
- Lesson 4 examines the workflow relationship between Adobe Photoshop and Dreamweaver, and teaches students about the Photoshop Smart Object.
- Lessons 5 and 6 demonstrate the use of several new or enhanced Spry widgets, including menu bars, accordion panels, and tabbed panels.
- Lesson 8 provides instruction in new Spry form validation and check box widgets.
- Lesson 9 examines the new way that Dreamweaver uses the <object> element to add Flash video and Flash movies to a web page.
- Lesson 10 introduces Library items and server-side includes.
- Lesson 11 describes how to cloak individual files.



## Managing student projects

One way to simplify file storage and retrieval in the classroom is to instruct students to create a main folder on their hard disks and name it [Student's Name] Lessons or [Student's Name] DW CIB, and then copy each project folder into the main folder. Having students save all of their working files to their own main folder makes it easy for you to clean up files when a course is completed.

The Lesson folder structure for each student, when complete, will look like this:

To save hard drive space, the individual lesson folders can be copied and deleted as necessary; but the images, SpryAssets, and Scripts folder should remain in the student's Lessons or DW CIB folder at all times during the course.

## Additional resources

Instructors and users of Dreamweaver rely on a variety of resources to supplement their knowledge of the program. You may find the following resources useful to explore.

### Dreamweaver Help

The Help contents installed with Dreamweaver include complete descriptions of all of its features, including topics not covered in the Dreamweaver User Guide (see below). They also contain links to movies that showcase new CS4 features and techniques.

### Adobe Dreamweaver CS4 User Guide

This guide, a subset of the Help contents, includes feature descriptions. It's useful for learning Dreamweaver features that are outside the scope of this book. Printed copies of the Adobe Dreamweaver CS4 documentation are available for purchase by visiting the "Official Product Documentation from Adobe" page ([www.adobe.com/go/buy\\_books](http://www.adobe.com/go/buy_books)).

### Application DVD

The Adobe Dreamweaver CS4 application DVD contains the Adobe Technical Info folder, PDF files of the User Guide and Quick Reference Card, demo versions of products by Adobe and other manufacturers, and the Adobe Dreamweaver CS4 Software Development Kit.

### Adobe CS4 Video Workshop DVD

Included with Dreamweaver, this DVD provides 250 instructional movies on Dreamweaver CS4 and other products in the Adobe Creative Suite 4.

## Adobe website

The Adobe website ([www.adobe.com](http://www.adobe.com)) contains a variety of Dreamweaver training and support resources, including the Adobe Studio Expert Center for Dreamweaver. Many step-by-step tutorials, galleries, and other resources are available here, and the user-to-user forums are great places for students to learn from more experienced users. The Adobe Design Center ([www.adobe.com/designcenter](http://www.adobe.com/designcenter)) provides hundreds of tutorials from experts and authors in the Adobe community, as well as thoughtful articles about design and technology.

## Adobe Certified Expert (ACE) program

This program is designed to validate an expert skill level of Adobe Dreamweaver CS4. Testing procedures ensure that each ACE-certified student demonstrates expert product knowledge of the current release of Adobe Dreamweaver, resulting in enhanced marketability and credentials for ACE graduates. ACE program training is available through Adobe Authorized Learning Providers (AALP) and self-study. For more information about this program, send an e-mail to [certification@adobe.com](mailto:certification@adobe.com) or visit the Adobe website ([www.adobe.com](http://www.adobe.com)).

# Lesson 1: Getting a Quick Start

Lesson 1 gets students off to a fast start by using one of Dreamweaver's ready-to-go CSS (Cascading Style Sheet) page layouts. Students modify a CSS layout and add content to create a page for a travel website. As students work through the exercises and complete that first page, they also become familiar with the Dreamweaver workspace.

## Defining a Dreamweaver site

Setting up the student's folder on the hard drive and defining it as the site root is the first exercise. This is an important step to perform for every site that the student will make now and in the future, so it should be emphasized here. All the files for the travel site and other exercises in the book need to be in this folder. Dreamweaver maintains links and relationships between all the files in the site by identifying the site root folder. Although the students won't set up the connection to a remote site in Lesson 1, make sure they realize that the site definition dialog box is where this task is done.

## Selecting a CSS layout

Students create their first new page using the Welcome screen. When the New Document dialog box appears, spend time exploring all the options for creating new documents before settling on the two-column, fixed-width layout that will be used for the new page. You might demonstrate some of the other layout options to

make sure that students understand the difference between fixed pixel width and stretchy layouts. You might also demonstrate the difference between the layouts by opening up an example of each and enlarging the text and expanding the window in a browser to show students how a layout changes when based on percent or ems compared with pixels. All the exercises in the book involve fixed-width designs, so students will benefit from time spent explaining the features of other Dreamweaver CSS layouts.

## **Saving a page**

When saving the page, make sure to explain the process of locating the student's main folder and saving the file in its lesson01 subfolder. The CSS file should also be saved in that lesson01 folder.

## **Explore the Dreamweaver views**

When the new page is saved, spend some time exploring Dreamweaver's views. In addition to the familiar Code, Design, and Code and Design (Split) views, Dreamweaver now has Live view, which allows you to toggle between a Design view and a live browser, all within Dreamweaver. Dreamweaver also has a new Live Code view (which you can explain but will not use in Lesson 1). In Live Code view, students can open up scripts and linked code. As changes are made in Live Code view, they are instantly reflected in the other Dreamweaver views.

## **Adding titles, headings, text, images and modifying the CSS**

Students are ready to change the page by adding a title, new headings and text, and images. Content for these changes is provided in an .rtf document, which students copy and paste. With real content added to the page, the students modify the page appearance: changing colors, fonts, and background color, and adding a graphic background for the header. They have their first experiences with the CSS panel, rules dialog box, and other Dreamweaver CSS features that they will use throughout the book. Their work with the CSS Rules Definition dialog box is limited in Lesson 1, but you might want to demonstrate it in more depth and explain its various categories.

## **Working with code and Code and Design Views**

When the page is almost complete, a few small changes are made to give the students practice with changing elements in the code. Students also work with a new Dreamweaver feature that splits code vertically as well as horizontally. With this feature, you can split a page of source code into two horizontal or vertical panes. Material can be copied, pasted, cut, or changed in either pane, or copied and pasted from one pane to the other. Changes in one pane of code are instantly reflected in the other.

## Using the Properties panel

The new HTML and CSS buttons in the Properties panel are explored in an exercise. Emphasize that different properties are displayed in the Properties panel, depending on whether HTML or CSS is selected. For certain properties, students will have to learn when to choose the HTML button and when to choose the CSS button. Explain that if they choose the CSS button and are unable to perform a task in the Properties panel—to select an HTML format such as an `<h1>`, for example—they should choose the HTML button.

## Previewing the page

Conclude the lesson with a look at Live view and helping the students set up Dreamweaver for viewing pages in a browser. Using Live view saves time and is a wonderful new feature, but students still need to know how to view and test pages in various browsers. Help them set up Dreamweaver to work with every browser they may have on their systems.

# Lesson 2: Working with Cascading Style Sheets

In Lesson 2, students make constant use of the Dreamweaver style tools and jump into using CSS skills to control the appearance of a page. Students attach external style sheets (including one for print use), create CSS rules, apply rules, use descendent selectors and custom classes, and add structural areas to the page (`<div>` tags) that will be assigned to particular style rules.

## Attaching an external style sheet

The CSS rules that were used in the document head in Lesson 1 are saved in an external style sheet in Lesson 2. Students link to those rules using a page of prepared content. This is a good time to explain when it is appropriate to use style rules in an internal style sheet. For example, you might mention that rules in an internal style sheet apply only to the document they are in, but that designers often like to work with internal styles during their initial design phase. In a real-world situation, a design may have to be modified or approved by many people, and retaining everything in a single file is a desirable workflow. Later in the book, students will export styles from the document head into an external style sheet.

## Creating new CSS rules

In the style sheets that Dreamweaver uses in its CSS layouts, almost every CSS rule you add will be a descendent selector. This is because the layouts all assign a class to the body selector, and the new rules descend from that parent. The layouts were designed this way so that more than one layout (for example, both two- and



three-column pages) could be used in a single site and with a single style sheet. Examples throughout the book show how to shorten some of the selector names (making them less specific). The first rule that students create, `#mainContent h1`, is an example of a less specific selector name than the selector that Dreamweaver suggests.

Students are shown how to use the color picker to select a color from the header background image, but it's a bit tricky to locate the exact shade of blue (`#345FA3`) suggested using the color picker. Typing the number in the color field is encouraged, and it demonstrates that data fields in the CSS Rule Definition dialog box are all editable and accept typed input. This might be a good time to talk about the way hexadecimal color names work. Note that Dreamweaver now uses the shorthand hexadecimal numbers, such as `#000`, `#FFF`, `#666`, for web-safe colors. You may want to explain that a number like `#345FA3` cannot be reduced to shorthand, but a number with matching pairs, such as `#000066`, can be expressed as `#006`.

The basics of the box model are touched on when the students create a new CSS rule. This is a good time to briefly explain the box model. A good resource of information is the *CSS Box Model Hierarchy* ([www.hicksdesign.co.uk/boxmodel](http://www.hicksdesign.co.uk/boxmodel)).

## Setting up custom classes

Students first create a class. In the next exercise, they apply this class to page elements. Emphasize that selectors based on tags such as `#mainContent h1` get applied automatically to any `<h1>` in the `#mainContent` area, but the class designation must be added to the HTML element before the CSS rule has any effect.

## Applying styles

Students click the CSS button in the Properties panel to find classes that have been defined in the external style sheet, and then apply them to the page.

## Applying custom classes to page divisions

Students learn how to add `<div>` tags to a page. They highlight areas of content and add a surrounding `<div>` that is assigned to a class. This is one of the most basic page-design techniques: creating a containment structure for a block of content. Repeating the process several times will help students retain the information and apply it to future situations.

## Floating images

Students use `float` to move images to the left side of the containing element and learn a bit about the frequent need to use `clear` with floated elements. At this point, if you'd like to discuss floats, a good reference source is *Digital*

*Web Magazine*. See “Web Design 101: Floats” ([www.digital-web.com/articles/web\\_design\\_101\\_floats](http://www.digital-web.com/articles/web_design_101_floats)).

## Examining selectors in the CSS Rules panel

This is a good time to help students explore how the three related panes in the CSS panel can be used to create, modify, and debug CSS rules.

## Creating style sheet for other media types

You might explore different media types and their uses. Then, you could explain concepts about the cascade. Since students will have attached two linked style sheets to a page, they will see how the `mystyles.css` rules affect the printed page when the `print.css` rules are added to the cascade. A helpful resource is the W3C’s “Assigning property values, Cascading, and Inheritance” page ([www.w3.org/TR/REC-CSS2/cascade.html](http://www.w3.org/TR/REC-CSS2/cascade.html)), which explains the cascade, and will also introduce students to the W3C and web standards.

Be sure that students know how to work with the Style Rendering toolbar, which is useful when designing for print and handheld media.

## Hiding unwanted page areas

This exercise teaches the students about `display: none`. You might want to tell students that `display: none` completely removes an element from the layout, whereas `visibility: hidden` removes an element from view but leaves it in the layout.

## Ensuring that colors work in print

Creating colors that will work on a printed page is a good opportunity to explore color issues. The color contrast analyzer at Juicy Studio (<http://juicystudio.com/services/colourcontrast.php>) can help you discuss color contrast, colorblindness, and other accessibility issues related to color.

# Lesson 3: Working Text, Lists, and Tables

In this lesson, students work with various HTML elements related to text. They create several types of text, check spelling, and work with the new Spry Tooltip widget.

## Importing text

To help students understand that web designers often receive content from multiple sources, they’ll practice importing text from a `.txt` file. When the text has been placed on the page, students use the Properties panel to format it. You could

enhance this lesson by providing students with a page in a Word .doc file and asking them to import it and use Dreamweaver's Clean Up Word tools to get rid of the unnecessary code and formatting.

## Creating headings, lists, and block quotes

Explain how the hierarchy of `<h1>`, `<h2>`, `<h3>`, and so on normally corresponds to the hierarchy of information on a web page. The most important page content should be given an `<h1>` heading, and so on down the information hierarchy.

You might want to assign additional activities creating lists that include nested lists to give the students more practice.

Regarding block quotes, be sure that students know that text can be indented with CSS rules, and that `<blockquote>` should not be used unless the material is actually quoted material. The discussion of `<blockquote>` could lead to an interesting lecture on the meaning and importance of semantic HTML. A good resource is "POSH: Plain Old Semantic HTML" ([www.456bereastreet.com/archive/200711/posh\\_plain\\_old\\_semantic\\_html](http://www.456bereastreet.com/archive/200711/posh_plain_old_semantic_html)).

## Styling a table with CSS

Students learn how to set table width, use padding (CSS padding replaces the older HTML attribute for cell padding), and create and collapse borders for the table. Dreamweaver does not include `border-collapse` in the CSS Style Rule dialog box, so students get their first experiences in manually adding a CSS property and value to a style sheet using the CSS Style panel. You might want to explain that Dreamweaver accepts user input for properties and values and handles them properly, as long as they are valid CSS.

A good information resource on styling tables is "Top 10 CSS Table Designs" ([www.smashingmagazine.com/2008/08/13/top-10-css-table-designs](http://www.smashingmagazine.com/2008/08/13/top-10-css-table-designs)). You can assign an extra exercise asking the students to complete a simple data table on their own.

## Spell-checking a document

You may want to provide students with a page of Dreamweaver content that includes more misspelled words than are in this exercise. Emphasize that misspelled words on a website can be sufficient reason for most people to question the professionalism and competence of the site and the information it contains.

## Using the Spry Tooltip widget

If you were frustrated when creating tooltips using `<acronym>` or `<abbr>`, you'll be delighted with this new Spry widget. You can now style the tooltip and customize a bit of JavaScript (JS) that controls it.

Students receive their first exposures to the Adobe Spry Framework for Ajax, so make sure they understand the dependent files for CSS and JS that are added to the site SpryAssets subfolder. You may want to point out that CSS and JS are automatically linked to the page and demonstrate by having the students view the links in the document head. You may also point out that dependent files are listed across the top of the document, another new feature in Dreamweaver CS4.

## Finding and Replacing text

Make sure that students understand the value this tool has for all their future work. It can find and replace text or source code in a file, a folder, or an entire site.

## Lesson 4: Working with Images

In Lesson 1, students added images in the HTML and also used a CSS background image. In this lesson, they expand on those methods. Additionally, they work with Dreamweaver's workflow tool, Bridge, and integrate Photoshop with Dreamweaver. A new feature in Dreamweaver, the Photoshop Smart Object, is explored in this lesson.

### Inserting an image

Image placeholders are used in this lesson to help students locate the positions for the images. The first image placement is a straightforward exercise in adding an `<img>` element to the page. This is a good time to discuss the importance of alt text and the principles behind writing good alt text. Two helpful alt text resources are "Writing good ALT text" ([www.gawds.org/show.php?contentid=28](http://www.gawds.org/show.php?contentid=28)) and "How to Write Good Alt Text for the Images on Your Web Page" ([www.ehow.com/how\\_2110225\\_alt-text-images-web-page.html](http://www.ehow.com/how_2110225_alt-text-images-web-page.html)).

### Using background images

The background image used in this lesson creates a dividing line between the sidebar and the main content. This technique can often be used to create a shaded sidebar rather than a thin line. The technique is known as *faux columns*. They were first described in "Faux Columns" at A List Apart ([www.alistapart.com/articles/fauxcolumns](http://www.alistapart.com/articles/fauxcolumns)). Students will quickly grasp the concept for fixed-width designs, but it's easily understood for liquid designs too. You might want to perform a quick demo or lesson on liquid faux columns. A good information resource is "Creating Liquid Faux Columns" ([www.communitymx.com/content/article.cfm?cid=afc58](http://www.communitymx.com/content/article.cfm?cid=afc58)).

## Using the Properties panel with images

Dreamweaver has a built-in graphics engine accessed via the Properties panel. It's like having Fireworks built into Dreamweaver. Spend time helping students realize that they can crop, optimize, and sharpen images within Dreamweaver.

Regarding optimization and the Image Preview dialog: This is a good time to help students understand the basics of GIF, JPEG, and PNG image formats. You might explain how to optimize a GIF by reducing colors compared with optimizing a JPEG by decreasing quality. This is also the perfect time to explain transparency in GIF and PNG formats.

## Copying and pasting from Photoshop

Because Photoshop is so closely integrated with Dreamweaver, some great timesavers are now available. For example, you can copy and paste individual layers from a Photoshop (PSD or other format) file into Dreamweaver, and this will make it easy to perform any necessary adjustments.

## Accessing Adobe Bridge

Bridge is a separate application that simplifies workflow between all the Creative Suite CS4 applications. In this exercise, students choose from images in a specific folder; but in a real-world situation, a user could browse through and choose any available images.

## Using Photoshop Smart Objects

Photoshop Smart Objects are new to Dreamweaver CS4. A Smart Object in Dreamweaver is an image asset placed on a web page that has a live connection to an original Photoshop (PSD) file. In Dreamweaver's Design view, a Smart Object is indicated by an icon in the upper-left corner of the image. A single image can be used in many places, in varying sizes and configurations, while remaining tied to a Photoshop document. You may want to show students a video about Smart Objects that is available at Adobe TV ([www.adobe.com/go/lrvid4043\\_dw](http://www.adobe.com/go/lrvid4043_dw)).

# Lesson 5: Working with Navigation

In Lesson 5, students work with relative and absolute links, create site navigation for the travel site using a Spry Menu Bar widget, and add tabbed panels to an existing page in the travel site.

## Linking to internal site pages

Because the site created in these exercises has a very simple structure, you may want to demonstrate a more complex site that uses several subfolders to help students grasp the notion of file paths and relative URLs. Be sure they understand the difference between a link value of `../../index.html` and `/greece/paros.html`.

This is also a good time to teach the difference between setting up a site to work with URLs relative to the document and with URLs relative to the site root. Be sure students know that not all servers work with links that start from the site root.

## Creating an external link

In addition to the *http* and *mailto* protocols mentioned in the text, you can mention other Internet protocols such as *ftp*, *telnet*, *irc*, and so on.

## Targeting links within a page

Linking to specific locations on a page has become easier with the advent of CSS layouts and their heavy use of the ID for structural elements. This simplifies targeting a link with an ID and makes the process less code intensive than it was with named anchors.

## Inserting Spry Menu Bars

Styled lists are now the default for menu items, with rollover images and graphical menus going the way of the dodo. Lists are inherently accessible (they're nothing more than text), and the Spry Menu Bar maintains that accessibility by fully displaying as text in browsers that don't have CSS or JavaScript enabled.

The menu bar that students create in this lesson is applied to all the travel site pages in a later chapter.

Working with the Spry Menu Bar widget is probably the most common use that students will make of the Adobe Spry Framework for Ajax after they complete the course. So, you may want to design additional exercises that let students practice with horizontal displays, lists nested three and four levels deep, and other menu bar variations.

A good resource for a discussion of styled lists is Listamatic (<http://css.maxdesign.com.au/listamatic>). Notice that these styled lists don't include the added JavaScript functionality that comes with Spry.

## Customizing the appearance of the Spry Menu Bar

It will be helpful to spend time customizing the CSS for Spry Menu Bars so that students become familiar with what each CSS selector controls.

## Inserting Spry tabbed panels

Tabbed panels are not truly navigation, although they are placed in Lesson 5. They can be thought of as navigation within a limited content block on a page. Shopping sites often used tabbed panels to concentrate an array of information in a small space. For example, a computer site might show Options, System Requirements, Specifications, and Add-ons in a tabbed arrangement for the shopper.

Make sure that students have a wide page display of the document window, so that the copy and paste operation will be easier for each individual `<div>` that gets pulled into a tabbed panel.

## Checking your page

Dreamweaver's built-in tools for checking links, browser compatibility, and accessibility are important for students to master. In addition, make sure they understand the built-in validation tools. Make sure that students know how to use the Adobe online help for any errors found in the browser-compatibility check.

# Lesson 6: Adding Interactivity

The interactive elements in this lesson involve Dreamweaver behaviors, which have been around for some time; and the newer Spry widgets, such as effects, data sets, and accordions.

## Learning about Dreamweaver behaviors

This section gives the student a general background in what behaviors can do and how they work. Make sure that students know how to pick an appropriate event handler, such as `onClick` or `onMouseOver`, and how to edit the parameters of any behaviors they use.

## Using a Spry effects behavior

Students may be amused by the more over-the-top Spry effects, so be sure to remind them that they are most effective when used in moderation.

## Learning about XML data

This is another informational section meant to help students with a very basic grasp of XML. You may want to expand on the topic by explaining how XML differs from HTML.

## Importing XML data with Spry

This section is rather complicated and includes a large number of steps, so be alert for students who may make a small error. The simple exercise that students complete in this lesson barely scratches the surface of XML data sets. You might want to create some extra credit or extra practice exercises that use other options when importing data sources.

Make sure that students understand why the display they see in the Document window contains placeholder names and not the actual data.

## Creating Spry accordions

Frankly, the page may be more attractive without the Spry accordions. If you agree with that, you may have students learn how to make the accordions, but then use the previous version of the page when they complete the site.

Carefully watch as students perform the cutting and pasting steps, and encourage them to watch the tag inspector and the Code and Design view to ensure that they are capturing exactly what they need when the transfer into the accordion panel is completed.

## Customizing a Spry accordion

As when customizing any other Spry CSS style sheet, the majority of the work involved occurs when you try to match the selector names with the changes you want to make. The time you spend helping students grasp Dreamweaver's CSS selector-naming conventions, such as `.AccordionPanelTabHover`, will be valuable any time they use Spry in the remainder of the course.

Students create a new rule using the “Add property” link in the CSS panel. Be sure they are comfortable with the way this part of the Dreamweaver CSS interface works.

## Lesson 7: Creating a Page Layout

This is a crucial lesson. Any extra time that students spend using the techniques in this chapter will be very beneficial. The lesson teaches concepts about working with `divs` that will help students understand the underlying structure of a CSS layout.

A helpful reference for understanding CSS layouts is “The Early Bird Catches the CSS: Planning Structural HTML” ([www.wise-women.org/tutorials/cssplanning/index.shtml](http://www.wise-women.org/tutorials/cssplanning/index.shtml)).



## Creating layout-styled <div> elements

A lot happens in this section. Students add <div> tags, assign IDs, place <div> tags in the document flow, and write CSS rules. The content is broken into small bits, as follows:

## Creating the page and defining the body tag

This is a good opportunity to discuss inheritance. Rules created for the <body> tag, such as color or font family, will be inherited by every other element on the page unless specifically overwritten.

## Defining the outer wrapper

Make sure that students understand the purpose of a wrapper <div>. A good resource is “Writing the Wrapper Selector” ([www.adobe.com/devnet/dreamweaver/articles/css\\_concepts\\_pt3\\_04.html](http://www.adobe.com/devnet/dreamweaver/articles/css_concepts_pt3_04.html)).

## Setting up the primary divisions

This is a good time for a discussion of the planning process for a website, including tasks such as wireframing, choosing the structural blocks you’ll need, selecting the number of columns, identifying the navigation areas, and deciding whether to use a header and footer. This resource, “Wire Frame Your Site” ([www.sitepoint.com/article/wire-frame-your-site](http://www.sitepoint.com/article/wire-frame-your-site)), discusses wireframes for both sites and pages.

## Adding content areas

In addition to the method described in these exercise, you may want to teach students how to place an insertion point in the document by using the Insert Div button to position a <div>. Dreamweaver always provides more than one way to accomplish any given task.

Encourage students to remember the final layout they previewed as they work through this section so that the options will make more sense to them.

## Inserting placeholders

You may want to tie this section in to the concept of wireframing. The completed layout is a type of wireframe, as are all of the built-in CSS layouts that come with Dreamweaver.

## Looking at Code view

This section gives students a look at options that can make finding and using code easier.

Have them try out the Code Navigator feature in a few places other than those suggested in the exercise. This will help them grasp the value that can be gained using this new Dreamweaver feature.

## Exporting CSS Styles

Being able to export an internal style sheet and clean up any lingering style tags is an important skill. In a team situation, or when requesting help from an online support group, a new design generally contains everything in one document, including the CSS. Later, it will be exported to an external style sheet so that it can be used on multiple pages.

## Using the ruler and guides

If you haven't yet discussed alignment on a page, this is a good time to do so. A helpful article is "How C.R.A.P is Your Site Design?" ([www.thinkvitamin.com/features/design/how-crap-is-your-site-designprint](http://www.thinkvitamin.com/features/design/how-crap-is-your-site-designprint)).

Ensure that students know how to use both the horizontal and vertical rulers and guides.

## Checking the layout

Once again, students run browser compatibility and validity checks on a page. You can't overstate the importance of this chore. It can make the difference between a site that works and one that is buggy.

# Lesson 8: Working with Forms

When students complete this lesson, they should be able to design an accessible web form that will help them gather useful information from site visitors.

A major concern when using forms is finding a good script. A good resource is a forms article from the Opera Web Standards Curriculum project, "HTML forms—the basics" (<http://dev.opera.com/articles/view/20-html-forms-the-basics>).

## Learning about forms

This is a general information section that explains how a form works and what role a script plays in its submission. It reviews the various form elements that will be used in the lesson.

## Adding a form to the page

In this exercise, the students add the initial `<form>` element to the page and configure an action. Make sure the students understand that the Thank You page

of HTML that the form triggers when submitted is not a script and doesn't submit any data.

This might be a good time to talk about the generic ID names that Dreamweaver assigns to form elements, such as *form1* or *spryTextField1*. Explain that students can change the Name field in the Property inspector to give any form element a more useful ID that would convey more meaningful information to the developer when the form was submitted in a real-world situation.

## Inserting text form elements

A number of text form elements are found in this section, which is broken up into smaller units, as follows:

### Setting preferences for accessible forms

Accessibility preferences should all be selected before working on forms. This ensures that Dreamweaver asks for label information when inserting any form element.

### Using text fields

Text fields, also called *input fields*, are used in several ways in these exercises. Make sure that students understand that input fields such as user name, password, e-mail, and so on are all variations on the same basic input element. The differing validation requirements placed on the text fields in these exercises are accomplished using JavaScript via the Spry Validation widgets.

### Including Spry text fields

Ensure that students grasp the basic underpinnings of CSS and JavaScript that come with the Spry Validation widgets.

The exercises all accept the default error messages, so this is a good opportunity to discuss the control that a student has over choices such as error messages and the CSS rules that apply to their appearance. Students should also understand their choices regarding when to validate: `onBlur`, `onChange`, and `onSubmit`.

### Creating a fieldset

Encourage students to use Code and Design view when adding fieldsets, so they know exactly what they are including. The Spry Form Validation widgets used in these exercises add additional code to the page (the error messages are stored in a `<span>` that isn't visible unless needed) that may not always get properly selected when choosing what to include in a fieldset in Design view.

A helpful resource for fieldsets is “Accessible, stylish form layout” ([www.themaninblue.com/writing/perspective/2004/03/24](http://www.themaninblue.com/writing/perspective/2004/03/24)).

## Creating user name and password text fields

Creating requirements for user name and password demonstrate the value of using Spry Form Validation widgets.

In the exercises, students add a paragraph explaining the requirements. Be sure that students realize this can also be done using a custom error message that tells users more about what they are expected to do.

## Incorporating text areas

Students should learn how to control the choices for width and number of lines in a text area, as this is not emphasized in the exercise. Let them know that it isn’t always necessary to add an initial value to a text area.

## Creating radio buttons

Help students to understand the difference between radio buttons and check boxes. Note that only one radio button in a group can be selected, but several check boxes in a group can be selected at once.

The new Spry Validation Radio Group saves time compared with the way radio buttons were added in previous versions of

Dreamweaver, although it is still possible to do it using the older method.

In the exercises in the book, label and value are often the same. This is not a requirement. Explain to the students that the label text can be anything meaningful to the website user, while the value is a field identifier that goes back to the developer when the form is submitted.

## Inserting check boxes

The Spry Checkbox Group widget is another time saver in this version of Dreamweaver. It adds a group name and assigns generic IDs to all the check box items.

## Working with lists

In this exercise, students create a select menu. You may want to design an additional practice exercise of your own that uses the list style of this form element, focusing on allowing multiple choices from a list.

## Adding a submit button

You may want to explore the common values that might be placed on submit buttons, such as Search or Confirm Order, and demonstrate by showing the class a number of online forms.

## Styling Forms

A helpful supplemental article is “Styling Forms with CSS—Part 1: Accessibility and Form Element Styles” ([www.adobe.com/devnet/dreamweaver/articles/styling\\_forms\\_pt1.html](http://www.adobe.com/devnet/dreamweaver/articles/styling_forms_pt1.html)). You may also want to direct students to an article about browser default styles for forms, “Styling form controls with CSS, revisited” ([www.456bereastreet.com/archive/200701/styling\\_form\\_controls\\_with\\_css\\_revisited](http://www.456bereastreet.com/archive/200701/styling_form_controls_with_css_revisited)).

You can explain additional CSS rules that can alter the appearance of forms, including adding background colors or images to fieldsets, changing font rules for labels, styling margins and padding for form elements, and so on.

Also point out that CSS rules for a form do not *have to be* isolated in a separate style sheet. They can be part of a site’s main style sheet. This exercise allows students to experience creating a new style sheet and linking it to the page, but it also shows how to incorporate form styles in the main site style sheet.

## Lesson 9: Working with Flash

A Flash movie and a Flash video are provided. Students use Dreamweaver to add them to the page that serves as the travel site home page.

### Adding a Flash movie to a page

Students must have the movie subfolder correctly placed in their main folder before they begin working on these exercises.

Students navigate to the movie subfolder to locate the .swf file needed in this exercise.

You may want to provide more details about the ways Flash is now embedded in Dreamweaver CS4. In CS4, Dreamweaver has been brought into compliance with web standards for <object> elements. This is briefly explained in a sidebar.

### Adding Flash video to a page

The .flv file is located in the same folder as the .swf file used in the previous exercise. Students should realize that the additional JavaScript file and the .swf installer file needed to make Flash work properly in the browser must be uploaded to the server.

Don't forget to mention that the Properties panel includes a Play button that allows the student to test the video within Dreamweaver.

## Lesson 10: Increasing Productivity

This lesson addresses three important productivity tools in Dreamweaver: templates, Library items, and server-side includes.

### Creating a template

In this exercise, students save one of the travel site pages as a template. When creating a template in this way, links to style sheets and other material in the document head are already placed in the template. Point out that any changes to documents linked to the template, such as the style sheets, will be reflected in pages built using the template.

### Inserting editable regions

Stress the importance of anticipating what needs to be editable in a template and the importance of creating editable regions to accommodate content needs in future child pages made from the template.

Point out that templates are a timesaver for the person who works alone, but editable regions protect important page elements from changes when multiple teams are adding content to child pages.

### Updating a template

If you can, demonstrate a site built with templates from the start of the design process and show students how Dreamweaver updates each child page when a template file is updated.

### Producing child pages

Students may find it easier to create a child page from the Template category of the Assets panel. In this section, reinforce the concept that several ways may be used to accomplish any given task in Dreamweaver and that students should feel free to experiment with other methods to find their preferred techniques.

Adding images to an editable region is a good reminder that any type of content can be added to an editable region.

### Using Library items

Library items have all the benefits of templates, but with fewer restrictions when changing individual pages.

## Creating a Library item

If possible, demonstrate creating and using a Library item that does not involve a Spry widget. Point out that not all Library items are as complex to create and add to pages as the site-navigation example in this exercise.

Remind students that the HTML snippet stored in a Library item is not linked to any style sheet, but when it is inserted in a page, it is acted on by any linked CSS in that page.

Stress the need to ensure that Library items contain only the part of a page that makes up the snippet of HTML you're adding, and no other tags.

## Inserting a Library item

Observe that for most Library items, simply setting an insertion point on a page and clicking the Insert button is the entire process.

## Linking to the dependent files

Be sure that students understand why the extra steps are needed in this exercise.

If some students are comfortable working in Code view, you might suggest that copying and pasting the links to the CSS and JS files is equally effective.

Point out that the method of linking to an external JavaScript file used in this exercise could be used for any scripts that students might find in JavaScript libraries on the Internet or in other sources of scripted behaviors.

## Updating Library items

The Library item update in this exercise adds a JavaScript operator. When looking at the Library item in Design view after the change, no apparent change is visible. You might want to create a demonstration in which some visible change in text is made to an updated Library item.

Explain that the JavaScript operator used in this exercise is normally added to a page at the very end to make the page download faster. If you would like, students could manually paste the operator into the page code for each page instead of moving it into the Library item.

## Adding the menu

Now that the Library item is complete, it needs to be inserted into each page. If some students struggle with the insertion point, suggest that they select whatever is currently at the top of the sidebar and press the Left Arrow key to move the insertion point to the immediate left of the element already placed there.

## Using server-side includes

It would be helpful to show students server-side includes that are in use on a server, if your lab setup allows it.

A helpful resource is “Apache Tutorial: Introduction to Server Side Includes” (<http://httpd.apache.org/docs/1.3/howto/ssi.html>).

## Creating server-side includes

You cannot overstate that no unneeded page elements are allowed in an SSI, especially additional `<html>` or `<body>` tags. Encourage students to check Code and Design view when creating SSIs to and watch for unwanted material.

Students should be instructed that *includes* is not the only option for a folder name. It might be *inc* or any other logical name. Stress that in a real-world situation, the includes folder would be in use by all of the site files and located at the site root.

## Inserting server-side includes

Be sure students realize that server-side includes appear in the Design view page only if the Dreamweaver option to view it was set in Preferences.

Students should know that individual hosting companies and server types may have specific rules about SSIs that can be found in the support pages for those providers, such as the “Support Resources” section of the Pair Networks website ([www.pair.com/support](http://www.pair.com/support)).

Also mention that programming languages such as PHP provide ways to use include files. If you have access to a Wordpress blog template, students might be interested to see how the blog is built using a series of PHP include files.

## Adding the server-side include to site pages

This section is basically a housekeeping exercise meant to add the SSI to several pages. Stress that, as with templates and Library items, the decision to use SSIs should come early in the design process to reap maximum time-saving and maintenance benefits when building the site.

## Updating server-side includes

Here is where students learn the value of uploading just one include file to the server rather than changing dozens or hundreds of pages in a template or Library item update. When students realize how easy it is to update and maintain SSIs, they may never slow down again.

Tell students that when an SSI is opened for editing in Design view, unwanted tags may creep in. Urge students to keep an eye on the Code view when updating an SSI.



# Lesson 11: Publishing to the Web

## Defining a remote site

Depending on the setup you have in your lab, you will probably use only one of the methods available in Dreamweaver to connect to a remote site.

## Setting up a remote FTP site

If students are not able to use FTP, suggest some no-cost hosting options that will allow FTP connections. A source for free hosting information is Free Web Hosting ([www.free-webhosts.com](http://www.free-webhosts.com)). Make sure the free hosting provider allows FTP.

For the FTP connection, students should know how to fill in the remote site information. Tell them that most web hosting companies provide all the information that is needed when an account is opened and storage space is set aside for that new account. Also make sure that students understand that most web hosts have extensive support sites explaining how to work with their server.

Stress the fact that any Put or Get operation overwrites the files at the destination. Students need to be especially careful not to get a file from a server that will overwrite a file on their local site that has been changed but not yet uploaded.

## Establishing a remote site on a local or network server

Many computers now come with built-in local web servers. Even if your classroom setup doesn't use this connection method, you may want to instruct students how to use it.

## Cloaking folders and files

Dreamweaver CS4 allows you to cloak individual files. Explain to students why some file items may not need to be uploaded to the server—particularly items like template and Library item files that are physically included in the HTML page when it is uploaded.

Also explain that in a real-world situation, a site root might contain related material, such as letters, contracts, invoices, or e-mail, that does not belong on the server.

## Putting your site online

The most important concept to stress here is that the file structures of the remote and local sites must be identical. Also, encourage students to change filenames and file structures within Dreamweaver rather than using the computer's file system. That way, Dreamweaver will manage links and other file relationships, and keep pages in working condition.

Have students examine the site structure that Dreamweaver creates on the remote site. They should notice folder names, in particular. Explain how these folder names become part of the page URL. For example, Dreamweaver creates a mirror image of the local site on the remote server, which becomes part of the URL. If a modified version of the example URL—<http://www.yourserver.com/lesson11/test-page.html>—does not open the page in a browser, have them re-examine the remote site structure to determine what the URL should be.

Look at some actual websites and discuss what a URL such as <http://www.adobe.com/products/dreamweaver/> means in terms of site structure. In this example, you can see that the server [adobe.com](http://www.adobe.com) contains folders for products and that within the products folder there is a dreamweaver folder.