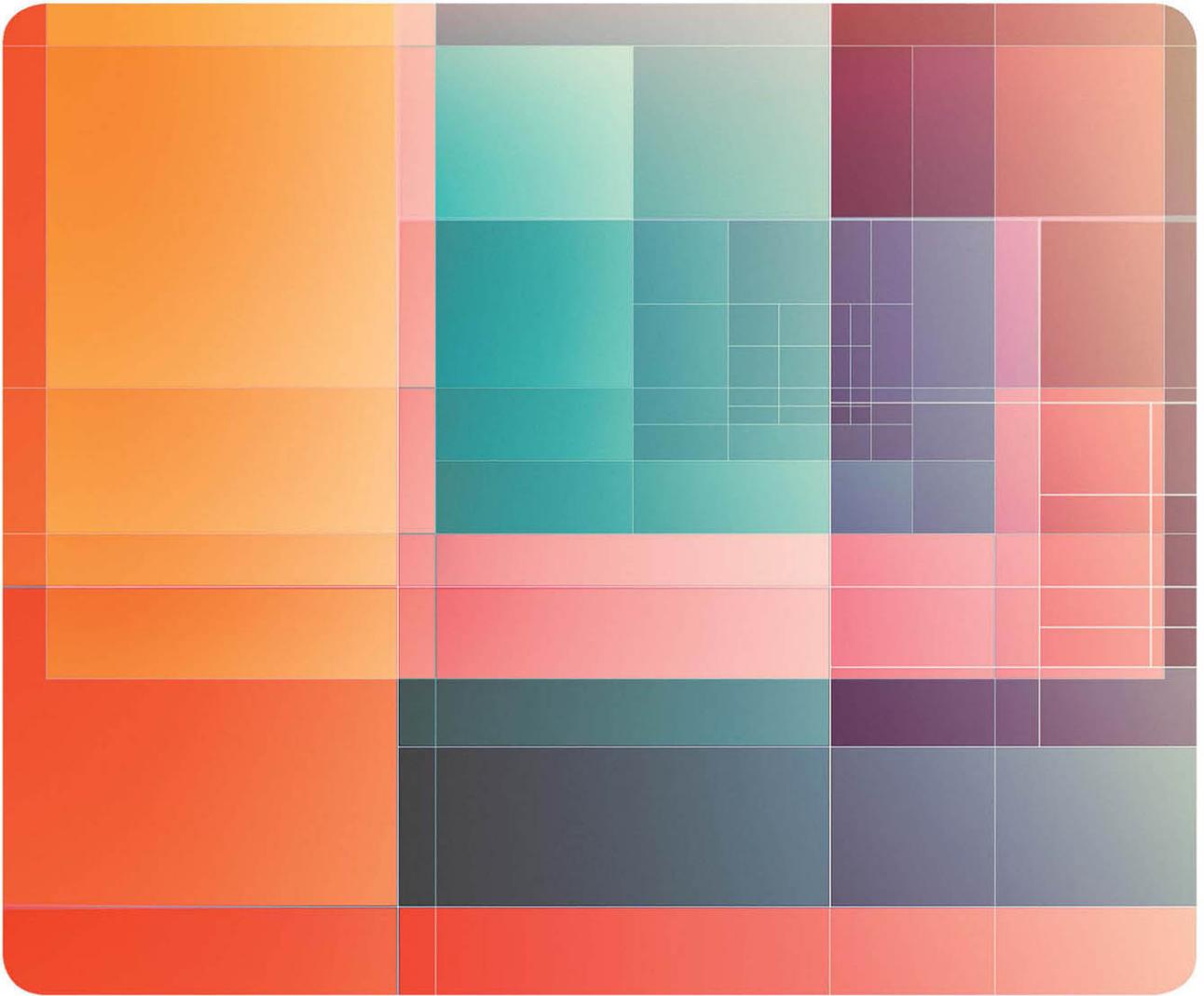




Adobe Dreamweaver

2021
release



Classroom in a Book[®]

The official training workbook from Adobe

Jim Maivald

FREE SAMPLE CHAPTER

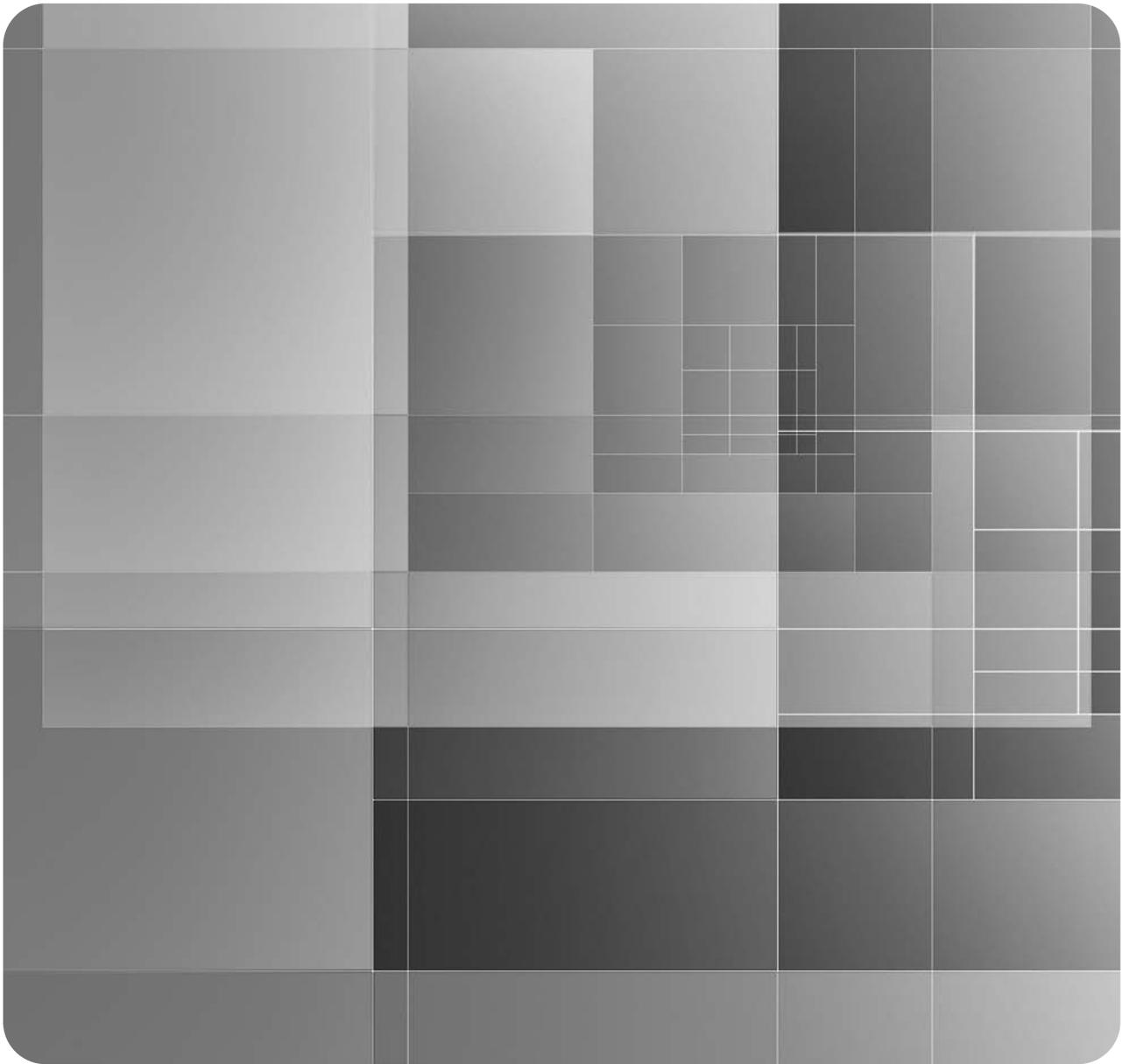
SHARE WITH OTHERS





Adobe Dreamweaver

2021
release



Classroom in a Book[®]

The official training workbook from Adobe

Jim Maivald

Adobe Dreamweaver Classroom in a Book® (2021 release)

© 2021 Adobe. All rights reserved.

Adobe Press is an imprint of Pearson Education, Inc. For the latest on Adobe Press books, go to www.adobepress.com. To report errors, please send a note to errata@peachpit.com. For information regarding permissions, request forms and the appropriate contacts within the Pearson Education Global Rights & Permissions department, please visit www.pearson.com/permissions.

If this guide is distributed with software that includes an end user license agreement, this guide, as well as the software described in it, is furnished under license and may be used or copied only in accordance with the terms of such license. Except as permitted by any such license, no part of this guide may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Adobe. 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. Adobe 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, Adobe logo, Classroom in a Book, Creative Cloud, Creative Cloud logo, Dreamweaver, Illustrator, InDesign, and Photoshop are either registered trademarks or trademarks of Adobe in the United States and/or other countries. Adobe product screenshots reprinted with permission from Adobe.

Apple, Mac OS, macOS, and Macintosh are trademarks of Apple, registered in the U.S. and other countries. Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the U.S. and/or other countries. JavaScript® is a trademark or registered trademark of Oracle in the U.S. and other countries. jQuery is a trademark of the jQuery Foundation. All other trademarks are the property of their respective owners.

Unless otherwise indicated herein, any third-party trademarks that may appear in this work are the property of their respective owners and any references to third-party trademarks, logos or other trade dress are for demonstrative or descriptive purposes only. Such references are not intended to imply any sponsorship, endorsement, authorization, or promotion of Pearson Education, Inc. products by the owners of such marks, or any relationship between the owner and Pearson Education, Inc. or its affiliates, authors, licensees or distributors.

Adobe Inc., 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 Inc., 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.

Writer: James J. Maivald

Executive Editor: Laura Norman

Development Editor: Robyn G. Thomas

Technical Reviewer: Candyce Mairs

Senior Production Editor: Tracey Croom

Copyeditor: Scout Festa

Composition: Kim Scott, Bumpy Design

Proofreader: Kim Wimpsett

Indexer: John Lewis, JJ Indexing

Cover Illustration: RETOKA

Interior Designer: Mimi Heft

ISBN-13: 978-0-13-687523-9

ISBN-10: 0-13-687523-8

WHERE ARE THE LESSON FILES?

Purchase of this Classroom in a Book in any format gives you access to the lesson files you'll need to complete the exercises in the book.

- 1 Go to www.adobe.com/DreamweaverCIB2021.
- 2 Sign in or create a new account.
- 3 Click Submit.

● **Note:** If you encounter problems registering your product or accessing the lesson files or web edition, go to www.adobe.com/ support for assistance.

- 4 Answer the questions as proof of purchase.
- 5 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.

● **Note:** If you purchased a digital product directly from www.adobe.com or www.peachpit.com, your product will already be registered. However, you still need to follow the registration steps and answer the proof of purchase question before the Access Bonus Content link will appear under the product on your Registered Products tab.

CONTENTS

GETTING STARTED	1
 About Classroom in a Book.....	1
TinyURLs	1
Prerequisites.....	2
Conventions used in this book	2
<i>Bolded text</i>	2
<i>Code font</i>	2
<i>Strikethrough</i>	3
<i>Missing punctuation</i>	3
<i>Element references</i>	3
Windows vs. macOS instructions	3
Installing the program	4
Updating Dreamweaver to the latest version.....	5
Online content	6
<i>Lesson files</i>	6
<i>Web Edition</i>	6
<i>Accessing the lesson files and Web Edition</i>	6
Recommended lesson order	7
<i>Bonus material</i>	8
On first launch.....	8
Choosing the program color theme	9
Setting up the workspace	9
Defining a Dreamweaver site.....	11
Checking for updates	14
Additional resources	15
<i>Adobe Authorized Training Centers</i>	15

1	CUSTOMIZING YOUR WORKSPACE	16
	Touring the workspace.....	18
	Using the Start Screen	19
	<i>Quick Start</i>	19
	<i>Starter Templates</i>	20
	<i>Create New and Open</i>	21
	Exploring the New Feature guides.....	21
	Setting interface preferences.....	22
	Switching and splitting views	25
	<i>Code view</i>	25
	<i>Design view</i>	25
	<i>Live view</i>	26
	<i>Split view</i>	27
	<i>Live Code mode</i>	28
	<i>Inspect mode</i>	28
	Selecting a workspace layout.....	29
	<i>Standard workspace</i>	29
	<i>Developer workspace</i>	30
	Working with panels.....	30
	<i>Minimizing panels</i>	31
	<i>Closing panels and panel groups</i>	32
	<i>Dragging</i>	33
	<i>Floating</i>	33
	<i>Grouping, stacking, and docking</i>	34
	Personalizing Dreamweaver.....	35
	<i>Saving a custom workspace</i>	35
	<i>Working with the Extract panel</i>	36
	<i>Loading Photoshop documents into Extract</i>	36
	Working with toolbars	37
	<i>Document toolbar</i>	37
	<i>Standard toolbar</i>	38
	<i>Common toolbar</i>	38
	Creating custom keyboard shortcuts	38
	Using the Property inspector	40
	<i>Using the HTML tab</i>	40
	<i>Using the CSS tab</i>	41

3 CSS BASICS 68



What is CSS?	70
HTML vs. CSS formatting	70
HTML defaults	72
<i>HTML5 defaults?</i>	73
<i>Browser antics</i>	74
CSS box model	75
Applying CSS styling	77
<i>Cascade theory</i>	78
<i>Inheritance theory</i>	80
<i>Descendant theory</i>	80
<i>Specificity theory</i>	81
<i>Code Navigator</i>	82
<i>CSS Designer</i>	87
Multiples, classes, and ids, oh my!	93
<i>Applying formatting to multiple elements</i>	93
<i>Using CSS shorthand</i>	93
<i>Creating class attributes</i>	94
<i>Creating id attributes</i>	95
<i>CSS3 features and effects</i>	95
<i>CSS3 overview and support</i>	97
Review questions	98
Review answers	99

4 WORKING WITH CODE 100



Creating HTML code	102
<i>Writing code manually</i>	102
<i>Writing code automatically</i>	106
Working with multicursor support	110
Commenting your code	112
Working with CSS preprocessors	113
<i>Enabling a preprocessor</i>	113
<i>Creating the CSS source file</i>	116
<i>Compiling CSS code</i>	120
<i>Nesting CSS selectors</i>	124
<i>Importing other style sheets</i>	126
<i>Learning more about preprocessors</i>	127
<i>Using linting support</i>	128

Selecting code	129
<i>Using line numbers</i>	129
<i>Using tag selectors</i>	130
<i>Using parent tag selectors</i>	132
Collapsing code	133
Expanding code	134
Accessing Split Code view	134
Previewing assets in Code view	136
Review questions	138
Review answers	139

5 WEB DESIGN BASICS 140



Developing a new website	142
<i>What is the purpose of the website?</i>	142
<i>Who is the audience?</i>	142
<i>How do they get here?</i>	143
Scenario	144
Working with thumbnails and wireframes	144
<i>Creating thumbnails</i>	146
<i>Creating a page design</i>	146
<i>Creating wireframes</i>	147
<i>Designing for mobile devices</i>	149
<i>The third way</i>	150
Review questions	152
Review answers	153

6 CREATING A PAGE LAYOUT 154



Evaluating page design options	156
Working with predefined layouts	156
Styling a predefined layout	159
Styling elements using the Extract panel	161
Troubleshooting CSS conflicts	167
Extracting text from a Photoshop mockup	170
Deleting components and attributes from a template	173
Inserting new menu items	176
Creating new elements with the DOM panel	179

Creating menu items with copy and paste	181
Extracting text styling	182
Creating a gradient background using Extract	186
Extracting image assets from a mockup	191
Creating new Bootstrap structures	193
Adding a background image to the header	197
Finishing up the layout	202
Review questions	204
Review answers	205

7 WORKING WITH TEMPLATES 206



Creating Dreamweaver templates	208
Removing unneeded components	209
Modifying a Bootstrap layout	216
Modifying text formatting in a Bootstrap element	221
Adding template boilerplate and placeholders	223
Fixing semantic errors	226
Inserting metadata	229
Validating HTML code	230
Working with editable regions	232
<i>Image carousel</i>	232
<i>Card-based section</i>	232
<i>List-based section</i>	232
<i>Inserting a new Bootstrap element</i>	233
<i>Inserting an editable region</i>	236
<i>Inserting an editable optional region</i>	239
Working with child pages	242
<i>Creating a new page</i>	242
<i>Adding content to child pages</i>	244
<i>Adding metadata to a child page</i>	246
Updating a template	247
<i>Removing an optional region from a child page</i>	251
<i>Removing unused sections from a child page</i>	252
Review questions	255
Review answers	255

8	WORKING WITH TEXT, LISTS, AND TABLES	256
	Previewing the completed file.....	258
	Creating and styling text.....	260
	<i>Importing text</i>	260
	<i>Duplicating Bootstrap rows</i>	264
	<i>Deleting unused Bootstrap components</i>	267
	Creating lists.....	268
	Basing content structures on lists.....	275
	<i>Pasting multiple elements in Live view</i>	278
	<i>Creating new list-based items</i>	279
	Creating and styling tables.....	282
	<i>Creating tables from scratch</i>	283
	<i>Copying and pasting tables</i>	288
	<i>Styling tables with CSS</i>	290
	<i>Styling table cells</i>	292
	<i>Adding header rows to tables</i>	294
	<i>Controlling table display</i>	296
	<i>Inserting tables from other sources</i>	298
	<i>Creating semantic text structures</i>	300
	<i>Adding and formatting caption elements</i>	303
	Spell-checking webpages.....	305
	Finding and replacing text.....	307
	Review questions.....	313
	Review answers.....	313
9	WORKING WITH IMAGES	314
	Web image basics.....	316
	<i>Vector graphics</i>	316
	<i>Raster graphics</i>	316
	<i>Raster image file formats</i>	320
	Previewing the completed files.....	321
	Inserting an image.....	323
	Inserting images in Design view.....	326
	Resizing images.....	328
	Inserting Photoshop Images.....	331
	Inserting images using the Assets panel.....	336
	Adapting images to mobile design.....	338

Using the Insert menu	340
Working with the Insert panel	342
Inserting images into the site template	344
Adding CSS classes to template structures	348
Adding images to a Bootstrap carousel	352
Styling headings and text in a Bootstrap carousel	354
Self-paced exercise: Inserting images in child pages	356
Review questions	359
Review answers	359

10 WORKING WITH NAVIGATION 360



Hyperlink basics	362
<i>Internal and external hyperlinks</i>	362
<i>Relative vs. absolute hyperlinks</i>	362
Previewing the completed files	363
Creating internal hyperlinks	369
<i>Creating relative links</i>	369
<i>Creating a home link</i>	372
<i>Updating links in child pages</i>	377
Creating an external link	380
<i>Creating an absolute external link in Live view</i>	380
Setting up email links	385
Creating an image-based link	387
<i>Creating image-based links using the Element Display</i>	387
<i>Creating text links using the Text Display</i>	388
Targeting page elements	390
<i>Creating internal targeted links</i>	391
Creating a destination link in the Element Display	394
<i>Targeting id-based link destinations</i>	395
Locking an element on the screen	396
Styling a navigation menu	398
Adding a telephone link	401
Checking your page	402
Self-paced exercise: Adding additional links	403
Review questions	405
Review answers	405

11 PUBLISHING TO THE WEB	406
 Defining a remote site	408
Setting up a remote FTP site (optional)	409
Cloaking folders and files	414
Wrapping things up.....	416
Creating a home page	417
Completing the home page	419
Putting your site online (optional)	422
Synchronizing local and remote sites	427
Review questions	432
Review answers.....	433

APPENDIX: TINY URLS	434
----------------------------	------------

INDEX	435
--------------	------------

● **Note:** You will find this lesson on your Account page once you register your book, as described in “Accessing the lesson files and Web Edition.”

12 WORKING WITH MOBILE DESIGN	B-2
--------------------------------------	------------

4

WORKING WITH CODE

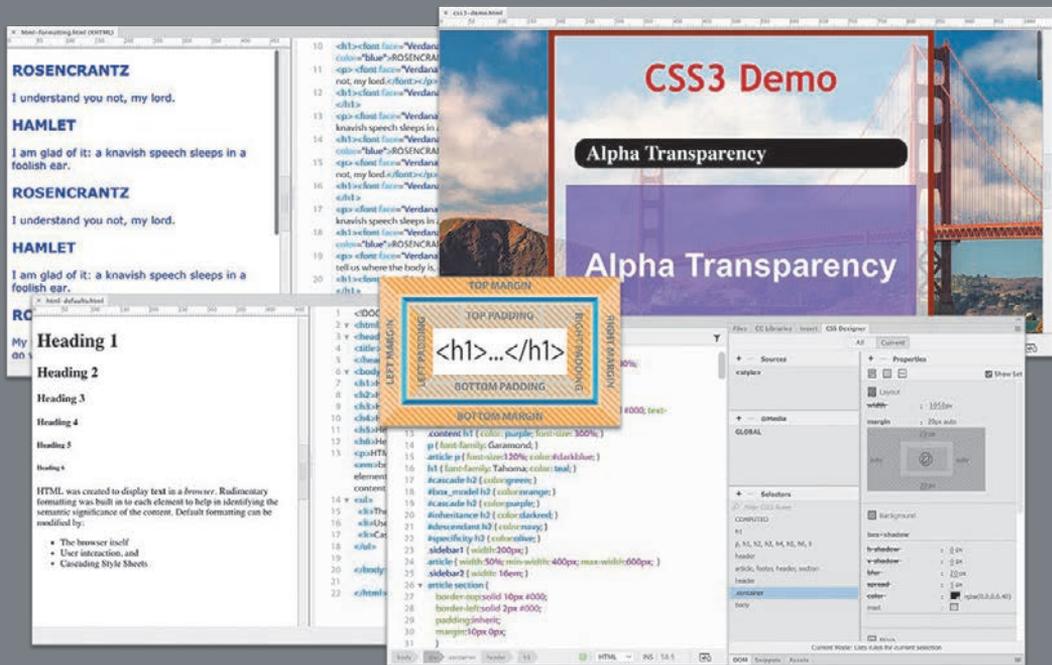
Lesson overview

In this lesson, you'll learn how to work with code and do the following:

- Write code using code hinting and Emmet shorthand.
- Set up a CSS preprocessor and create SCSS styling.
- Use multiple cursors to select and edit code.
- Collapse and expand code entries.
- Use Live Code view to test and troubleshoot dynamic code.
- Use Inspect mode to identify HTML elements and associated styling.
- Access and edit attached files using the Related Files interface.



This lesson will take about 90 minutes to complete. To get the lesson files used in this lesson, download them from the webpage for this book at www.adobepress.com/DreamweaverCIB2021. For more information, see “Accessing the lesson files and Web Edition” in the “Getting Started” section at the beginning of this book.



Dreamweaver's claim to fame is as a visually based HTML editor, but its code-editing features don't take a back seat to its graphical interface, and they offer few compromises to professional coders and developers.

Creating HTML code

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

● **Note:** Some tools and options are available only when Code view is active.

As one of the leading WYSIWYG HTML editors, Dreamweaver allows users to create elaborate webpages and applications without touching or even seeing the code that does all the work behind the scenes. But for many designers, working with the code is not only a desire but a necessity.

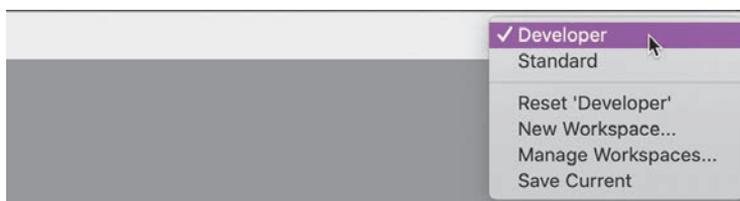
Although Dreamweaver has always made it as easy to work with a page in Code view as it is in Design view or Live view, some developers believe that the code-editing tools took a back seat to the visual design interface. Although in the past this was partially true, Dreamweaver CC (2021 release) is fully invested in the vastly improved tools and workflows for coders and developers that were brought to the program in the previous version. In fact, Dreamweaver can now unify your entire web development team as never before by providing a single platform that can handle almost any task.

You’ll often find that a specific task is actually easier to accomplish in Code view than in Live view or Design view alone. In the following exercises, you’ll learn more about how Dreamweaver makes working with the code an effortless and surprisingly enjoyable task.

Writing code manually

As you complete this and the next eight lessons, you will have numerous opportunities to view and edit code by hand. But for anyone jumping directly to this lesson, this exercise will provide a quick overview of the topic. One way to experience Dreamweaver’s code-writing and editing tools is to create a new file.

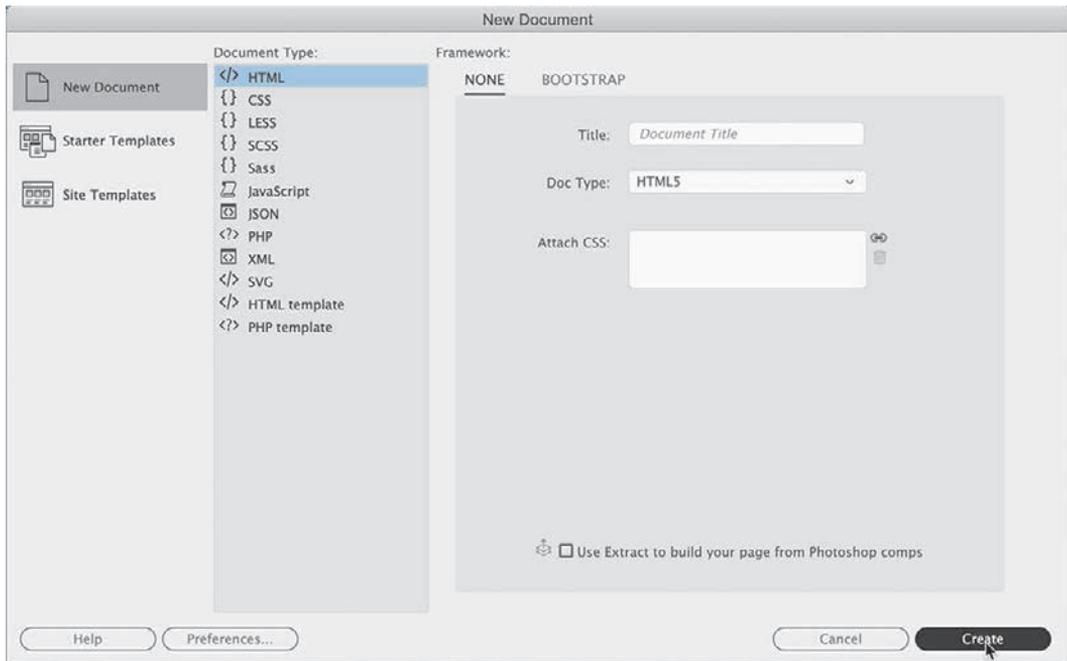
- 1 Define a site based on the lesson04 folder downloaded from your account page, as described in the “Getting Started” section at the beginning of the book.
- 2 Select Developer from the Workspace menu.



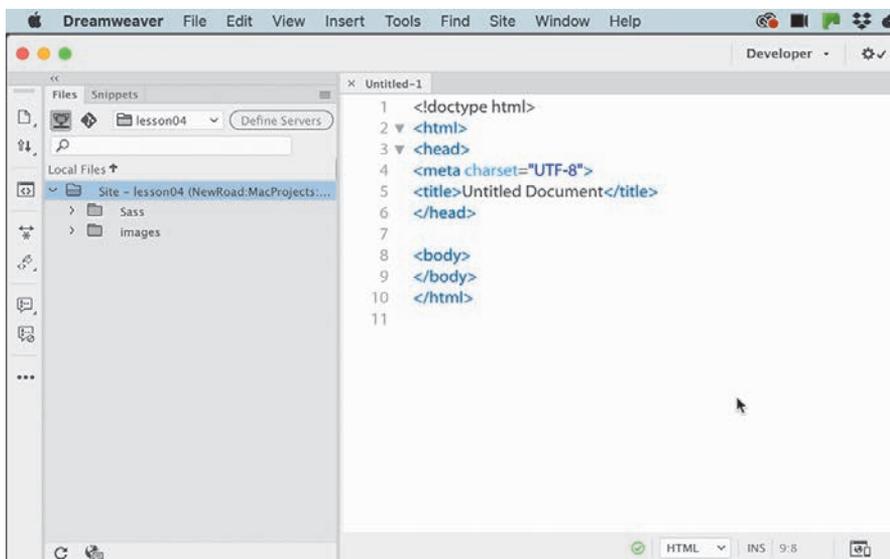
All the code-editing tools work identically in either workspace, but the Developer workspace focuses on the Code view window and provides a better experience for the following exercises.

3 Choose File > New.

The New Document dialog appears.



4 Choose New Document > HTML > None.
Click Create.

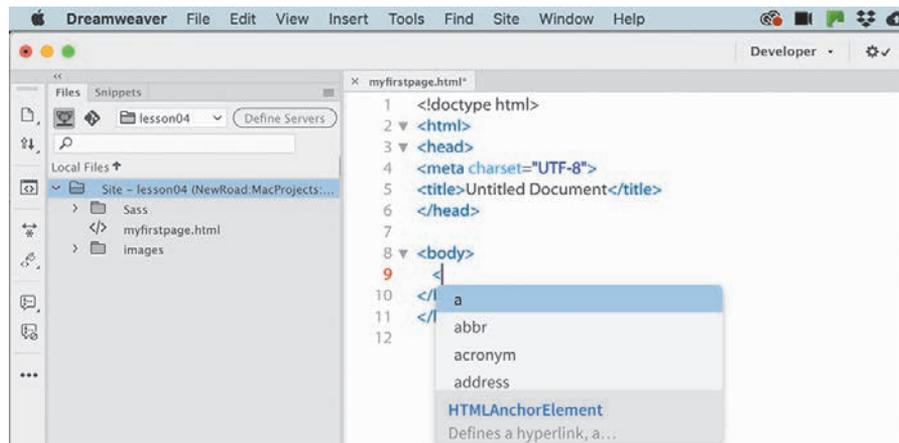


● **Note:** In all screen shots, we use the Classic color theme, which can be selected in Preferences. See the “Getting Started” section at the beginning of the book for more details.

Dreamweaver creates the basic structure of a webpage automatically. The cursor will probably appear at the beginning of the code.

As you can see, Dreamweaver provides color-coded tags and markup to make it easier to read, but that's not all. It also offers code hinting for ten different web development languages, including but not limited to HTML, CSS, JavaScript, and PHP.

- 5 Choose File > Save.
- 6 Name the file **myfirstpage.html** and save it in the lesson04 folder.
- 7 Insert the cursor after the <body> tag.
Press Enter/Return to create a new line. Type <



● **Note:** Line breaks, indenting, and other whitespace is not required in HTML and is used only to make the code easier to read and edit.

A code-hinting window appears, showing you a list of HTML-compatible codes you can select from.

- 8 Type **d**
The code-hinting window filters to code elements that start with the letter *d*. You can continue to type the tag name directly or use this list to select the desired element. By using the list, you can eliminate simple typing errors.
- 9 Press the Down Arrow key.
The **dd** tag in the code-hinting window is highlighted.
- 10 Continue pressing the Down Arrow key until the tag **div** is highlighted.
Press Enter/Return.



The tag name `div` is inserted in the code. The cursor remains at the end of the tag name, waiting for your next input. For example, you could complete the tag name or enter various HTML attributes. Let's add an `id` attribute to the `div` element.

- 11 Press the spacebar to insert a space.

The hinting menu opens again, displaying a different list; this time the list contains various appropriate HTML attributes.

- 12 Type `id` and press Enter/Return.



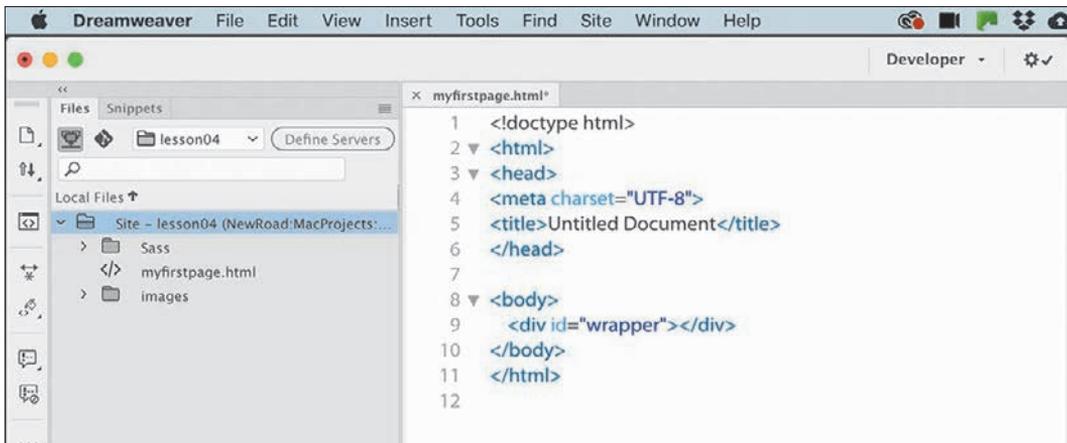
Dreamweaver creates the `id` attribute, complete with equals sign and quotation marks. Note that the cursor appears within the quotation marks, ready for your entry.

- 13 Type `wrapper` and press the Right Arrow key once.

The cursor moves outside the closing quotation mark.

- **Note:** If your tag closed automatically, you can skip step 14.

- 14 Type `></`



When you type the backslash (`/`), Dreamweaver closes the `div` element automatically. As you see, the program can provide a lot of help as you write code manually. But it can help you write code automatically too.

- 15 Choose File > Save.

● **Note:** Depending on the settings in your program, tags may close automatically and you may have to move the cursor to complete the next step. This behavior can be turned off or adjusted in the Code Hints section of Preferences.

● **Note:** In HTML5, quotation marks are no longer required for tag attributes. However, older browsers and applications may not display the code properly without them. It does no harm to use them, so continue to add them to your code.

Writing code automatically

Emmet is a web-developer toolkit that was added to a previous version of Dreamweaver and enables you to supercharge your code-writing tasks. When you enter shorthand characters and operators, Emmet enables you to create whole blocks of code with just a few keystrokes. To experience the power of Emmet, try this exercise.

- 1 If necessary, open **myfirstpage.html**.
- 2 In Code view, insert the cursor within the `div` element and press Enter/Return to create a new line.

Emmet is enabled by default and works whenever you are typing in Code view. In the original site mockup, the navigation menu appears at the top of the page. HTML5 uses the `<nav>` element as the foundation of site navigation.

- 3 Type **nav** and press Tab.

<pre>8 ▾ <body> 9 <div id="wrapper"> 10 nav</div> 11 </body> 12 </html> 13</pre>	<pre>8 ▾ <body> 9 <div id="wrapper"> 10 <nav></nav></div> 11 </body> 12 </html> 13</pre>
--	--

Dreamweaver creates the opening and closing tags all at once. The cursor appears inside the `nav` element, ready for you to add another element, some content, or both.

HTML navigation menus are usually based on an unordered list, which consists of a `` element with one or more child `` elements. Emmet allows you to create multiple elements at the same time, and by using one or more operators, you can specify whether the subsequent elements follow the first (+) or are nested one within the other (>).

- 4 Type **ul>li** and press Tab.

<pre>8 ▾ <body> 9 <div id="wrapper"> 10 <nav>ul>li</nav></div> 11 </body> 12 </html> 13</pre>	<pre>8 ▾ <body> 9 ▾ <div id="wrapper"> 10 ▾ <nav> 11 12 </nav></div> 13 </body> 14 </html></pre>
--	---

A `` element containing one list item appears. The greater-than symbol (>) is used to create the parent-child structure you see here. By adding another operator, you can create several list items.

- 5 Choose Edit > Undo.

The code reverts to the `ul>li` shorthand. It's easy to adapt this shorthand markup to create a menu with five items.

- 6 Edit the existing shorthand phrase as highlighted—`ul>li*5`—and press Tab.

<pre>8 <body> 9 <div id="wrapper"> 10 <nav>ul>li*5</nav></div> 11 </body> 12 </html> 13</pre>	<pre>8 <body> 9 <div id="wrapper"> 10 <nav> 11 12 13 14 15 16 </nav></div> 17 </body> 18 </html> 19</pre>
--	--

A new unordered list appears, this time with five `` elements. The asterisk (*) is the mathematical symbol for multiplication, so this latest change says “`` times 5.”

To create a proper menu, you also need to add a hyperlink to each menu item.

- 7 Press Ctrl+Z/Cmd+Z or choose Edit > Undo.

The code reverts to the `ul>li*5` shorthand.

- 8 Edit the existing shorthand phrase as highlighted:

`ul>li*5>a`

If you guessed that adding the markup `>a` would create a hyperlink child element for each link item, you are correct. Emmet can also create placeholder content. Let’s use it to insert some text in each link item.

- 9 Edit the shorthand phrase as highlighted:

`ul>li*5>a{Link}`

Adding text within braces passes it to the final structure of the hyperlink, but we’re not done yet. You can also increment the items, such as Link 1, Link 2, Link 3, and so on, by adding a variable character (\$).

- 10 Edit the shorthand phrase as highlighted—`ul>li*5>a{Link $}`—and press Tab.

● **Note:** The cursor must be outside the brace before you press Tab.

<pre>8 <body> 9 <div id="wrapper"> 10 <nav>ul>li*5>a{Link \$}</nav></div> 11 </body> 12 </html> 13</pre>	<pre>8 <body> 9 <div id="wrapper"> 10 <nav> 11 Link 1 12 Link 2 13 Link 3 14 Link 4 15 Link 5 16 </nav></div> 17 </body> 18 </html> 19</pre>
--	---

● **Note:** Adding the new line makes the code easier to read and edit, but it has no effect on how it operates.

The new menu appears fully structured, with five link items and hyperlink placeholders incremented 1 through 5. The menu is nearly complete. The only thing missing are targets for the href attributes. You could add them now using another Emmet phrase, but let's save that change for the next exercise.

- 11 Insert the cursor after the closing `</nav>` tag.

Press Enter/Return to create a new line.

Let's see how easy it is to use Emmet to add a header element to your new page.

- 12 Type `header` and press Tab.

As with the `<nav>` element you created earlier, the opening and closing header tags appear, with the cursor positioned to insert the content. We will model the header after one you will use in Lesson 6, "Creating a Page Layout." You need to add two text components: an `<h2>` for the company name and a `<p>` element for the motto. Emmet provides a method for adding not only the tags but also the content.

- 13 Type `h2{Favorite City Tour}+p{Travel with a purpose}` and press Tab.

```
8 <body>
9 <div id="wrapper">
10 <nav><ul>
11 <li><a href="">Link 1</a></li>
12 <li><a href="">Link 2</a></li>
13 <li><a href="">Link 3</a></li>
14 <li><a href="">Link 4</a></li>
15 <li><a href="">Link 5</a></li>
16 </ul></nav>
17 <header>h2{Favorite City Tour}+p{Travel with a
  purpose} </header></div>
18 </body>
19 </html>
```

```
8 <body>
9 <div id="wrapper">
10 <nav><ul>
11 <li><a href="">Link 1</a></li>
12 <li><a href="">Link 2</a></li>
13 <li><a href="">Link 3</a></li>
14 <li><a href="">Link 4</a></li>
15 <li><a href="">Link 5</a></li>
16 </ul></nav>
17 <header><h2>Favorite City Tour</h2>
18 <p>Travel with a purpose</p></header></div>
19 </body>
20 </html>
```

The two elements appear complete and contain the company name and motto. Note how you added the text to each item using braces. The plus (+) sign designates that the `<p>` element should be added as a peer to the heading.

- 14 Insert the cursor after the closing `</header>` tag.

- 15 Press Enter/Return to insert a new line.

As you can see, Emmet enables you to quickly build complex multifaceted parent-child structures like the navigation menu and the header, but it doesn't stop there. As you string together several elements with placeholder text, you can even add `id` and `class` attributes. To insert an `id`, start the name with the hash symbol (#); to add a class, start the name with a dot (.). It's time to push your skills to the next level.

● **Note:** The entire phrase may wrap to more than one line in Code view, but make sure there are no spaces or line breaks within the markup.

- 16 Type `main#content>aside.sidebar1>p(lorem)^article>p(lorem100)^aside.sidebar2>p(lorem)` and press Tab.

```
8 <body>
9 <div id="wrapper">
10 <nav><ul>
11 <li><a href="">Link 1</a></li>
12 <li><a href="">Link 2</a></li>
13 <li><a href="">Link 3</a></li>
14 <li><a href="">Link 4</a></li>
15 <li><a href="">Link 5</a></li>
16 </ul></nav>
17 <header><h2>Favorite City Tour</h2>
18 <p>Travel with a purpose</p></header>
19 main#content>aside.sidebar1>p(lorem)^article>p(lorem100)^aside.sidebar2>p(lorem)</div>
20 </body>
21 </html>
22

18 <p>Travel with a purpose</p></header>
19 <main id="content">
20 <aside class="sidebar1">
21 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Similique dignissimos nostrum voluptates assumenda? Dolor enim ex ipsum dignissimos! Asperiores dolor minus ab placeat fuga neque vero suscipit aspernatur nihil doloribus!</p>
22 </aside>
23 <article>
24 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. In repudiandae iusto nisi quasi, soluta architecto. Ea, quaerat voluptatum. Unde omnis incidunt architecto sunt, pariatur possimus? Ipsam nostrum, assumenda recusandae quia
```

A `<main>` element is created with three child elements (`aside`, `article`, `aside`), along with `id` and `class` attributes. The caret (^) symbol in the shorthand is used to ensure that the `article` and `aside.sidebar2` elements are created as siblings of `aside.sidebar1`. Within each child element, you should see a paragraph of placeholder text.

Emmet includes a *Lorem* generator to create blocks of placeholder text automatically. When you add `lorem` in parentheses after an element name, such as `p(lorem)`, Emmet will generate 30 words of placeholder content. To specify a larger or smaller amount of text, just add a number at the end, such as `p(lorem100)` for 100 words.

Let's finish up the page with a footer element containing a copyright statement.

- 17 Insert the cursor after the closing `</main>` tag.

Create a new line.

Type `footer{Copyright 2021 Favorite City Tour.`

`All rights reserved.}` and press Tab.

```
28 explicabo sit quia rerum optio cum magni nostrum fuga.</p>
29 </aside>
30 </main>footer{Copyright 2021 Favorite City Tour. All rights reserved.} </div>
31 </body>
32 </html>

28 explicabo sit quia rerum optio cum magni nostrum fuga.</p>
29 </aside>
30 </main><footer>Copyright 2021 Favorite City Tour. All rights reserved.</footer> </div>
31 </body>
32 </html>
```

- 18 Save the file.

Using a few shorthand phrases, you have built a complete webpage structure and some placeholder content. You can see how Emmet can supercharge your code-writing tasks. Feel free to use this amazing toolkit at any time to add a single element or a complex, multifaceted component. It's there anytime you need it.

This exercise has barely scratched the surface of what Emmet can do. It is simply too powerful to fully describe in just a few pages. But you got a good peek at its capabilities.

Check out <http://emmet.io> to learn more about Emmet. Check out <http://docs.emmet.io/cheat-sheet/> for a handy Emmet shorthand cheat sheet.

Working with multicursor support

Have you ever wanted to edit more than one line of code at a time? Another code-editing feature in Dreamweaver CC (2021 release) is multicursor support. This feature allows you to select and edit multiple lines of code at once to speed up a variety of mundane tasks. Let's take a look at how it works.

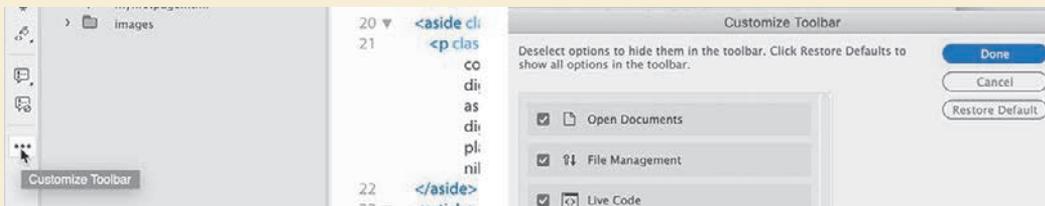
- 1 If necessary, open **myfirstpage.html** as it appears at the end of the previous exercise.

The file contains a complete webpage with header, nav, main, and footer elements. The content features classes and several paragraphs of placeholder text. The `<nav>` element includes five placeholders for a navigation menu, but the href attributes are empty. For the menu and links to appear and behave properly, you need to add a filename, URL, or placeholder element to each link. The hash mark (#) is used as placeholder content until the final link destinations can be added.

Customizing the Common toolbar

Some of the code-editing exercises in this lesson require tools that may not appear in the interface by default. The Common toolbar was previously called the Coding toolbar and appeared only in Code view. The new toolbar appears in all views, but some tools may be visible only when the cursor is inserted directly in the Code view window.

If the exercise calls for a tool that is not visible, even with the cursor in the proper position, you may need to customize the toolbar yourself. This can be done by first clicking the Customize Toolbar icon  and then enabling the tools within the Customize Toolbar dialog. At the same time, feel free to disable tools you don't use.



- 2 Insert the cursor between the quotation marks in the `href=""` attribute in Link 1.

Normally, you would have to add a hash mark (#) to each attribute individually. Multicursor support makes this task much easier, but don't be surprised if it takes you a little practice. Note that all the link attributes are aligned vertically on consecutive lines.

- 3 Hold the Alt key (Windows) or Option key (macOS) and drag the mouse down through all five links.

Using the Alt/Option key enables you to select code or insert cursors in consecutive lines. Be careful to drag down in a straight line. If you slip a little to the left or right, you may select some of the surrounding markup. If that happens, you can just start over. When you are finished, you should see a cursor flashing in the `href` attribute for each link.

- 4 Type #

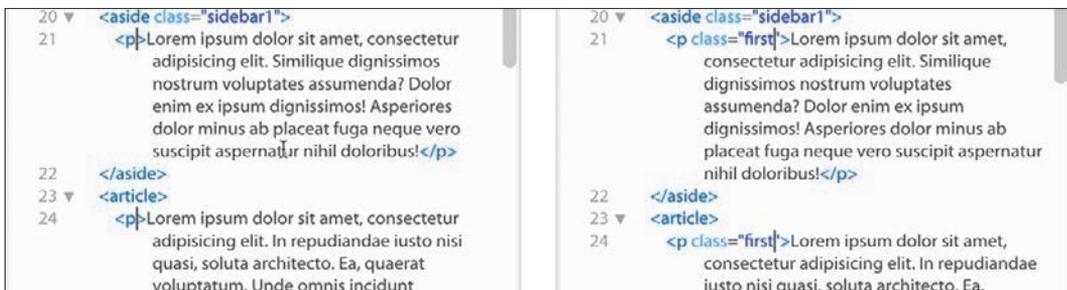
The hash mark (#) appears in all five attributes at the same time.



The screenshot shows two side-by-side code editors. The left editor shows a list of five links with the cursor at the end of the first `href=""` attribute. The right editor shows the same code after the hash mark has been added to all five `href` attributes, resulting in `href="#"` for each link.

The Ctrl/Cmd key enables you to select code or insert cursors in nonconsecutive lines of code.

- 5 Hold the Ctrl/Cmd key and click to insert the cursor between the `p` and the `>` bracket in each of the three opening `<p>` tags in the `<main>` element.
- 6 Press the spacebar to insert a space, and type `class="first"`



The screenshot shows two side-by-side code editors. The left editor shows three `<p>` tags within an `<aside class="sidebar1">` element. The right editor shows the same code after the `class="first"` attribute has been added to all three opening `<p>` tags.

The attribute appears simultaneously in all three `<p>` tags.

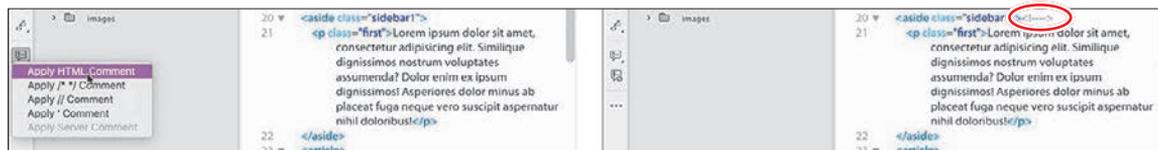
- 7 Save the file.

Multicursor support can save tons of time in repetitive code-editing tasks.

Commenting your code

Comments allow you to leave notes within the code—invisible in the browser—to describe the purpose of certain markup or provide important information to other coders. Although you can add comments manually at any time, Dreamweaver has a built-in feature that can speed up the process.

- 1 Open **myfirstpage.html** and switch to Code view, if necessary.
- 2 Insert the cursor after the opening tag `<aside class="sidebar1">`.
- 3 Click the Apply Comment icon .

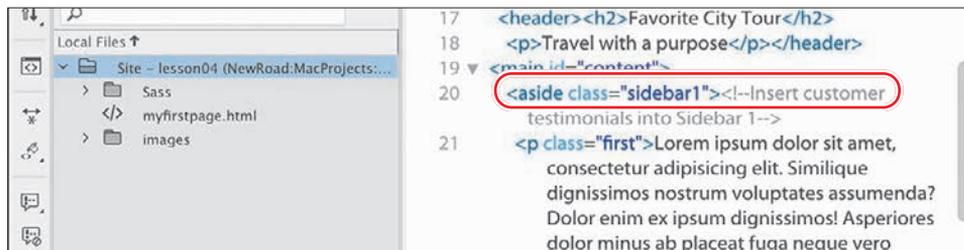


A pop-up menu appears with several comment options. Dreamweaver supports comment markup for various web-compatible languages, including HTML, CSS, JavaScript, and PHP.

- 4 Choose Apply HTML Comment.

An HTML comment block appears, with the text cursor positioned in the center.

- 5 Type **Insert customer testimonials into Sidebar 1**

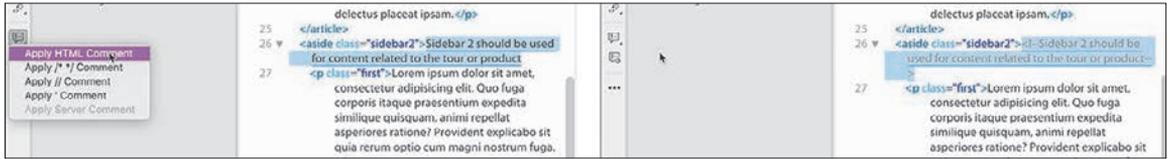


The comment appears in gray between the `<!--` and `-->` markup. The tool can also apply comment markup to existing text.

- 6 Insert the cursor after the opening tag `<aside class="sidebar2">`.
- 7 Type **Sidebar 2 should be used for content related to the tour or product**
- 8 Select the text created in step 7. Click the Apply Comment icon .

A pop-up menu opens.

9 Select Apply HTML Comment.



Dreamweaver applies the `<!--` and `-->` markup to the selection. If you need to remove existing comment markup from a selection, click the Remove Comment icon  in the toolbar.

10 Save all files.

You've created a basic webpage complete with placeholder text. The next step is to style the page. Dreamweaver CC (2021 release) supports CSS preprocessors for LESS, Sass, and SCSS. In the next exercise, you'll learn how to set up and create CSS styling using a preprocessor.

Working with CSS preprocessors

One of the biggest additions to Dreamweaver was support for LESS, Sass, and SCSS. These industry-standard CSS preprocessors are scripting languages that enable you to extend the capabilities of cascading style sheets with a variety of productivity enhancements that can then be compiled in a standard CSS file. These languages provide a variety of benefits for designers and developers who prefer to write their code by hand, including speed, ease of use, reusable snippets, variables, logic, calculations, and much more. No other software is needed to work in these preprocessors, but Dreamweaver also supports other frameworks, such as Compass and Bourbon.

In this exercise, you'll get a taste of how easy it is to use preprocessors with Dreamweaver as well as what advantages they offer compared to a regular CSS workflow.

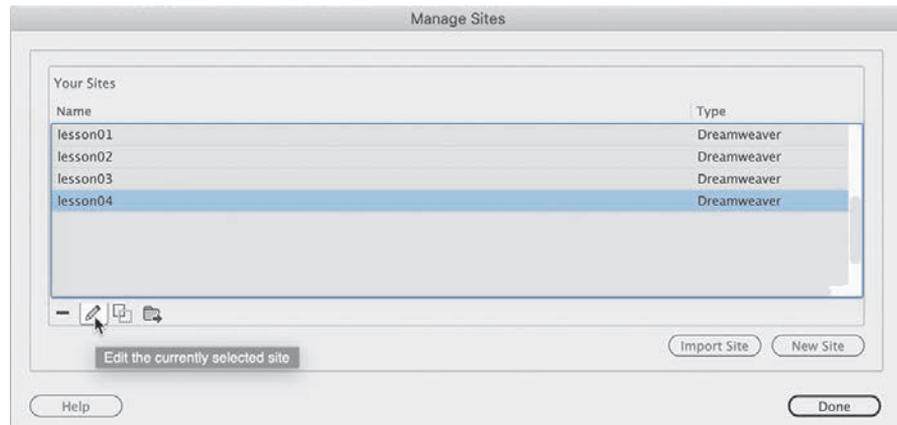
Enabling a preprocessor

Support for CSS preprocessors is site-specific and must be enabled for each site defined in Dreamweaver, as desired. To enable LESS, Sass, or SCSS, you first define a site and then enable the CSS Preprocessors option within the Site Definition dialog.

1 Select Site > Manage Sites.

The Manage Sites dialog appears.

- 2 Select **lesson04** in the Manage Sites window.
Click the Edit icon  at the bottom of the Manage Sites window.



The Site Definition dialog for lesson04 appears.

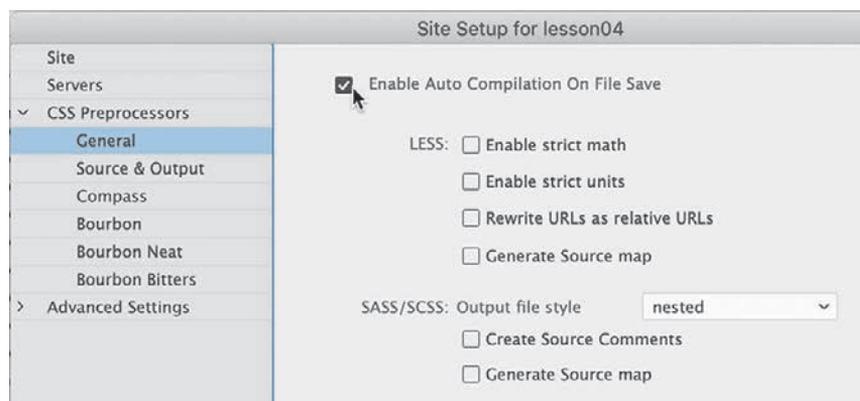
- 3 Select the **CSS Preprocessors** option in the Site Definition dialog.

The CSS Preprocessors option contains six subcategories, including General, Source & Output, and options for various Compass and Bourbon frameworks. You can check out the Dreamweaver Help topics for more information on these frameworks. For this exercise, you need only the features that are built into the program itself.

- 4 Select the General category.

When selected, this category features the on/off switch for the LESS, Sass, or SCSS compiler, as well as various options for how the languages operate. For our purposes, the default settings will work fine.

- 5 Select the Enable Auto Compilation On File Save checkbox to enable the pre-processor compiler, if necessary.



When this is enabled, Dreamweaver will automatically compile your CSS from your LESS, Sass, or SCSS source files whenever they are saved. Some designers and developers use the root folder of the site for compilation. In this case, we'll separate the source and output files in distinct folders.

LESS or Sass—the choice is yours

LESS and Sass offer similar features and functions, so which one should you choose? That's hard to say. Some think that LESS is easier to learn but that Sass offers more powerful functionality. Both make the chore of writing CSS by hand faster and easier and, more importantly, provide significant advantages for maintaining and extending your CSS over time. There are lots of opinions on which preprocessor is better, but you'll find that it comes down to personal preference.

Before you decide, check out the following links to get some informed perspectives:

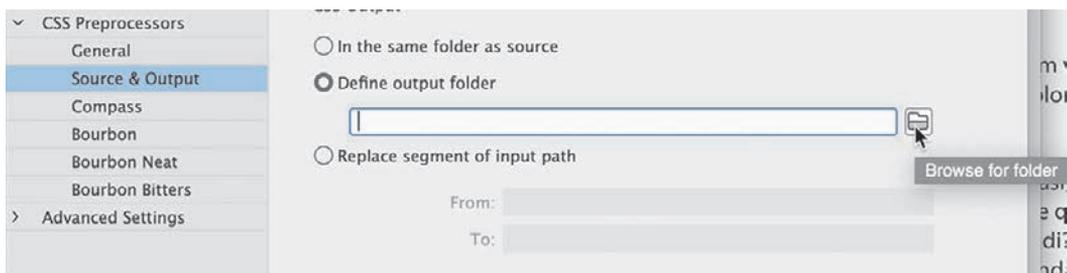
- blog.udemy.com/less-vs-sass/
- css-tricks.com/sass-vs-less/
- keycdn.com/blog/sass-vs-less

Dreamweaver provides two syntaxes for Sass. In this lesson, we use SCSS (Sassy CSS), which is a form of Sass that is written like and looks more like regular CSS.

6 Select the Source & Output category.

This category enables you to designate the source and output folders for your CSS preprocessor. The default option targets the folder where the source file is saved.

7 Select the Define Output Folder option.

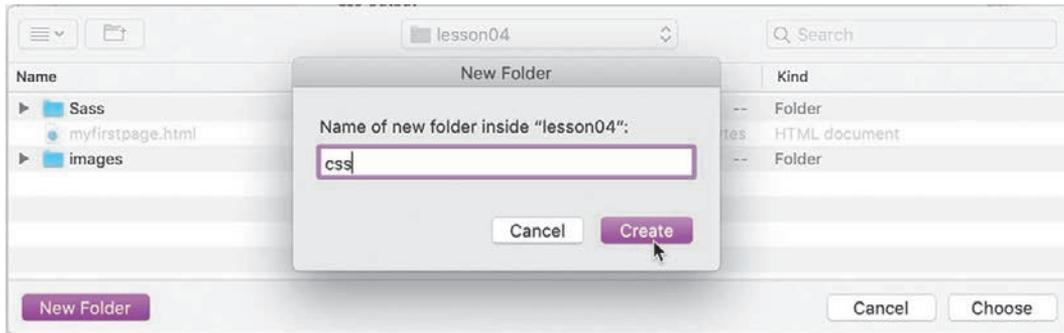


8 Click the Browse For Folder icon .

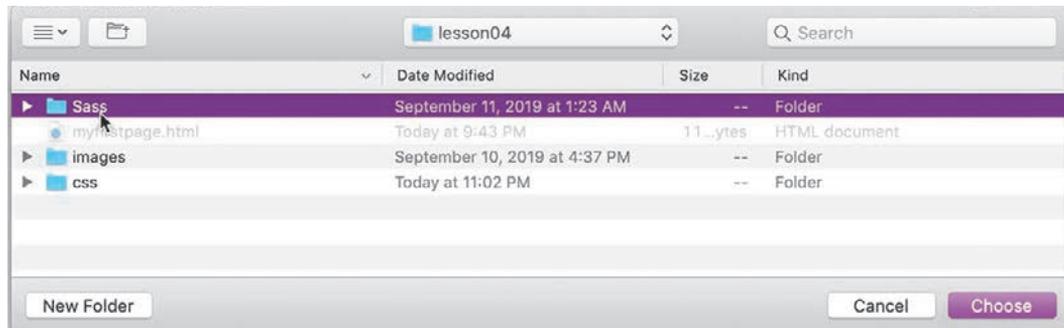
A file browser dialog appears.

9 Navigate to the Site Root folder, if necessary. Create a new folder.

- 10 Name the new folder `css`.
Click Create.



- 11 Select the `css` folder and click Select Folder/Choose.
- 12 Click the Browse For Folder icon  beside the Source Folder field.
- 13 Navigate to the Site Root folder.



Select the existing Sass folder, and click Select Folder/Choose.

- 14 Save the changes and click Done to return to your site.

The CSS preprocessor is enabled, and the source and output folders are now designated. Next, you'll create the CSS source file.

Creating the CSS source file

When using a preprocessor workflow, you do not write the CSS code directly. Instead, you write rules and other code in a source file that is then compiled to the output file. For the following exercise, you'll create a Sass source file and learn some of the functions of that language.

- 1 Select Standard from the Workspace menu.
- 2 Choose Window > Files to display the Files panel, if necessary.
Select `lesson04` from the Site List dropdown menu, if necessary.

- 3 If necessary, open **myfirstpage.html** and switch to Split view.

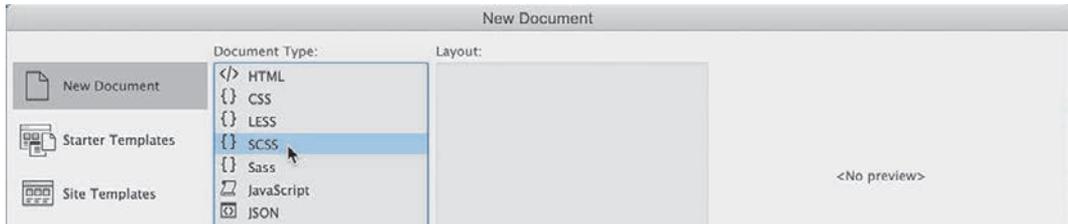
The webpage is unstyled at the moment.

- 4 Choose File > New.

The New Document dialog appears. This dialog allows you to create all types of web-compatible documents. In the Document Type section of the dialog, you will see the LESS, Sass, and SCSS file types.

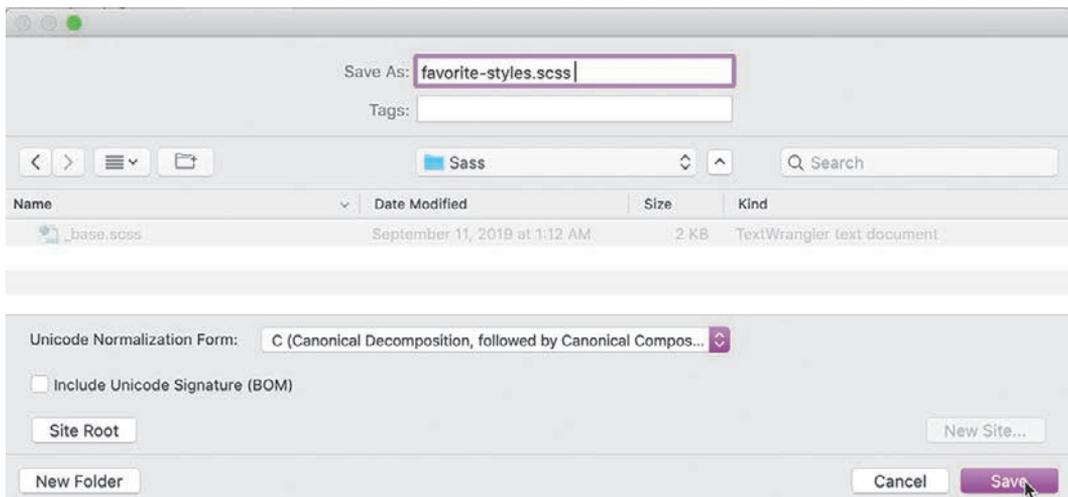
- 5 Choose New Document > SCSS.

Click the Create button.



A new blank SCSS document appears in the document window. SCSS is a flavor of Sass that uses a syntax similar to regular CSS that many users find easier to learn and work with.

- 6 Save the file as **favorite-styles.scss** in the Sass folder you targeted as the Source folder in the previous exercise.



There's no need to create the CSS file; the compiler in Dreamweaver will do that for you. You're all set to start working with Sass. The first step is to define variables. Variables are programmatic constructs that enable you to store CSS specifications you want to use multiple times, such as colors in your site theme.

By using a variable, you have to define it only once. If you need to change it in the future, you can edit one entry in the style sheet and all the instances of the variable will update automatically.

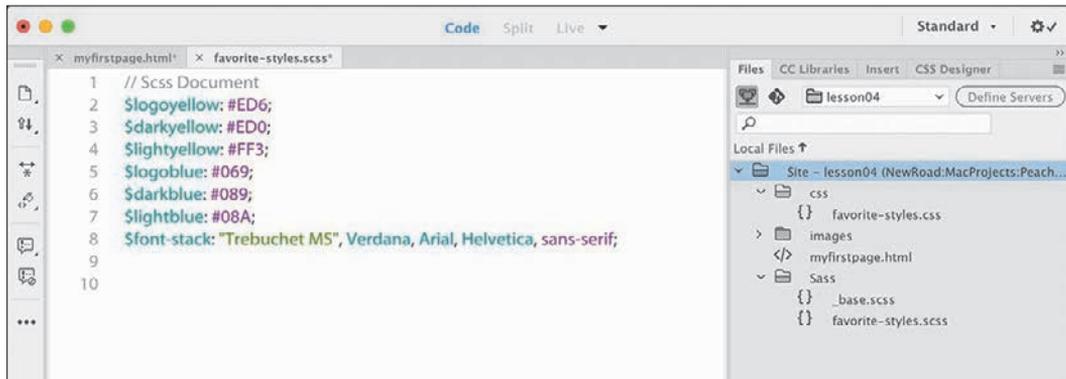
- 7 Insert the cursor into line 2 of **favorite-styles.scss**.

Type **\$logoyellow: #ED6;** and press Enter/Return.

You've created your first variable. This is the main green color of the site theme. Let's create the rest of the variables.

- 8 Type **\$darkyellow: #E00;**
\$lightyellow: #FF3;
\$logoblue: #069;
\$darkblue: #089;
\$lightblue: #08A;
\$font-stack: "Trebuchet MS", Verdana, Arial, Helvetica, sans-serif;

and press Enter/Return to create a new line.



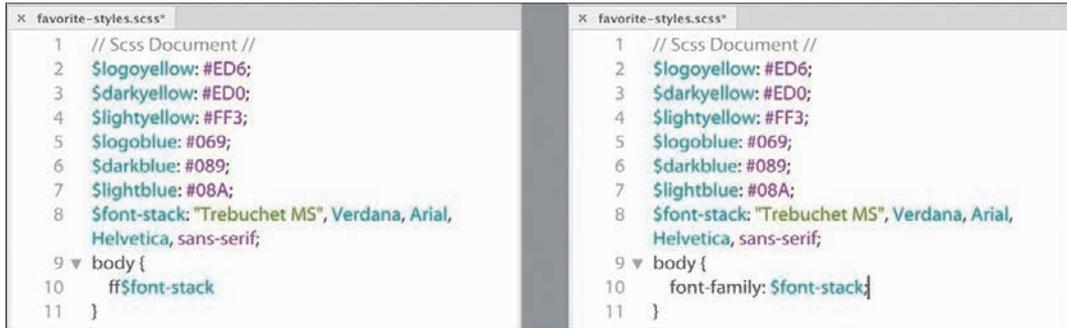
Entering the variables on separate lines makes them easier to read and edit but does not affect how they perform. Just make sure you add a semicolon (;) at the end of each variable.

Let's start the style sheet with the base or default styling of the body element. SCSS markup in most cases looks just like regular CSS, except in this case you'll use one of your variables to set the font family.

- 9 Type **body** and press the spacebar.
Type **{** and press Enter/Return.

When you typed the opening brace ({), Dreamweaver created the closing brace automatically. When you created the new line, the cursor was indented by default, and pressing Enter/Return moved the closing brace to the following line. You can also use Emmet to enter the settings more quickly.

10 Type **ff\$font-stack** and press Tab.



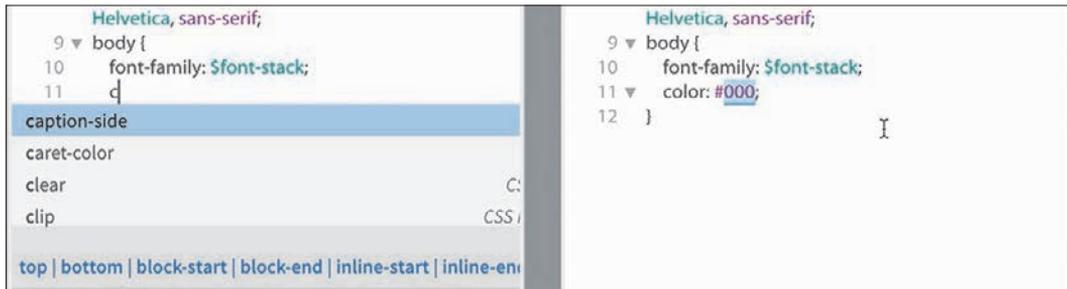
```
1 // Scss Document //
2 $logoyellow: #ED6;
3 $darkyellow: #ED0;
4 $lightyellow: #FF3;
5 $logoblue: #069;
6 $darkblue: #089;
7 $lightblue: #08A;
8 $font-stack: "Trebuchet MS", Verdana, Arial,
  Helvetica, sans-serif;
9 body {
10   ff$font-stack
11 }
```

```
1 // Scss Document //
2 $logoyellow: #ED6;
3 $darkyellow: #ED0;
4 $lightyellow: #FF3;
5 $logoblue: #069;
6 $darkblue: #089;
7 $lightblue: #08A;
8 $font-stack: "Trebuchet MS", Verdana, Arial,
  Helvetica, sans-serif;
9 body {
10   font-family: $font-stack;
11 }
```

The shorthand expands to `font-family: $font-stack;`.

11 Press Enter/Return to create a new line.

Type **c** and press Tab.



```
9 body {
10   font-family: $font-stack;
11   c
caption-side
caret-color
clear
clip
top | bottom | block-start | block-end | inline-start | inline-end
```

```
9 body {
10   font-family: $font-stack;
11   color: #000;
12 }
```

The shorthand expands to `color: #000;`. The default color is acceptable.

12 Hold the Alt/Cmd key and press the Right Arrow key to move the cursor to the end of the current line of code.

13 Press Enter/Return to create a new line.

Type **m0** and press Tab.



```
9 body {
10   font-family: $font-stack;
11   color: #000;
12   m0
13 }
```

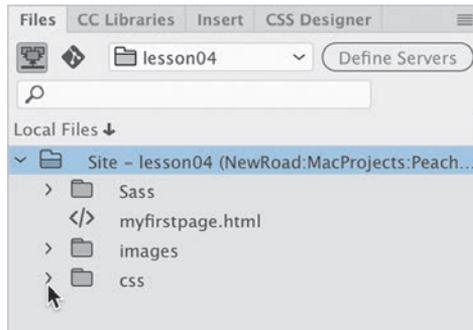
```
9 body {
10   font-family: $font-stack;
11   color: #000;
12   margin: 0;
13 }
```

The shorthand expands to `margin: 0;`, completing the basic styling for the body element. Before you save the file, this is a good time to see how preprocessors do their work.

Compiling CSS code

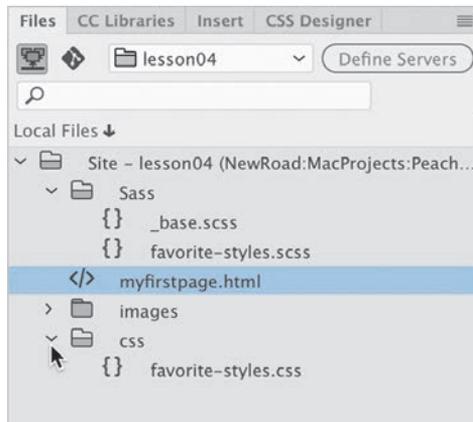
You have completed the specifications for the body element. But you have not created the styling directly in a CSS file. Your entries were made entirely in the SCSS source file. In this exercise, you will see how the compiler that is built into Dreamweaver generates the CSS output.

- 1 Display the Files panel, if necessary, and expand the list of site files.



The site consists of one HTML file and three folders: Sass, images, and css.

- 2 Expand the view of the css and Sass folders.

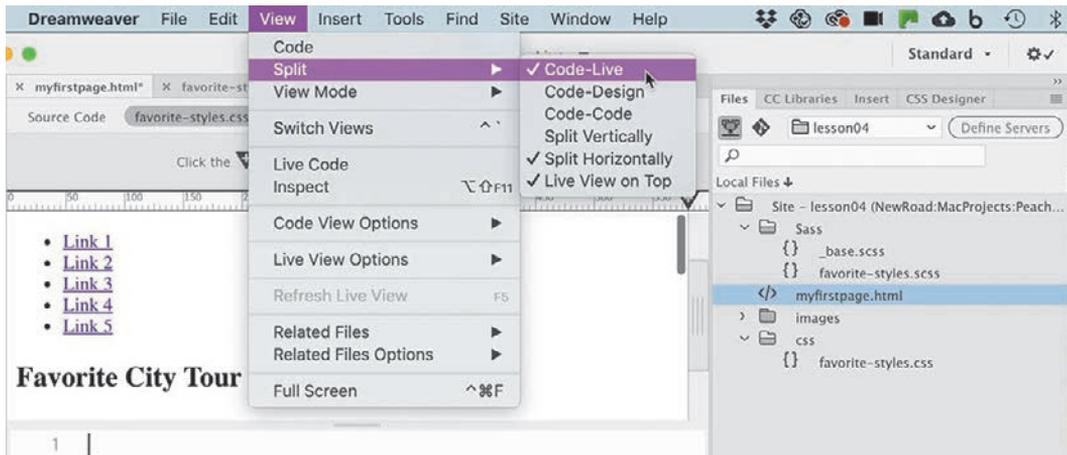


The Sass folder contains **favorite-styles.scss** and **_base.scss**. The css folder contains **favorite-styles.css**. This file did not exist when you started the lesson. It was generated automatically when you created the SCSS file and saved it into the site folder defined as the Source folder. At the moment, the CSS file should contain no CSS rules or markup. It's also not referenced in the sample webpage.

● **Note:** The **favorite-styles.css** file should have been created automatically in the previous exercise when the SCSS file was saved. If you do not see the .css file, you may need to shut down and relaunch Dreamweaver.

3 Select the document tab for **myfirstpage.html**.

Select View > Split > Code-Live.



The page shows only default HTML styling.

4 In the Code view window, insert the cursor after the opening <head> tag and press Enter/Return to insert a new line.

5 Type <link and press the spacebar.

The hinting menu appears. You'll link the webpage to the generated CSS file.

6 Type href and press Enter/Return.



The complete href="" attribute appears, and the hinting menu changes to display the Browse command and a list of pathnames to folders available in the site.

7 Press the Down Arrow key to select the path css/ and press Enter/Return.

The hinting menu now displays the path and filename to **favorite-styles.css**.

- Press the Down Arrow key to select `css/favorite-styles.css` and press Enter/Return.



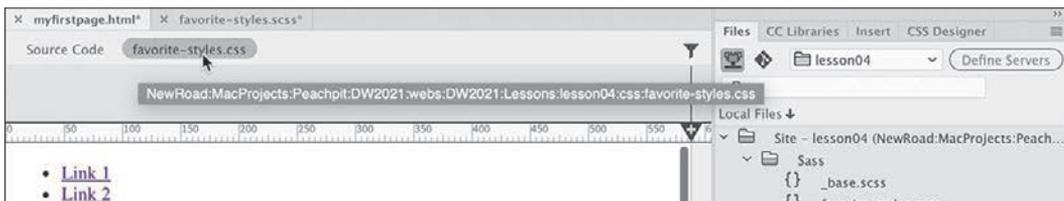
The URL to the CSS output file appears in the attribute. The cursor is moved outside the closing quotation mark and is ready for the next entry. For the style sheet reference to be valid, you need to create one more attribute.

- Press the spacebar, type `rel`, and then press Enter/Return.



Select `stylesheet` from the hinting menu.

- Move the cursor outside the closing quotation mark. Type `>` to close the link.



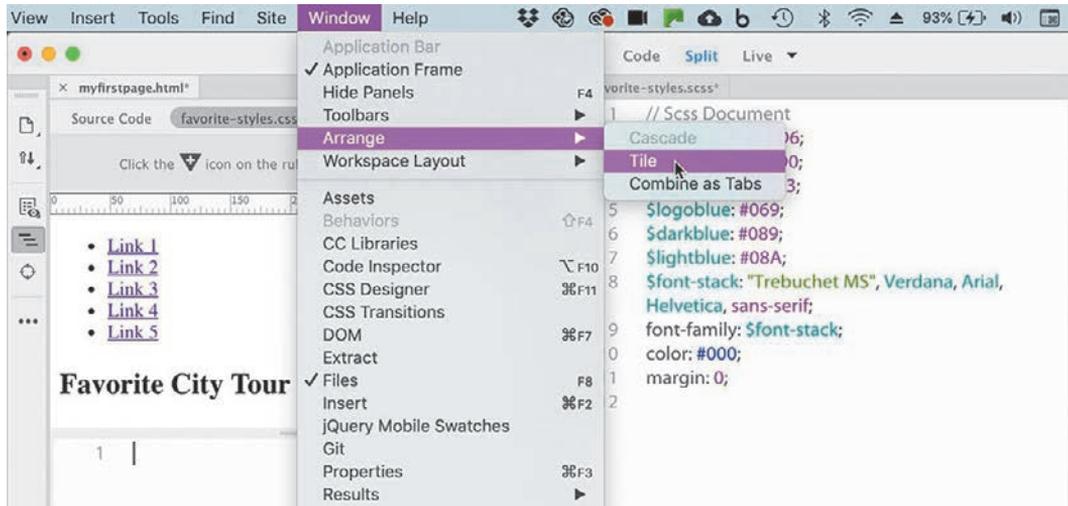
The CSS output file is now referenced by the webpage. In the Live view window, there should be no difference in the styling, but you should now see `favorite-styles.css` displayed in the Related Files interface.

Note: If you accidentally saved the SCSS file before this step, you may see styling in the HTML file and another filename in the Related Files interface.

- Select `favorite-styles.css` in the Related Files interface.

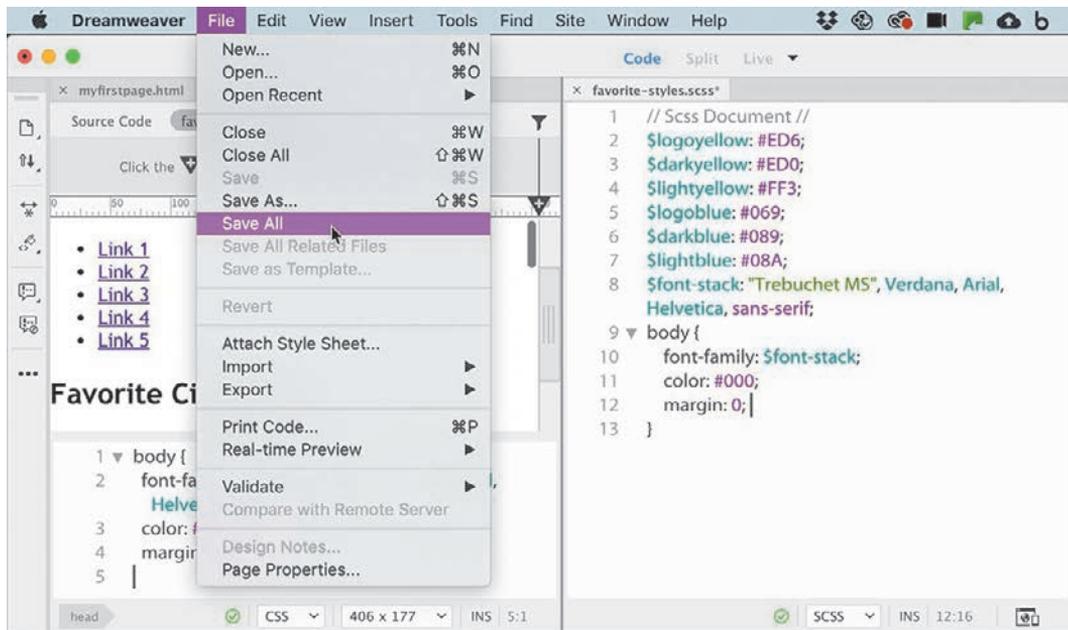
Code view displays the contents of `favorite-styles.css`, which is empty at the moment. An asterisk appears next to the filename in the document tab for `favorite-styles.scss`, indicating that the file has been changed but not saved.

12 Choose Window > Arrange > Tile.



The webpage and the source file appear side by side in the program window.

13 Insert the cursor anywhere in the **favorite-styles.scss** document window and choose File > Save All.



After a moment, the display of **myfirstpage.html** changes, showing the new font and margin settings. The Code view window also updates to display the new contents of **favoritestyles.css**. Each time you save the SCSS source file, Dreamweaver will update the output file.

Nesting CSS selectors

Targeting CSS styling to one element without accidentally affecting another is a constant challenge for web designers everywhere. Descendant selectors are one method for ensuring that the styling is applied correctly. But creating and maintaining the correct descendant structure becomes more difficult as the site and style sheets grow in size. All preprocessor languages offer some form of nesting for selector names.

● **Note:** Make sure you are working in the SCSS file.

In this exercise, you will learn how to nest selectors while styling the navigation menu. First, you'll set the basic styling for the `<nav>` element itself.

1 In the `favoritestyles.scss` window, insert the cursor after the closing brace `}` on line 13 for the body rule.

2 Create a new line; type `nav {` and press Enter/Return.

The `nav` selector and declaration structure are created and ready for your entry. Emmet provides shorthand entries for all CSS properties.

3 Type `bg$logoyellow` and press Tab. Press Enter/Return.

The shorthand expands to `background: $logoblue`, which is the first variable you created in the SCSS source file. This will apply the color `#069` to the `nav` element.

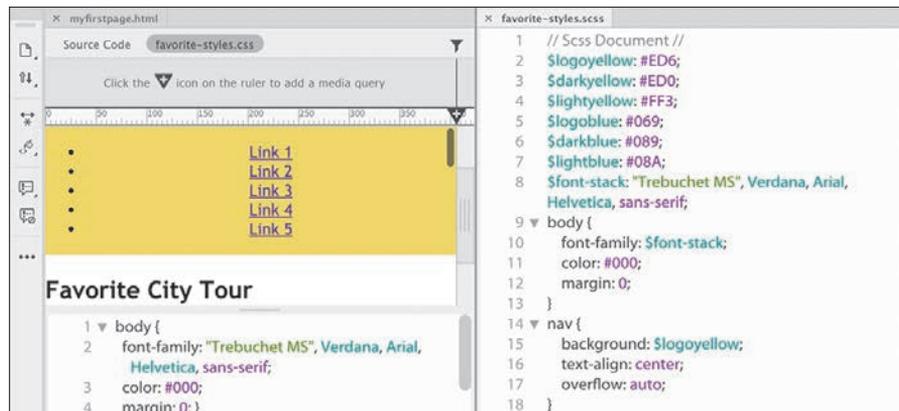
4 Type `ta:c` and press Tab. Press Enter/Return.

The shorthand expands to `text-align: center`.

5 Type `ov:a` and press Tab. Press Enter/Return.

The shorthand expands to `overflow: auto`.

6 Save the source file.



The `<nav>` element in `myfirstpage.html` displays the color `#069`. The menu doesn't look like much yet, but you've only just begun. Next, you'll format the

 element. Note that the cursor is still within the declaration structure for the nav selector.

- 7 Type `ul {` and press Enter/Return.

The new selector and declaration are created within the nav rule.

- 8 Create the following properties:

```
list-style: none; margin: 5px;
```

These properties reset the default styling of the unordered list, removing the bullet and indent. Next, you'll override the styling of the list items.

- 9 Press Enter/Return and type `li {`

Press Enter/Return again.

As before, the new selector and declaration are fully within the ul rule.

- 10 Create the property `display: inline-block;` and press Enter/Return.

This property will display all the links in a single row, side by side. The last element to style is the <a> for the link itself.

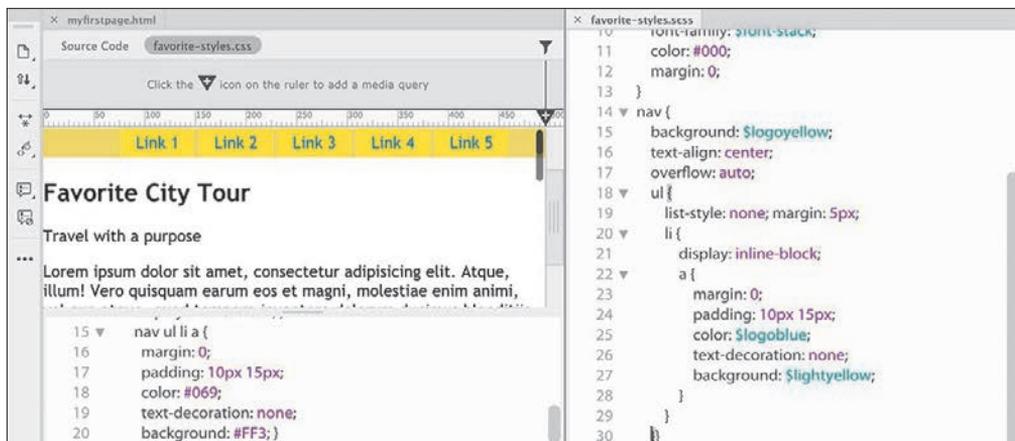
- 11 Type `a {` and press Enter/Return.

Create the following properties:

```
margin: 0;  
padding: 10px 15px;  
color: $logoblue;  
text-decoration: none;  
background: $lightyellow;
```

The rule and declaration for a appear entirely within the li rule. Each of the rules styling the navigation menu has been nested one inside the other in a logical, intuitive manner and will result in an equally logical and intuitive CSS output.

- 12 Save the file.



The navigation menu in **myfirstpage.html** is reformatted to display a single line of links, side by side. The CSS output file displays several new CSS rules. The new rules are not nested as in the source file. They are separate and distinct. And more surprisingly, the selectors have been rewritten to target the descendant structures of the menu, such as `nav ul li a`. As you can see, nesting rules in the SCSS source file eliminates the chore of writing complex selectors.

Importing other style sheets

To make CSS styling more manageable, many designers split their style sheets into multiple separate files, such as one for navigation components, another for feature articles, and still another for dynamic elements. Large companies may create an overall corporate standard style sheet and then allow various departments or subsidiaries to write custom style sheets for their own products and purposes. Eventually, all these CSS files need to be brought together and called by the webpages on the site, but this can create a big problem.

Every resource linked to a page creates an HTTP request that can bog down the loading of your pages and assets. This is not a big deal for small sites or lightly traveled ones. But popular, heavily traveled sites with tons of HTTP requests can overload a web server and even cause pages to freeze in a visitor's browser. Too many experiences like this can cause visitors to flee.

Reducing or eliminating superfluous HTTP calls should be the goal of any designer or developer, but especially those working on large enterprise or highly popular sites. One important technique is to cut down on the number of individual style sheets called by each page. If a page needs to link to more than one CSS file, it's usually recommended that you designate one file as the main style sheet and then simply import the other files into it, creating one large universal style sheet.

In a normal CSS file, importing multiple style sheets would not produce any benefit, because the import command creates the same type of HTTP request that you're trying to avoid in the first place. But since you are using a CSS preprocessor, the import command happens *before* any HTTP request occurs. The various style sheets are imported and combined. Although this makes the resulting style sheet larger, this file is downloaded only once by the visitor's computer and then cached for their entire visit, speeding up the process overall.

Let's see how easy it is to combine multiple style sheets in one file.

- 1 Open **myfirstpage.html** and switch to Split view, if necessary.

Open **favoritestyles.scss** and choose Window > Arrange > Tile.

The two files are displayed side by side to make it easier to edit the CSS and see the changes as they occur.

- 2 In **myfirstpage.html**, click **favoritestyles.css** in the Related Files interface.

Code view displays the content of **favoritestyles.css**. It contains the output of rules written in the SCSS source file.

- 3 In **favoritestyles.scss**, insert the cursor before the body rule.
Type `@import "_base.scss";` and press Enter/Return to insert a new line.
This command imports the contents of the file `_base.scss` stored in the Sass folder. The file was created ahead of time to style other portions of your page.
At the moment, nothing has changed, because **favorite-styles.scss** has not been saved yet.
- 4 Save **favoritestyles.scss** and observe the changes in **myfirstpage.html**.



If you correctly followed the instructions on how to create the HTML structure earlier in this lesson, the page should be entirely formatted now. If you examine **favoritestyles.css**, you will see that several rules were inserted before the body rule. Imported content will be added starting at line 2, the position of the `@import` command. Once the content has been imported, normal CSS precedence and specificity take effect. Just make sure that all rules and file references appear after the variables; otherwise, the variables won't work.

- 5 Save and close all files.

In this section, you created an SCSS file and learned how to work with a CSS pre-processor. You experienced various productivity enhancements and advanced functionality and have glimpsed just a bit of the breadth and scope of what is possible.

Learning more about preprocessors

Check out the following books to learn more about CSS preprocessors and supercharging your CSS workflow:

Beginning CSS Preprocessors: With SASS, Compass.js, and Less.js, by Anirudh Prabhu, Apress (2015), ISBN: 978-1484213483

Instant LESS CSS Preprocessor How-to, by Alex Libby, Packt Publishing (2013), ISBN: 978-1782163763

Jump Start Sass: Get Up to Speed with Sass in a Weekend, by Hugo Giraudel and Miriam Suzanne, SitePoint (2016), ISBN: 978-0994182678

Using linting support

Dreamweaver CC (2021 release) provides live code error checking. Linting support is enabled by default in Preferences, which means the program monitors your code writing and flags errors in real time.

- 1 Open **myfirstpage.html**, if necessary, and switch to Code view. If necessary, select Source Code in the Related Files interface.
- 2 Insert the cursor after the opening `<article>` tag and press Enter/Return to create a new line.
- 3 Type **`<h1>Insert headline here`**
- 4 Save the file.

● **Note:** Dreamweaver may create the opening and closing tags at once. If so, delete the closing `</h1>` tag before proceeding to step 4.

You failed to close the `<h1>` element in step 3. When an error occurs, a red X will appear at the bottom of the document window whenever you save the page.

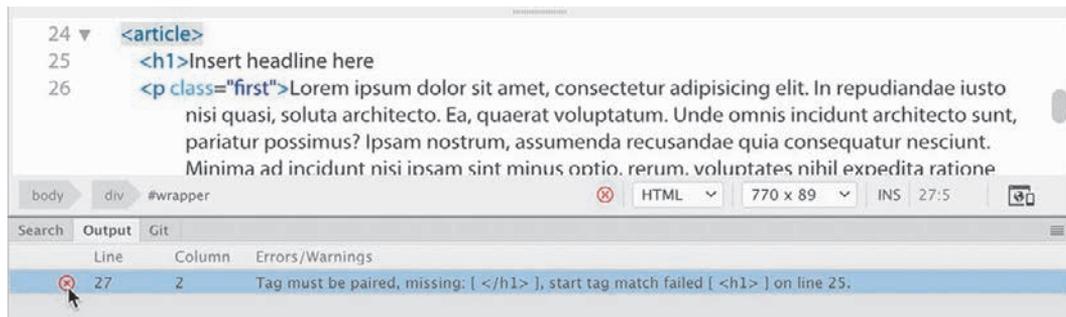
- 5 Click the X  icon.



● **Note:** You may need to click the Refresh button to display the Linting report.

The Output panel opens automatically and displays the coding errors. In this case, the message says that the tag must be paired and identifies what line it thinks the error occurs on. The message erroneously targets line 27, but this can happen because of the nature of HTML tags and structures.

- 6 Double-click the error message.



Dreamweaver focuses on the article in the Code view window that it identifies as containing the error. Since Dreamweaver is looking for the closing tag for the `<h1>` element, the first closing tag it encounters is `</article>` and flags it, which is incorrect. This behavior will get you close to the error, but often you will have to track down the actual issue yourself.

- 7 Insert the cursor at the end of the code
`<h1>Insert headline here. Type </`

● **Note:** If your heading closed automatically in step 3, typing `</` will probably not close the tag. Check your preference settings for code rewriting and adjust them as desired.

Dreamweaver should close the `<h1>` tag automatically. If not, go ahead and finish it properly.

- 8 Save the file.



Once the error is corrected, the red X is replaced by a green checkmark ✓.

- 9 Right-click the Output panel tab and select Close Tab Group from the context menu.

It's important to be alert for this icon as you save your work. No other error message will pop up indicating any problems, and you'll want to catch and correct any errors before uploading your pages to the web server.

Selecting code

Dreamweaver provides several methods for interacting with and selecting code in Code view.

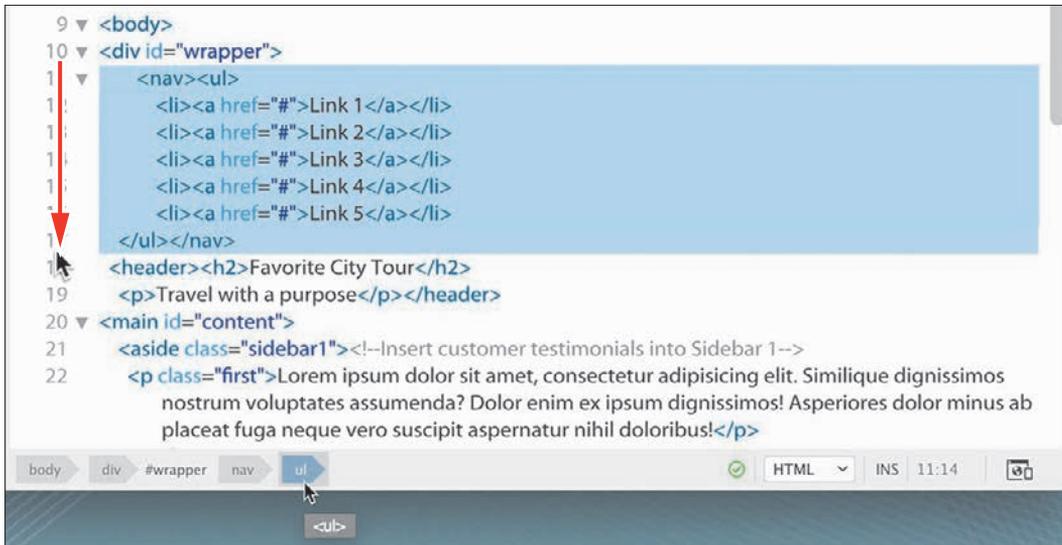
Using line numbers

You can use your cursor to interact with the code in several ways.

- 1 Open **myfirstpage.html**, if necessary, and switch to Code view.
- 2 Scroll down and locate the `<nav>` element (around line 11).
- 3 Drag the cursor across the entire element, including the menu items.

Using the cursor in this way, you can select any portion of the code or its entirety. However, using the cursor in this way can be prone to error, causing you to miss vital portions of the code. At times, using line numbers to select whole lines of code is easier.

- 4 Click the line number beside the `<nav>` tag.
The entire line is selected within the window.
- 5 Drag down the line numbers to select the entire `<nav>` element.



Dreamweaver completely highlights all seven lines. Using line numbers can save a lot of time and avoid errors during selection, but it doesn't take into account the actual structure of the code elements, which may begin and end in the middle of a line. Tag selectors provide a better way to select logical code structures.

Using tag selectors

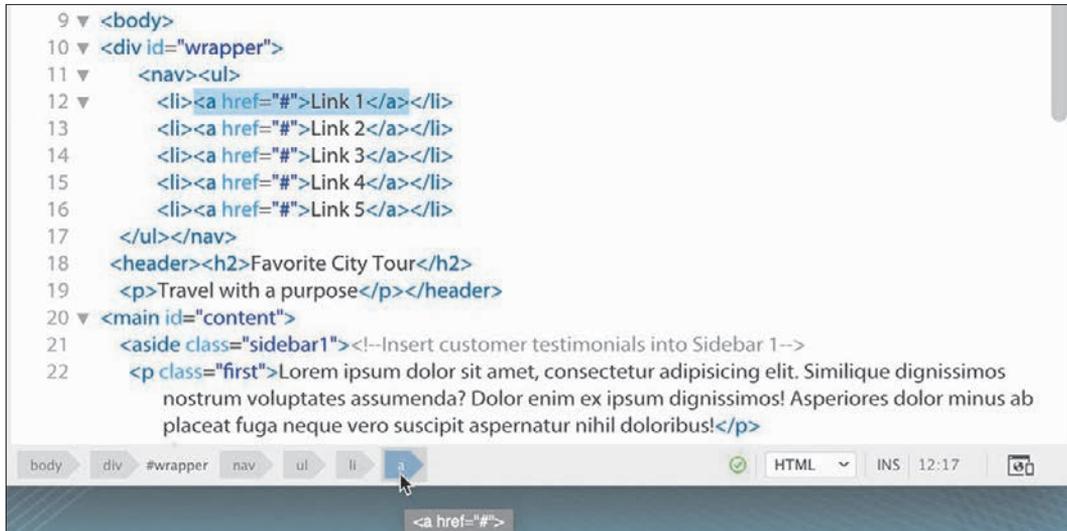
One of the easiest and most efficient ways to select code is to use the tag selectors, as you will frequently do in upcoming lessons.

- 1 Scroll down and locate the following code:
`Link 1`
- 2 Insert the cursor anywhere in the text `Link 1`.
Examine the tag selectors at the bottom of the document window.

The tag selectors in Code view display the `<a>` tag and all its parent elements, the same way they do in Live or Design view.

3 Select the <a> tag selector.

```
9 <body>
10 <div id="wrapper">
11 <nav><ul>
12 <li><a href="#">Link 1</a></li>
13 <li><a href="#">Link 2</a></li>
14 <li><a href="#">Link 3</a></li>
15 <li><a href="#">Link 4</a></li>
16 <li><a href="#">Link 5</a></li>
17 </ul></nav>
18 <header><h2>Favorite City Tour</h2>
19 <p>Travel with a purpose</p></header>
20 <main id="content">
21 <aside class="sidebar1"><!--Insert customer testimonials into Sidebar 1-->
22 <p class="first">Lorem ipsum dolor sit amet, consectetur adipisicing elit. Similique dignissimos
    nostrum voluptates assumenda? Dolor enim ex ipsum dignissimos! Asperiores dolor minus ab
    placeat fuga neque vero suscipit aspernatur nihil doloribus!</p>
```



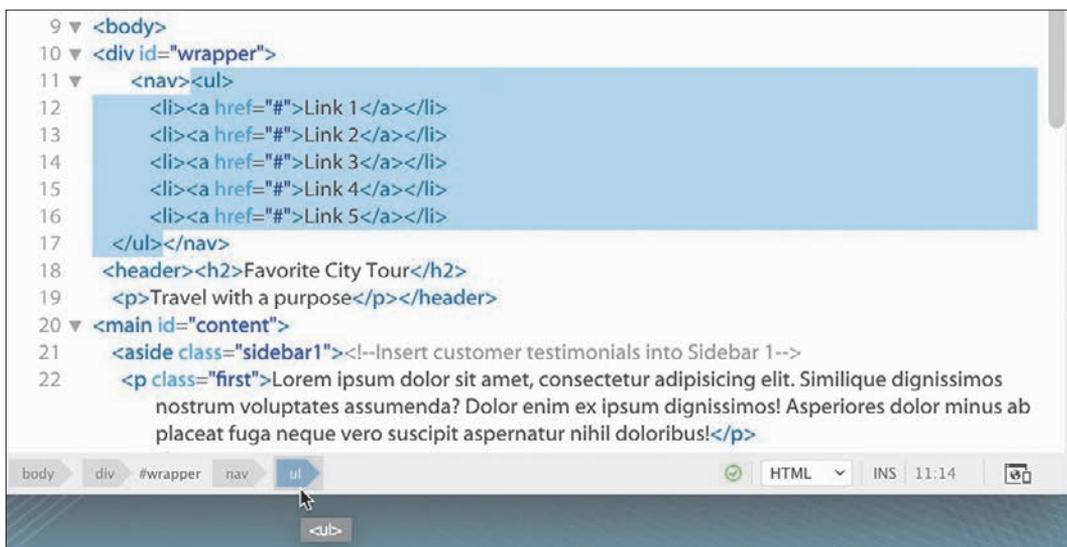
The screenshot shows the Dreamweaver Code view with the breadcrumb trail at the bottom: body > div > #wrapper > nav > ul > li > a. The 'a' tag is highlighted, and a tooltip below it shows ''. The code editor shows the HTML structure with the Link 1 element selected.

The entire <a> element, including its content, is highlighted in Code view. It can now be copied, cut, moved, or collapsed. The tag selectors clearly reveal the structure of the code, even without referring to the Code view display. The <a> is a child of the element, which is a child of , which is in turn a child of <nav>, which is a child of <div#wrapper>, and so on.

The tag selectors make it a simple chore to select any part of the code structure.

4 Select the tag selector.

```
9 <body>
10 <div id="wrapper">
11 <nav><ul>
12 <li><a href="#">Link 1</a></li>
13 <li><a href="#">Link 2</a></li>
14 <li><a href="#">Link 3</a></li>
15 <li><a href="#">Link 4</a></li>
16 <li><a href="#">Link 5</a></li>
17 </ul></nav>
18 <header><h2>Favorite City Tour</h2>
19 <p>Travel with a purpose</p></header>
20 <main id="content">
21 <aside class="sidebar1"><!--Insert customer testimonials into Sidebar 1-->
22 <p class="first">Lorem ipsum dolor sit amet, consectetur adipisicing elit. Similique dignissimos
    nostrum voluptates assumenda? Dolor enim ex ipsum dignissimos! Asperiores dolor minus ab
    placeat fuga neque vero suscipit aspernatur nihil doloribus!</p>
```



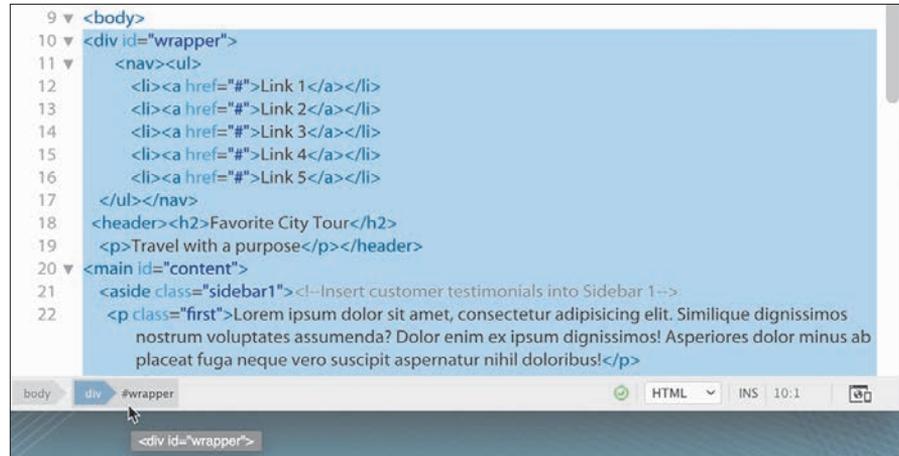
The screenshot shows the Dreamweaver Code view with the breadcrumb trail at the bottom: body > div > #wrapper > nav > ul. The 'ul' tag is highlighted, and a tooltip below it shows ''. The code editor shows the HTML structure with the element selected.

The code for the unordered list is entirely selected.

- 5 Select the `<nav>` tag selector.

The code for the entire menu is selected.

- 6 Select the `<div#wrapper>` tag selector.



```
9 <body>
10 <div id="wrapper">
11 <nav><ul>
12 <li><a href="#">Link 1</a></li>
13 <li><a href="#">Link 2</a></li>
14 <li><a href="#">Link 3</a></li>
15 <li><a href="#">Link 4</a></li>
16 <li><a href="#">Link 5</a></li>
17 </ul></nav>
18 <header><h2>Favorite City Tour</h2>
19 <p>Travel with a purpose</p></header>
20 <main id="content">
21 <aside class="sidebar1"><!--Insert customer testimonials into Sidebar 1-->
22 <p class="first">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Similique dignissimos nostrum voluptates assumenda? Dolor enim ex ipsum dignissimos! Asperiores dolor minus ab placeat fuga neque vero suscipit aspernatur nihil doloribus!</p>
```

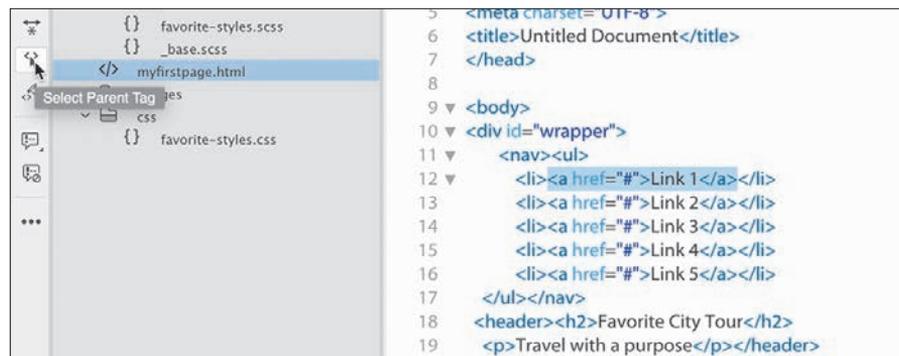
The code for the entire page is now selected. Using the tag selectors allows you to identify and select the structure of any element on your page, but it requires you to identify and select the parent tag yourself. Dreamweaver offers another tool that can do it for you automatically.

Using parent tag selectors

Using the parent tag selector in the Code view window makes the job of selecting the hierarchical structure of your page even simpler.

- 1 Choose Window > Toolbars > Common to display the Common toolbar, if necessary.
- 2 Insert the cursor anywhere in the text Link 1.
- 3 In the Common toolbar, click the Select Parent Tag icon .

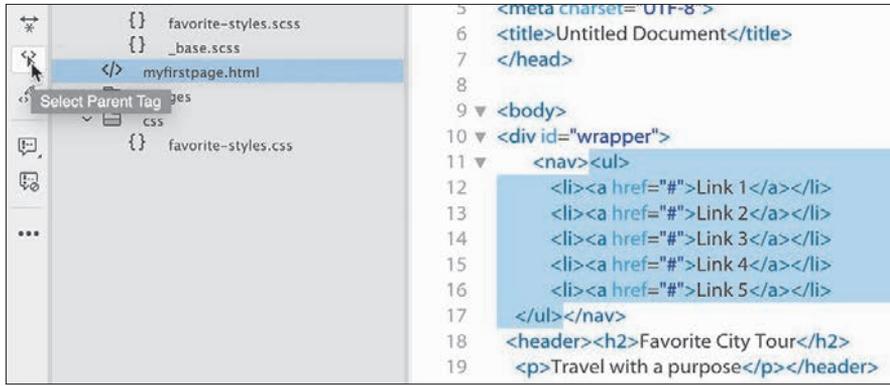
Note: The Select Parent Tag icon may not be displayed by default in the Common toolbar. Click the Customize Toolbar icon  and enable the tool before proceeding to step 3, if necessary.



```
5 <meta charset="UTF-8">
6 <title>Untitled Document</title>
7 </head>
8
9 <body>
10 <div id="wrapper">
11 <nav><ul>
12 <li><a href="#">Link 1</a></li>
13 <li><a href="#">Link 2</a></li>
14 <li><a href="#">Link 3</a></li>
15 <li><a href="#">Link 4</a></li>
16 <li><a href="#">Link 5</a></li>
17 </ul></nav>
18 <header><h2>Favorite City Tour</h2>
19 <p>Travel with a purpose</p></header>
```

The entire `<a>` element is highlighted.

- Click the Select Parent Tag icon  again or press Ctrl+[/ Cmd+[(left bracket).
The entire `` element is selected.
- Click the Select Parent Tag icon .



The entire `` element is selected.

- Press Ctrl+[/ Cmd+[until `<div#wrapper>` is selected.

Each time you click the icon or press the shortcut key, Dreamweaver selects the parent element of the current selection. Once you've selected it, you may find working with long sections of code unwieldy. Code view offers other handy options to collapse long sections to make them easier to work with.

Collapsing code

Collapsing code is a productivity tool that makes a simple process out of copying or moving large sections of code. Coders and developers also collapse code sections when they are looking for a particular element or section of a page and want to temporarily hide unneeded sections from view. Code can be collapsed either by selection or by logical element.

- Select the first three Link items in the `<nav>` element.

Note the Collapse icon  along the left edge of Code view; it indicates that the selection is currently expanded.

- Click the Collapse icon  to collapse the selection.



The selection collapses, showing only the first `` element and a snippet of text from it.

You can also collapse code based on logical elements, like `` or `<nav>`. Notice that each line that contains an opening element tag also displays a Collapse icon.

- 3 Click the Collapse icon ▼ beside the line for the `<nav>` element.

The entire `<nav>` element collapses in the Code window, showing only an abbreviated snippet of the entire element. In either instance, the code hasn't been deleted or damaged in any way. It still functions and operates as expected. Also, the collapse functionality appears only in Code view in Dreamweaver; on the web or in another application, the code will appear normally. To expand the code, just reverse the process, as described in the following section.

Expanding code

When the code is collapsed, you can still copy, cut, or move it like you would any other selected element. You can then expand elements one at a time or all at once.

- 1 Click the Expand icon ► beside the line for the `<nav>` element.



The image shows two side-by-side screenshots of a code editor. The left screenshot shows a code window with a collapsed `<nav>` element. Line 11 contains `<nav>...`. Line 12 contains `Link 1...`. Line 13 contains `Link 4`. Line 14 contains `Link 5`. Line 15 contains `</nav>`. The right screenshot shows the same code window after the `<nav>` element has been expanded. Line 11 now contains `<nav>`. Line 12 contains `Link 1...`. Line 13 contains `Link 4`. Line 14 contains `Link 5`. Line 15 contains `</nav>`. The `` elements are now expanded and visible.

The `<nav>` element expands, but the three `` elements collapsed in the previous exercise are still collapsed.

- 2 Click the Expand icon ► beside the line for the `` elements.

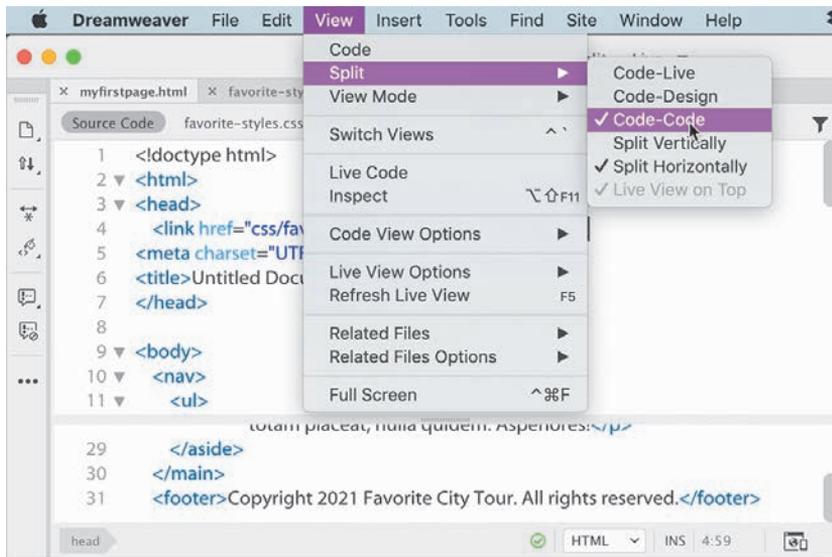
All collapsed elements are now expanded. Note that the Expand icon for the three `` elements disappears altogether.

Accessing Split Code view

Why should coders be denied the ability to work in two windows at the same time? Split Code view enables you to work in two different documents or two different sections of the same document at once. Take your pick.

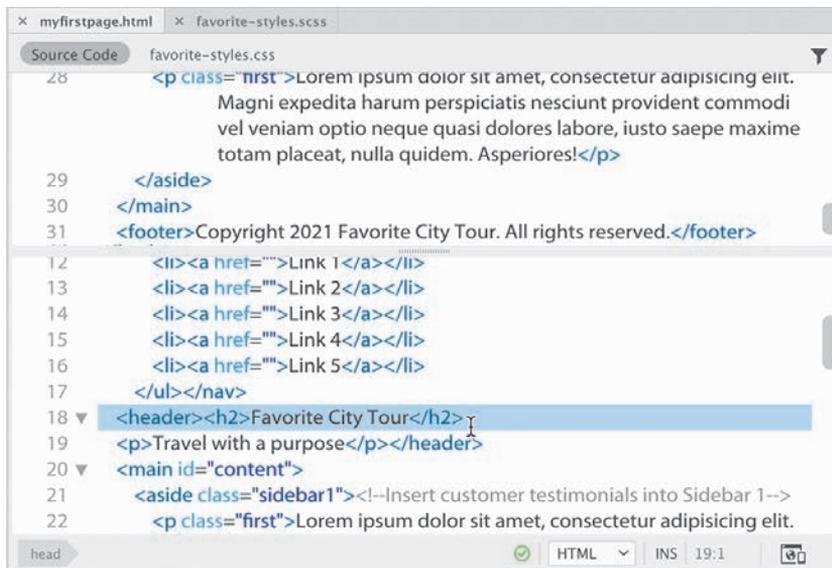
- 1 If necessary, switch to Code view.

- 2 Choose View > Split > Code-Code.



The document displays two Code view windows, both focusing on **myfirstpage.html**.

- 3 Insert the cursor in the top window and scroll down to the <footer> element.
Split Code view enables you to view and edit two different sections of the same file.
- 4 Insert the cursor in the bottom window and scroll to the <header> element.



You can also view and edit the contents of any related file.

- 5 In the Related Files interface, select **favoritestyles.css**.

The window loads the style sheet into one of the windows. You can work in either window and save your changes in real time. Dreamweaver displays an asterisk (*) on any filename in the interface that has been changed but not saved. If you select File > Save or press Ctrl+S/Cmd+S, Dreamweaver saves the changes in the document where your cursor is inserted. Since Dreamweaver can make changes to documents even when they are not open, this feature allows you to edit and update even the files that are closed but linked to your webpage.

Previewing assets in Code view

Although you may be a diehard coder or developer, there's no reason you can't feel the love from Dreamweaver's graphical display too. The program provides visual previews of graphic assets and certain CSS properties in Code view.

- 1 Open **myfirstpage.html**. Select Code view.

In Code view, you see only the HTML. The graphical assets are simply references that appear in the CSS file **favoritestyles.css**.

- 2 Click **favoritestyles.css** in the Related Files interface.

The style sheet appears in the window. Although it's fully editable, don't waste your time making any changes to it. Since the file is the output of the SCSS source file, any changes you make will be overwritten the next time the file compiles.

- 3 Locate the header rule (around line 6).

The header consists of two text elements and two images. You should be able to see the image references in the background property.

- 4 Position the cursor over the markup `url(../images/favcity-logo.jpg)` in the background property (line 8).



A miniature preview of the company logo appears below the cursor.

- 5 Position the cursor over the markup `background-color: #ED6;` in the `background` property.



```
5 ▼ header {
6   text-align: center;
7   padding: 30px 10px;
8   background: url(".../favcity-logo.jpg");
9   background-repeat: repeat;
10  background-size: auto 100%;
11  background-color: #ED6;
12  border-bottom: solid 1px #ED6;
13  background-position: center; }
14 ▼ header h2 {
```

A small color chip appears, displaying the color specified. The preview functions the same way for all color models. You no longer have to guess what image or color you specified before you can see it in Live view or the browser.

In this lesson, you learned a number of techniques to make working with code easier and more efficient. You learned how to write code manually using hinting and auto-code completion and how to write code automatically using Emmet shorthand. You learned how to check code construction using built-in linting support. You learned how to select, collapse, and expand code, as well as how to create HTML comments and view code in different ways.

Overall, you learned that whether you are a visual designer or a hands-on coder, you can rely on Dreamweaver to offer vital features and power that will allow you to create and edit HTML and CSS code without compromises. Remember these techniques as you work through the book and use any of them whenever appropriate.

Review questions

- 1 In what ways does Dreamweaver assist you in creating new code?
- 2 What is Emmet, and what functionality does it provide to users?
- 3 What do you have to install to create a LESS, Sass, or SCSS workflow in Dreamweaver?
- 4 What feature in Dreamweaver reports code errors when you save a file?
- 5 True or false: Collapsed code will not appear in Live view or the browser until it is expanded.
- 6 What Dreamweaver feature provides instant access to files linked or referenced within the document?

Review answers

- 1** Dreamweaver provides code hinting and auto-completion for HTML tags, attributes, and CSS styling as you type, along with support for ColdFusion, JavaScript, and PHP, among other languages.
- 2** Emmet is a scripting toolkit that creates HTML code by converting shorthand entries into complete elements, placeholders, and even content.
- 3** No additional software or services are needed to use LESS, Sass, or SCSS. Dreamweaver supports these CSS preprocessors out of the box. You merely have to enable the compiler in the Site Definition dialog.
- 4** Linting checks the HTML code and structure every time you save a file, and then displays a red X icon at the bottom of the document window when an error is detected.
- 5** False. Collapsing code has no effect on the display or operation of the code outside of Dreamweaver.
- 6** The Related Files interface appears at the top of the document window and enables users to instantly access and review CSS, JavaScript, and other compatible file types linked to the webpage. In some cases, a file displayed in the interface will be stored on a remote resource on the internet. While the Related Files interface enables you to view the contents of all the files displayed, you will be able to edit only ones stored on your local hard drive.

INDEX

NUMBERS

- 1-bit color (monochrome), 318
- 4-bit color space, 318
- 8-bit color space (web-safe color space), 319
- 16-bit color space (high color), 319
- 24-bit color space (true color), 319
- 32-bit color space (alpha transparency), 319, 321

A

- Absolute hyperlinks, 362–363, 380–384
- Add Element icon, DOM panel, 179–181, 233, 235–237
- Add Link icon, new menu items, 178
- Add New Server icon, remote FTP site, 410
- Address element, templates, 219–224
- Adobe Authorized Training Centers, 15
- Adobe Generator, 151
- Adobe Illustrator, 148
- Advanced tab
 - Find and Replace dialog, 310–311
 - New Optional Region dialog, 239–240
 - remote FTP site setup, 412–413
- Alerts, FTP connection, 411
- All mode. *See* CSS Designer, All button
- Alpha transparency (32-bit color space), 319, 321
- Alternative FTP move method, FTP conflicts, 412
- Analytics, questions to ask in web design, 143–144
- App Theme window, 9, 24
- Apply Comment icon, code, 112
- Apply HTML Comment icon, code, 112–113
- Assets panel
 - creating lists, 268
 - importing text, 261
 - inserting images, 336–338
- Assets, previewing in Code view, 136–137
- <a> tag, 130–131, 373–374
- Attributes
 - creating link targets using element, 391–393
 - deleting from page layout template, 173–176
- Audience, website design and, 142
- Automatically Upload Files To Server On Save, remote FTP site setup, 413

B

- Background
 - adding image to header, 197–201
 - creating gradient, 186–191, 400–401
- Background File Activity dialog, 425–426, 431
- Basic tab, Find and Replace, 309
- Berners-Lee, Tim, 58–59
- Bit depth, image color, 318
- _blank option, Target menu, 384
- Boilerplate, 209, 223–226
- Bolded text, conventions, 2
- Bonus material, for this book, 8
- Bootstrap
 - accessing templates, 158
 - adding images to carousel, 352–354
 - creating new list-based items, 279–282
 - creating new structures, 193–197
 - deleting unused components, 267–268
 - duplicating rows, 264–267
 - extracting text from Photoshop mockup, 170–173
 - extracting text styling, 182–186
 - inserting new element, 233–236
 - modifying layout, 216–221
 - modifying text formatting in element, 221–223
 - style sheet is read-only, not editable, 165
 - styling heading and text in carousel, 354–356
 - troubleshooting CSS conflicts, 167–170
- Borders
 - new list-based items, 281–282
 - styling table cells, 293
 - table display, 296
- Box model, CSS, 75–77
- Browse For File icon, creating hyperlinks, 370
- Browser
 - creating absolute external link, 381–383
 - hierarchy for CSS rules, 82
 - HTML5 and, 74–75
 - previewing webpage links, 378–379

C

Cache

- defining Dreamweaver site, 14
- remote FTP site setup, 413

Captions, table, 303–305

Card-based content, 28

Card-based section

- adding boilerplate/placeholders, 225–226
- as content model for template in this book, 209
- creating and styling text, 263–268
- creating lists, 273–274, 276
- deleting unused sections from child page, 252–253
- editable regions, 232
- removing components, 209–212

Cascade theory, CSS styles, 78, 82

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

Case sensitivity, usernames/passwords, 411

Cells, styling table, 292–293

Character entities, HTML, 62–63

Check page, navigation, 402–403

Check Spelling dialog, webpages, 305–306

Child pages

- adding content, 244–245
- adding CSS classes to template, 349–351
- adding metadata, 246–247
- adding unique content. *See* Editable regions, templates
- creating from template, 208, 242
- creating new, 242–243
- inserting images, 356–358
- removing optional region from, 251–252
- removing unused sections from, 252–253
- updating links, 377–379

Class attribute

- conventions for element references, 3
- creating, 94–95
- writing code automatically, 108

Classes

- adding CSS, to template, 348–351
- controlling table display, 296–298

Cloaked folders and files

- publishing to web and, 414–416
- synchronization and, 429
- will not be uploaded, 426

Code

- accessing Split Code view, 134–135
- collapsing, 133–134
- commenting, 112–113

creating HTML, 102–103

CSS preprocessors and. *See* CSS preprocessors

expanding, 134

lesson overview, 100–101

linting support, 128–129

multicursor support, 110–111

previewing assets in Code view, 136–137

review Q&A, 138–139

selecting, 129–133

validating HTML, 230–231

writing automatically, 106–110

writing manually, 103–105

Code font, 2

Code hinting

adding images to Bootstrap carousel, 353

inserting images in Split view, 346

locking element onscreen, 397

writing code manually, 104

Code Navigator, 82–87

Code Theme window, 24–25

Code view

adding images to Bootstrap carousel in, 352–354

customizing workspace in, 25

Find And Replace text in, 307–309

inserting images in, 345

interface preference settings in, 23

previewing assets in, 136–137

Split view and, 27

swapping with Live view, 70

templates work properly in, 244

writing code automatically in, 106–110

writing code manually in, 102–105

Coding toolbar. *See* Common toolbar

Collapsing code, 133–134, 427–428

Color

nesting CSS selectors, 124–126

raster graphics and, 318–319

theme, 9, 24

Columns

Bootstrap layout modification, 216–221

content structures based on lists, 275–278

CSS Designer displaying, 44, 88–89

CSS multicolumn layouts, 157

HTML formatting and, 70

inserting tables from other sources, 299

new Bootstrap structures, 193–194

new list-based items, 279–280

removing unneeded, 213

responsive web design for, 145

table display, 296–298

- Comments, adding/removing, 112–113
- Common toolbar, 38, 110
- Company logo
 - applying as background image, 197–201
 - creating home link from, 372–377
 - extracting image assets from mockup of, 191–193
 - as only image in site template, 344
 - page design considerations, 147
 - previewing assets in Code view, 136–137
 - vector graphic formats for, 316
- Compare, synchronization option, 430
- Competition, web design basics, 142
- Compiling CSS code, 120–123
- Components
 - deleting from page layout template, 173–176
 - deleting unused Bootstrap, 267–268
 - removing unneeded template, 209–216
- COMPUTED option, CSS Designer, 91–92
- Conflicts, troubleshooting page layout, 167–170
- Connect To Remote Server icon, putting site online, 423
- Content
 - adding to child page, 244–245
 - structures, basing on lists, 275–278
- Context menu, adding rows to tables, 287
- Conventions, used in this book, 2–3
- Copy and paste
 - Find And Replace, 312
 - menu items, 181–182
 - styles, 45
 - tables, 288–289
- Create New button, 21
- Creative Cloud, 4–6, 160–161
- CSS (cascading style sheets)
 - adding classes to template, 348–351
 - applying styling, 77–78
 - Bootstrap layout modifications, 217–218
 - box model, 75–77
 - cascade theory, 78
 - Code Navigator and, 82–87
 - creating class attributes, 94–95
 - creating gradient background, 186–190
 - creating id attributes, 95
 - CSS3, 95–97
 - descendant theory, 80–81
 - extracting text styling, 182–186
 - as formatting instructions, 77–78
 - formatting multiple elements, 93
 - HTML defaults, 72–75
 - HTML formatting versus, 70–72
 - inheritance theory, 80
 - lesson overview, 68–69
 - overview of, 77–78
 - review Q&A, 98–99
 - rule syntax, 79
 - rules. *See* Rules, CSS
 - shorthand, 93–94
 - specificity theory, 81–82
 - styling heading and text in Bootstrap carousel, 354–356
 - styling layout elements, 162–165
 - styling tables, 290
 - syntax, 79
 - understanding, 70
- CSS (cascading style sheets), troubleshooting
 - with Code Navigator, 82–87
 - with CSS Designer, 87–92
 - with Inspect mode, 28–29
 - page layout conflicts, 167–170
- CSS Designer
 - @Media window, 46
 - heading and text in Bootstrap carousel, 354–356
 - home page link, 373, 376–777
 - layout elements, 162–167
 - layout modifications in Bootstrap, 216–217
 - navigation menu, 398–401
 - overview of, 44–45
 - Properties window, 47–48
 - Selectors window, 46–47
 - semantic text structures, 301–303
 - Sources window, 45
 - synchronizing local and remote sites, 427
 - tag selector integration with, 44
 - text formatting modifications in Bootstrap, 221–223
 - troubleshooting CSS styling, 92
- CSS Designer, All button
 - adding new style sheet to Sources pane, 166
 - CSS rules and, 90
 - defined, 48–49
 - extracting text styling, 184
 - gradient background, 187
 - overview of, 48–49
 - styling heading/text in Bootstrap carousel, 355
 - styling navigation menu, 398–401
 - styling tables with CSS, 290–292
 - synchronizing local/remote sites, 428

- CSS Designer, Current button
 - CSS rules, 90
 - defined, 48–49
 - gradient background, 187
 - layout elements, 164
 - layout modifications in Bootstrap, 216–217
 - predefined layouts, 157–159
 - styling heading/text in Bootstrap carousel, 355
 - text formatting modifications in Bootstrap, 221–223
 - text styling, 183
 - troubleshooting CSS conflicts, 167–170
- CSS preprocessors
 - compiling CSS code, 120–123
 - creating CSS source file, 116–119
 - enabling, 113–116
 - importing other style sheets, 126–127
 - learning more, 127
 - nesting CSS selectors, 124–126
 - overview of, 113
- CSS Preprocessors option, 114
- CSS Source icon, 165
- CSS tab, Property inspector, 41
- CSS3 (cascading style sheets 3), 95–97
- Current mode. *See* CSS Designer, Current button
- Customizing
 - Common toolbar, 110
 - workspace. *See* Workspace, customizing

D

- Data
 - headers help reader identity, 294–295
 - inserting tables from other sources, 298–300
- Declarations, CSS rules, 77–78
- Default option, Target menu, 384
- Defaults, HTML tag, 72–75
- Delete
 - synchronization option, 430
 - template components/attributes, 173–176
 - unnneeded page components, 212
 - unused Bootstrap components, 267–268
- Dependent files, putting site online, 423–424, 426
- Deprecated, defined, 70
- Descendant theory, CSS styles, 80–82
- Description
 - adding metadata to webpage, 229–230
 - creating lists, 268–269
 - importing text, 261–264

- Design view
 - adding data to table manually in, 285–286
 - adding home page link to text in, 375–376
 - adding telephone link in, 402
 - customizing workspace in, 25–26
 - inserting images in, 326–328, 345, 347
 - inserting tables from other sources, 298–300
 - resizing images in, 328–331
 - spell-check webpages only in, 305–306
 - Split view pairs Code view with, 27
 - targeting text and code in, 308
 - template features work properly in, 244
 - updating template in, 247–248, 250
 - working with tables in, 296
- Desktop computers, web design, 143–144, 150–151
- Destination link, creating, 394–396
- Developer workspace, 30, 102–105
- Display
 - controlling table, 296–298
 - Element, Image, and Text, 49–51
- Dithering, 318
- Docking panels, 35
- Document toolbar
 - overview of, 37
 - selecting Code view in, 25
 - selecting Design view in, 26
 - selecting Live view in, 26, 71, 83
 - selecting Split view in, 27
 - selecting Synchronize icon in, 429
- DOM panel
 - adding CSS classes to template, 348–351
 - adding metadata to child page, 246–247
 - creating page layout elements, 179–181
 - creating unordered list, 273
 - editing meta description for pages, 261–262
 - fixing semantic errors, 227–228
 - inserting editable optional region, 239–240
 - inserting new Bootstrap element, 233–236
 - locking element onscreen, 396–398
 - modifying Bootstrap layout, 216
 - removing unnneeded template components, 214–215
- DOM Viewer, 49
- Dots per inch (dpi), printer quality, 317
- Dragging panels, 33–35
- Drop zone, 34
- Duplicate Row icon, Bootstrap, 265–266

E

Editable optional regions, templates
adding images to Bootstrap carousel, 352–354
defined, 232
inserting, 239–241
removing from child page, 251–252

Editable regions, templates
defined, 209
inserting editable region, 236–239
inserting new Bootstrap element, 233–236
overview of, 232

8-bit color space (web-safe color space), 319

Element dialogs, 49–50

Element Display

background image, adding to header, 201
basing content structures on lists, 276
deleting template components/attributes,
173–176
destination links, 394–396
duplicating Bootstrap rows, 265–266
entering alt text, 335
header rows, adding to tables, 294–295
image-based links, 387–388
internal targeted links, 391–392
menu items created with copy/paste, 181–182
modifying Bootstrap layout, 218–220
new Bootstrap structures, 193–194
new menu items, 177–179
overview of, 50
pasting multiple elements, 279
removing unneeded components, 209–210, 213
styling element layout, 163
table captions, 303–305
table properties, 41
text extracted from Photoshop mockup, 171

Elements

creating with DOM panel, 179–181
CSS formatting for multiple, 93
fixing semantic errors, 226–228
frequently used HTML, 60–63
HTML conventions for, 3
HTML tags and, 59
locking onscreen, 396–398
selecting using tag selectors, 43
troubleshooting with Inspect mode, 28–29
writing code automatically with multiple,
106–110

Email hyperlinks

bad actors and, 390
image-based, 387–388
setting up, 385–386
testing, 366–367
text-based, 388–389

Embedded, CSS formatting, 77

Emmet, writing code automatically, 106–110

Empty elements, HTML, 59

Enable Auto Compilation On File Save, CSS
Preprocessors, 114–115

Enable File Check-Out, remote FTP site, 413

Encryption, FTP over SSL/TLS implicit/explicit,
25–26

Error checking, live code, 128–129

Expand icon

expanding code, 134
fixing semantic errors, 226
putting site online, 422–423
synchronizing local and remote sites, 427–428

Explicit encryption (FTP over SSL/TLS), remote
sites, 408

Extensions, 415, 416–417

External hyperlinks, 365–366, 380–384

Extract panel

creating gradient background, 186–191
extracting image assets from mockup, 191–193
extracting text from Photoshop mockup,
170–173
extracting text styling in page layout, 182–186
inserting Photoshop document in webpage,
332–334
styling page layout elements, 161–167
styling predefined layout, 160–161
working with, 36–37

F

Favorite-styles.css, always save changes to, 378

File formats

raster images, 320–321
vector graphics, 316

File Transfer Protocol (FTP), remote sites, 408–414

Files

cloaking, 414–416
previewing completed navigation lesson,
258–260
previewing completed text, list, tables lesson,
258–260
Related Files interface, 42
saving as templates, 232

- Files panel
 - expanding, 427–428
 - interface preference settings, 22–23
 - putting your site online, 422–425
 - site name in, 11, 14
- Find And Replace text, 307–312
- Finishing up page layout, 202–203
- First launch, getting started, 8
- Floating panels, 33–35
- Folders
 - cloaking, 414–416
 - Compiling CSS code, 120–123
 - CSS Preprocessors options, 115–116
 - managing Dreamweaver site, 13
- Fonts, conventions for bolded, 2
- Footer
 - accessing Split Code view, 135
 - adding content to child pages, 244
 - adding telephone link, 401–402
 - containing copyright statement, 109
 - creating page design, 146
 - evaluating page design, 156
 - finishing up layout, 202–203
 - multicursor support, 110
 - new in HTML5, 64–65
 - updating template, 250
- 4-bit color space, 318
- FTP (File Transfer Protocol), remote sites, 408–414

G

- General, CSS Preprocessors option, 114
- Get, synchronization option, 430
- Getting started
 - about this book, 1
 - additional resources, 15
 - bonus material, 8
 - checking for updates, 14–15
 - choosing program color theme, 9
 - conventions in this book, 2–3
 - defining Dreamweaver site, 11–14
 - on first launch, 8
 - installing program, 4–5
 - online content, 6–7
 - prerequisites, 2
 - recommended lesson order, 7–8
 - TinyURLs, 1
 - updating Dreamweaver to latest version, 5–6
 - Windows vs. macOS instructions, 3–4
 - workspace setup, 9–10

- GIF (Graphics Interchange Format), raster images, 320
- Git, version control setup, 51–52
- Gradient backgrounds, 186–191, 400–401
- Groups
 - closing panel, 32–33
 - dragging panels into, 34

H

- Hash mark (#), multicursor support, 111
- Header
 - accessing Split Code view, 135
 - adding background image to, 197–201
 - adding content to child pages, 244
 - adding rows to tables, 294–295
 - creating home page link, 372–375
 - creating page design, 146
 - evaluating page design, 156
 - inserting tables from other sources, 299–300
 - locking element onscreen, 397
 - new in HTML5, 64–65
 - updating template and, 250
 - writing code automatically, 108
- Header rule, 136
- Headings, styling in Bootstrap carousel, 354–356
- High color (16-bit color space), 319
- Home links, creating, 372–377
- Home page
 - completing, 419–422
 - creating, 417–419
 - extensions/naming, 416–417
- Href attributes, 108, 110–111
- HTML comment, code, 112
- HTML (HyperText Markup Language)
 - Dreamweaver as leading editor for, 18–19
 - HTML character entities, 62–63
 - HTML tags, 60–62
 - lesson overview, 56–57
 - review Q&A, 66–67
 - understanding, 58
 - web is based on, 58
 - what's new in HTML5, 63–65
 - where it began, 58–60
- HTML (Hypertext Markup Language)
 - creating code, 102
 - formatting with CSS vs., 70–72
 - images do not appear in code, 324
 - validating webpage code, 230–231
 - writing code automatically, 106–110
 - writing code manually, 103–105

HTML tab, Property inspector, 40–41

HTTP requests, 126–127

Hyperlinks

- adding boilerplate/placeholders, 224–225

- navigation with. *See* Navigation

- prelaunch testing of sitewide, 422

- removing extraneous, 214–216

- testing after putting your site online, 426

- writing code automatically, 107–108

HyperText Markup Language. *See* HTML
(Hypertext Markup Language)

I

Icons, minimizing panels to, 32

Id attributes

- conventions for element references, 3

- creating, 95

- creating destination link, 394–396

- creating internal targeted links, 391–393

- id-based link destinations, 395–396

- targeting id-based link destinations, 395–396

- targeting specific page content, 390

- writing code automatically, 108

Ignore All, spell-check webpages, 306

Ignore option, synchronization, 430

Image-based links, creating, 387–388

Image carousel

- adding CSS classes to template, 348–351

- adding images to Bootstrap, 352–354

- adding template boilerplate/placeholders, 224

- adding to editable optional region, 236, 239–240

- basing content structures on lists, 275–276

- as content model for template in this book, 209

- in Live Code mode, 28

- removing optional region from child page,
251–252

- styling headings and text in Bootstrap, 354–356

- working with editable regions, 232

Image compression algorithm, JPEGs, 320

Image Display, 50

Image placeholders

- basing content structures on lists, 277

- duplicating Bootstrap rows, 265

- in editable regions, 232

- inserting image in Design view, 326–328

- inserting image into webpage, 323–326

- predefined layouts and, 159

- removing unneeded components, 209–212

- removing unused sections from child page,
252–253

- resizing image, 328–331

Image Size dialog, Photoshop, 329–330

Images

- accessing properties, 41

- adding CSS classes to template, 348–351

- adding to Bootstrap carousel, 352–354

- background, adding to header, 197–201

- defining Dreamweaver site folders for, 13

- extracting assets from Photoshop mockup,
191–193

- Insert menu, 340–342

- Insert panel, 342–344

- inserting, 323–326

- inserting in child pages, 356–358

- inserting in Design view, 326–328

- inserting into site template, 344–348

- inserting Photoshop, 331–336

- inserting using Assets panel, 336–338

- lesson overview, 314–315

- for mobile design, 338–340

- previewing completed files, 321–323

- raster graphic, 316–320

- raster image file formats, 320–321

- resizing, 328–331

- review Q&A, 359

- right size/wrong size, 336

- styling headings and text in Bootstrap carousel,
354–356

- vector graphics, 316

- web image basics, 314

Implicit encryption (FTP over SSL/TLS), remote
sites, 408

Importing

- other style sheets, 126–127

- text, 260–264

Index.html, putting site online, 423–424

Inheritance theory, CSS styles, 80, 82

Inline styles, 77, 85, 87

Insert menu, 300, 340–342

Insert panel, 342–344

Insert Rows Or Columns dialog, tables, 287–288

Inspect mode, customizing workspace, 28–29

Installation, Dreamweaver program, 4–5

Instructions, Windows vs. macOS, 3–4

Interface, preferences, 22–25

Internal hyperlinks

- defined, 362

- home links, 372–377

- internal targeted links, 391–393

- relative links, 369–372

- testing, 364

- updating in child pages, 377–379

IP (Internet Protocol) address, remote FTP sites, 410
IPv6 Transfer Mode, default FTP server, 412

J

JPEG (Joint Photographic Experts Group), raster images, 320

K

Keyboard shortcuts
 customizing, 38–40
 Windows vs. macOS, 3–4

L

Laptops, web design questions, 143–144
Launch, getting started, 8
Layers pane, 333
Layout
 Developer workspace, 30
 page. *See* Page layout
 Standard workspace, 29–30
LESS, CSS preprocessor
 choosing Sass vs., 115
 creating CSS source file, 116–119
 defined, 113
 enabling, 113–116
Lesson files for this book, accessing online content, 6–7
Lesson order, recommended in this book, 7–8
 element (list item), 61, 106–107
Line numbers, selecting code, 129–130
Link Checker panel, 403
Linked, applying CSS formatting as, 77
Links. *See* Hyperlinks; Navigation
Linting support, live code error checking, 128–129
List-based content section
 adding boilerplate/placeholders, 226
 as content model for template in this book, 209
 creating new list-based items, 279–282
 in editable regions, 232
 modifying Bootstrap layout, 216
 removing unneeded components, 212–216
 removing unused sections from child page, 252–253
Lists, creating
 content structures based on, 275–278
 deleting unused Bootstrap components, 267–268
 Find And Replace text, 307–312
 lesson overview, 256–257
 new list-based items, 279–282

ordered, 270–271
overview of, 268–270
pasting multiple elements in Live view, 278–279
previewing completed file, 258–260
review Q&A, 313
spell-checking webpages, 305–306
unordered, 271–274

Live code error checking, 128–129

Live Code mode, customizing workspace, 28

Live view

 absolute external links in, 380–384
 activating Live Code mode, 28
 customizing workspace, 26–27
 Dreamweaver usually defaults to, 71
 inserting images in, 347
 pasting elements in, 278–279
 Split view pairs Code view with, 27
 styling heading and text in Bootstrap carousel, 354–356
 styling tables in, 290–292
 swapping with Code view window, 70
 updating template in, 248–249
 web images for mobile design, 335

Local/network, connecting to remote site, 408

Local sites, 408, 427–431

Locking element, screens, 396–398

Logo. *See* Company logo

Lorem generator, 109

M

macOS, vs. Windows instructions, 3–4
<main> element, writing code automatically, 109
Maintain Synchronization Information, 412–414
Manage Sites, remote FTP site setup, 409
Marcotte, Ethan, 145
Margin spacing, web images for mobile design, 340
Master page, template as type of, 208
Math, vector graphics and, 316
@Media pane, CSS Designer
 modifying Bootstrap layout, 217
 overview of, 46
 styling layout elements with Extract, 166
 troubleshooting in, 88–89
Menu items
 creating with copy/paste, 181–182
 inserting new, 176–179
Meta description element
 basing content structures on lists, 275–276
 creating lists, 268–269
 importing text, 261–264
 inserting editable optional region, 241

Metadata
adding to child page, 246–247
inserting, 229–230
updating/replacing for placeholder text in new page, 261

Microsoft Word files, cloaking, 414

Mobile devices
adapting images for, 338–340
developing webpages, 146–147
responsive web design for, 145
third way of site design for, 150–151
web design for, 149–150
web design questions for, 143–144

Mobile-first strategy, web design, 149

Mockups. *See also* Photoshop mockups
extracting image assets from, 191–193
third way of site design, 150–151
web design, 148–149

Monochrome (1-bit color), 318

Multicursor support, working with code, 108

N

Named anchor method, now deprecated, 390

Naming conventions

- class attributes, 94–95
- defining Dreamweaver site, 11–12
- editable regions, 238
- home page, 417
- id attributes, 94–95, 394
- new template file, 159

<nav> element (navigation), 64

Navigation

- adding telephone link, 401–402
- checking page, 402–403
- creating destination link, 394–396
- creating external link, 380–384
- creating home link, 372–377
- creating image-based link, 387–388
- creating internal hyperlinks, 369–379
- creating relative links, 369–372
- creating text-based links with Text Display, 388–390
- creating thumbnails for website, 146
- email links, 385–386
- exercise, adding additional links, 403–404
- hyperlink basics, 362–363
- hyperlink pseudo-classes, 374
- lesson overview, 360–361
- locking element onscreen, 396–398
- previewing completed files, 363–368

- review Q&A, 405
- styling menu, 398–401, 405
- targeting page elements, 390–393
- updating links in child pages, 377–379

Nesting, CSS selectors, 124–126

New Document dialog

- accessing, 21
- creating CSS source file, 117
- writing code manually, 103

New Feature guides, 21

New option, Target menu, 384

O

Online content, materials for this book, 6–7

Open button, 21

Open dialog, 21

Optimization, FTP performance, 412

Optimize slider, inserting Photoshop documents, 334

Options, FTP server, 411–412

Ordered lists, 270–273

Output, resolution effects image, 317

P

Page description, adding metadata, 229–230

Page design, web

- creating, 146–147
- creating wireframes for, 147–149
- evaluating options for, 156
- templates allow control of, 208

Page elements, targeting, 390–393

Page layout

- adding background image to header, 197–201
- creating elements with DOM panel, 179–181
- deleting template components/attributes, 173–176
- design, 146–149
- design options, 156
- Dreamweaver templates and, 208–209
- extracting image assets from mockup, 191–193
- extracting text from Photoshop mockup, 170–173
- extracting text styling, 182–186
- finishing up, 202–203
- gradient background, 186–191
- lesson overview, 154–155
- menu items, 176–179, 181–182
- new Bootstrap structures, 193–197
- predefined layouts, 156–159
- review Q&A, 204–205

- styling elements, 161–167
- styling predefined, 159–161
- troubleshooting CSS conflicts, 167–170
- Panels
 - closing, 32–33
 - dragging, 33
 - floating, 33–34
 - grouping, stacking, and docking, 34–35
 - minimizing/expanding, 31–32
 - working with, 30–31
- Paragraph
 - returns, 263, 264
 - text, 380–381
- _parent option, Target menu, 384
- Parent tag selectors, selecting code, 129–130
- Passive FTP, 412
- Password
 - remote FTP site setup, 410–411
 - troubleshooting FTP connection, 412
- Paste command, 182, 278–279
- Paste Styles, 167, 189
- Personalizing, saving custom workspace, 35–36
- Photoshop
 - documents. *See* PSD (Photoshop document) files
 - resizing images in, 329–330
- Photoshop mockups
 - creating wireframes, 148–149
 - extracting image assets from, 191–193
 - extracting text from, 170–173
 - extracting text styling from, 182–186
 - finishing up layout, 202–203
 - styling predefined layout, 160–161
 - third way of site design, 151
- Pixels fitting in 1 inch (ppi), 317–318
- Pixels, raster graphics and, 316–317
- Placeholder content, writing code automatically, 109
- Placeholder text
 - adding content to child pages, 244–245
 - adding metadata to webpage, 229–230
 - basing content structures on lists, 276–277
 - commenting code, 112–113
 - creating lists, 268–269
 - creating new Bootstrap structures, 194–195
 - creating new elements with DOM panel, 180
 - creating unordered list, 271–273
 - creating with Lorem generator, 109
 - importing text, 263–264
 - inserting metadata, 229–230
 - inserting new Bootstrap element, 234–235
 - inserting new menu items, 177
 - multicursor support and, 110–111
 - pasting elements in Live view, 278–279
 - updating/replacing metadata in new page, 261
 - validating HTML code, 231
 - webpages with, 113
 - writing code automatically, 108–109
- Placeholders
 - adding before converting layout into template, 223–226
 - image. *See* Image placeholders
 - removing unneeded, 209–216
- PNG (Portable Network Graphics), raster images, 321
- Point to File icon, creating hyperlinks, 372
- Position Assist dialog
 - creating new Bootstrap structures for layout, 194
 - inserting new menu items, 177
 - internal targeted links, 391–392
 - overview of, 49–50
- Power positions, page design and, 147
- Ppi (pixels fitting in 1 inch), 317–318
- Predefined layouts, webpage, 156–161
- Preferences
 - interface, 22–25
 - program color theme, 9
- Prelaunch checklist, publishing to web, 422
- Preprocessors. *See* CSS preprocessors
- Prerequisites, to using this book, 2
- Previewing
 - completed image lesson files, 321–323
 - completed navigation lesson files, 363–368
 - completed text, list, tables lesson files, 258–260
- Proof of concept (mockups), webpage design, 148–149
- Properties
 - adding background image to header, 199–201
 - creating semantic text structures, 301–303
 - finishing up layout, 202–203
 - nesting CSS selectors, 124–126
 - styling table cells, 293
 - styling tables with CSS, 291–292
- Properties pane, CSS Designer
 - creating gradient background, 189–190
 - overview of, 47–48
 - styling layout elements, 164–165
 - troubleshooting in, 90–92
- Property inspector
 - adding metadata, 229–230
 - adding rows to tables, 286
 - controlling table display, 297

- creating hyperlinks, 370–372
- creating new Bootstrap structures for layout, 195
- entering alt text, 335
- importing text, 261
- inserting image into webpage, 323–326
- inserting images in Design view, 327
- overview of, 40–41
- resizing images, 328–331
- Proxy host connection, default FTP server option, 412
- PSD (Photoshop document) files
 - cloaking, 414
 - creating wireframes, 149
 - inserting images into webpage, 331–336
 - loading into Extract panel, 36–37
 - styling predefined layout, 160–161
 - uploading into Extract, 37
 - uploading into webpage, 332
- Pseudo-classes, hyperlink, 399
- Publishing to web
 - cloaking folders and files, 414–416
 - completing home page, 419–421
 - creating new home page, 417–419
 - defining remote site, 408–409
 - installing testing server, 414
 - lesson overview, 406–407
 - prelaunch checklist, 422
 - putting your site online, 422–427
 - review Q&A, 432–433
 - setting up remote FTP site, 409–414
 - synchronizing local and remote sites, 427–432
 - wrapping things up, 416–417
- Punctuation, conventions for missing, 3
- Purpose, website design, 142
- Put, synchronization option, 430
- Putting site online, publishing to web, 422–427

Q

- Q&A, review
 - code basics, 138–139
 - CSS basics, 98–99
 - customizing workspace, 54–55
 - HTML basics, 66–67
 - images, 359
 - navigation, 405
 - page layout, 204–205
 - publishing to web, 432–433
 - templates, 255
 - text, lists, and tables, 313
 - web design basics, 152–153

- Questions to ask, website design, 142–144
- Quick Property Inspector icon, 50
- Quick Start, 19–20
- Quick Tag Editor, 194–195

R

- Raster graphics
 - color, 318–320
 - image file formats, 320–321
 - overview of, 316–317
 - resolution, 317–318
 - size, 318
- RDS (Remote Development Services), ColdFusion sites, 409
- Regular expressions, Find And Replace text, 307–312
- Related Files interface
 - accessing Split Code view, 136
 - compiling CSS code, 122
 - importing other style sheets, 126
 - using linting support, 128
 - working with, 42
- Relative hyperlinks, 362–363, 369–372
- Remote Development Services (RDS), ColdFusion sites, 409
- Remote site
 - connecting to, 408–409
 - defining, 408–409
 - FTP site setup, 409–414
 - installing testing server, 414
 - local site vs., 408
 - putting online, 422–427
- Remove Comment icon, 113
- Resolution, raster image quality, 316–317
- Resource folders, cloaking, 414
- Resources, additional, 15
- Responsive web design, 145
- Review. *See* Q&A, review
- Rows
 - adding to tables, 286–288
 - duplicating Bootstrap, 264–267
 - header, adding to tables, 294–295
 - inserting tables from other sources, 299
- Rules, CSS
 - adding background image to header, 197–201
 - adding header rows to tables, 294
 - caption elements, 304–305
 - cascade theory, 78
 - Code Navigator, 82–87
 - controlling table display, 296–298

- creating internal targeted links, 392
- creating links with Text Display, 389
- descendant theory, 80–81
- extracting text styling, 182–186
- finishing up layout, 202–203
- fixing semantic errors, 226–228
 - as formatting instructions, 77–78
- gradient backgrounds, 186–190
- home links, 373–377
- inheritance theory, 80
- inserting tables from other sources, 298–299
- link updates in child pages, 378
- modifying Bootstrap layout, 216–218
- modifying text formatting in Bootstrap, 221–223
- new Bootstrap structures, 196–197
- new list-based items, 281–282
- organizing in style sheets, 373–374
- selectors, 77–78, 124–126
- semantic text structures, 301–303
- specificity theory and, 81–82
- styling headings and text, 355–356
- styling layout elements with Extract, 162–165
- styling navigation menu, 398–400
- styling tables with CSS, 290–292
- synchronizing local and remote sites, 428
- syntax, write or wrong, 79
- troubleshooting with Code Navigator, 82–87
- troubleshooting with CSS Designer, 87–92

S

- Sass CSS preprocessor, 113–119
- Saving
 - all changes to favorite-styles.css, 378
 - custom workspace, 35
 - customized keyboard shortcuts, 40
 - multiple files updated by template, 241
 - new template files, 158
 - remote FTP site setup, 413
 - username and password, 411
- Scalable Vector Graphic (SVG) format, 316
- Scenario, fictitious website design, 144
- Screen, locking element on, 396–398
- Scripting languages, CSS preprocessor, 113
- SCSS preprocessor, 113–119
- Search and replace text, 307–312
- Search engines, getting high ranking on, 229–230, 300–303
- Search Report panel, Find And Replace, 308
- Secure File Transfer Protocol (SFTP), remote sites, 408

- Selecting code, 129–130
- Selectors, CSS rules, 77–78, 124–126
- Selectors pane, CSS Designer
 - extracting text styling, 183–186
 - modifying Bootstrap layout, 216–217
 - modifying text formatting in Bootstrap, 221–223
 - overview of, 46–47
 - styling layout elements with Extract, 166–167
 - troubleshooting in, 88–89, 91, 167–170
- _self option, Target menu, 384
- Semantic errors, fixing, 226–228
- Semantic text structures, creating, 300–303
- Semantic web design, new in HTML5, 64–65
- SFTP (Secure File Transfer Protocol), remote sites, 408
- Shorthand
 - creating CSS source file, 119
 - nesting CSS selectors, 124–126
 - using CSS, 93–94
 - writing code automatically, 106–109
- Show Log option, Update Pages dialog, 249
- Show Set option, CSS Designer
 - creating gradient background with Extract, 187, 190
 - styling layout elements with Extract, 162
 - styling tables with CSS, 291
 - troubleshooting with, 90–91
- Site, defining Dreamweaver, 11–14
- Site Definition dialog, enabling CSS preprocessors, 113–116
- Site Setup dialog
 - cloaking folders and files in, 415–416
 - defining Dreamweaver site, 11–13
 - remote FTP site setup, 409–410
- 16-bit color space (high color), 319
- Size, image
 - and color depth, 319
 - overview of, 318
 - resizing, 328–331
 - right/wrong, 336
- Smartphones
 - responsive web design for, 145
 - third way of site design for, 150–151
 - web design for, 149–150
 - web design questions for, 143–144
 - webpage development for, 146–147
- Source & Output, CSS Preprocessors option, 115
- Source Code option, Related Files interface, 42
- Source file, creating CSS, 116–119

- Sources pane, CSS Designer
 - adding new style sheet to, 165–166
 - creating gradient background with Extract, 187
 - extracting text styling, 183
 - overview of, 45
 - troubleshooting in, 88–89
- Spam, email links and limiting, 390
- Specificity theory, CSS styles, 81–82
- Spell-check
 - prelaunch checklist, 422
 - webpages, 305–306
- Split Code view, 134–135
- Split view
 - comparing HTML to CSS formatting, 70–72
 - customizing workspace, 27
 - inserting images using, 346
 - interface preference settings, 23
- Src field, Property inspector, 324–325
- Stacking panels, 34
- Standard toolbar, 38
- Standard Workspace, 9–10, 29–30
- Start Screen, 19–21
- Starter Templates, 20–21, 157–159
- Statistics, web design questions, 143–144
- Strikethrough, conventions in this book, 3
- Style Rendering command, CSS box model, 76–77
- Style sheets, importing other, 126–127
- Styling. *See* CSS (cascading style sheets); CSS Designer
- SVG (Scalable Vector Graphic) format, 316
- Synchronization
 - of local and remote sites, 427–431
 - options, 430
 - Sync Settings dialog in first launch, 8
 - in workgroup environments, 426
- Synchronize icon, 429

T

Tables

- accessing properties, 41
- adding and formatting caption elements, 303–305
- adding data to, 285–286
- adding/formatting captions, 303–305
- adding header rows to, 294–295
- adding rows to, 286–288
- controlling display of, 296–298
- copying and pasting, 288–289
- creating from scratch, 283–285

- creating semantic text structures, 300–303
- Find And Replace text, 307–312
- inserting from other sources, 298–300
- lesson overview, 256–257
- overview of, 282–283
- previewing completed file, 258–260
- review Q&A, 313
- spell-checking webpages, 305–306
- styling table cells, 292–293
- styling with CSS, 290–292

Tablets

- third way of site design for, 150–151
- web design for, 149–150

Tag name, element references, 3

Tag selectors

- parent, 132–133
- removing unneeded page components, 210–213
- selecting code, 130–132
- styling layout elements with Extract, 163
- working with, 43–44

Tags, HTML

- defaults, 72–75
- new in HTML5, 63–64
- overview of, 59

Targeting

- menu options, 384
- navigation page elements, 390–393

<td> (table data), 62, 73–74, 292–295

Techniques, new in HTML5, 65

Technology, new in HTML5, 65

Telephone link, adding, 401–402

Templates

- accessing predefined layout, 157–159
- adding boilerplate and placeholders, 223–226
- adding CSS classes to, 348–351
- child pages, 242–247
- cloaking, 414, 416
- creating Dreamweaver, 208–209
- creating home page link on, 372–377
- creating new page on site, 261
- deleting components/attributes from page layout, 173–176
- fixing semantic errors, 226–228
- inserting images into site, 344–348
- inserting metadata, 229–230
- lesson overview, 206–207
- locking element onscreen, 396–398
- modifying Bootstrap layout, 216–221
- modifying text formatting in Bootstrap, 221–223
- putting your site online, 426

- removing optional region from child page, 251–252
 - removing unneeded components, 209–216
 - removing unused sections from child page, 252–254
 - review Q&A, 255
 - Starter Templates option, 20
 - updating, 247–251
 - validating HTML code, 230–231
 - Templates, editable optional regions
 - adding images to Bootstrap carousel, 352–354
 - defined, 232
 - inserting, 239–241
 - removing from child page, 251–252
 - Templates, editable regions
 - defined, 209
 - inserting editable region, 236–239
 - inserting new Bootstrap element, 233–236
 - overview of, 232
 - Templates tab, Asset panel, 261
 - Testing
 - all links created on every page, 379
 - checking site links, 402–403
 - hyperlinks, 383–384
 - Testing server, installing, 414
 - Text
 - adding home page link to, 375–376
 - basing content structures on lists, 276–278
 - conventions for bolded, 2
 - converting into HTML list, 270
 - converting into ordered list, 270–271
 - converting into unordered list, 271–273
 - creating absolute external link in Live view, 380–384
 - creating Bootstrap structures for, 195–197
 - creating gradient background, 190
 - creating link from any range of, 373
 - creating new list-based items, 279–282
 - creating semantic structures, 300–303
 - creating tables from scratch, 283–286
 - deleting unused Bootstrap components, 267–268
 - duplicating Bootstrap rows, 264–267
 - extracting from Photoshop mockup, 170–173
 - extracting styling for page layout, 182–186
 - Find And Replace feature, 307–312
 - finishing up layout, 202–203
 - importing, 260–264
 - lesson overview, 256–257
 - paragraph, 380–381
 - placeholder. *See* Placeholder text
 - previewing completed file, 258–260
 - setting up email links, 385–386
 - spell-checking webpages, 305–306
 - styling in Bootstrap carousel, 354–356
 - styling tables with CSS, 290–292
 - Text Display
 - absolute external links, 383
 - email links, 386
 - image-based links, 388
 - new elements with DOM panel, 180
 - new menu items, 178
 - overview of, 50–51
 - targeting id-based link destinations, 395–396
 - text-based links, 388–390
 - <th> (table header), 62, 73–74, 292–295
 - 32-bit color space (alpha transparency), 319, 321
 - Thumbnails, web design, 146–148
 - TinyURLs, in this book, 1
 - Tips, New Feature guides, 21
 - Title, adding to webpage, 229
 - Toggle Size Constraint icon, resizing images, 328–329
 - Toolbars
 - Common, 38
 - Document, 37
 - Standard, 38
 - _top option, Target menu, 384
 - Troubleshooting
 - CSS conflicts in page layout, 167–170
 - CSS formatting with Code Navigator, 82–87
 - CSS formatting with CSS Designer, 87–92
 - CSS with Inspect mode, 28–29
 - FTP connections, 412
 - live code errors, with linting support, 128–129
 - semantic errors, 226–228
 - True color (24-bit color space), 319
 - Typekit fonts, web design mockup, 149
- ## U
- element, unordered lists
 - basing content structures on lists, 277
 - collapsing code based on, 134
 - creating gradient background, 187
 - creating unordered list, 271
 - defined, 62
 - HTML navigation menus based on, 106–107
 - nesting CSS selectors, 125
 - using parent tag selectors, 133
 - using tag selectors, 131
 - Uniform resource locators (URLs), 1, 410

- Unordered lists
 - creating, 271–273
 - writing code automatically, 106–107
- Update Pages dialog, inserting images, 348
- Updates
 - after adding additional site links, 404
 - checking for software, 14–15
 - Dreamweaver, 5–6
 - links in child pages, 377–379
 - template, 247–251
- Uploading
 - PSD into Extract panel, 37
 - PSD into webpage, 332
 - PSD predefined layout, 160–161
- URLs (uniform resource locators), 1, 410
- Username, 410–412

V

- Validation, HTML webpage code, 230–231
- Variables, creating CSS source file, 117–118
- Vector graphics, 316
- Version control setup, 51–52
- Views
 - Code view, 25
 - Design view, 25–26
 - Inspect mode, 28–29
 - Live Code mode, 28
 - Live view, 26–27
 - Split view, 27
- Visual Media Query (VMQ) interface, 49

W

- W3C Validator Notification dialog, HTML code, 231
- W3Schools, access/usage statistics, 143–144
- Web design basics
 - lesson overview, 140–141
 - mobile devices, 149–150
 - page design, 146–147
 - questions to ask, 142–144
 - responsive web design, 145
 - review Q&A, 152–153
 - scenario, 144
 - third way, 150–151
 - thumbnails, 146
 - wireframes, 147–149

- Web Edition, online content for this book, 6–7
- Web-safe color space (8-bit color), 319
- WebDav (Web Distributed Authoring and Versioning), 409
- Webpage, basic structure of, 60
- Weight, CSS specificity theory and, 81–82
- Whitespace, Find And Replace ignores, 308
- Window menu, panels listed in, 31
- Windows, vs. macOS instructions, 3–4
- Wireframes, creating website, 147–149
- Workspace, customizing
 - CSS Designer, 44–49
 - custom keyboard shortcuts, 38–40
 - displays and inspectors, 50–51
 - DOM Viewer, 49
 - element dialogs, displays, and inspectors, 49–50
 - interface preferences, 22–25
 - layout selection, 29–30
 - lesson overview, 16–17
 - New Feature guides, 21
 - panels, 30–35
 - personalizing, 35–37
 - Property inspector, 40–41
 - Related Files interface, 42
 - review Q&A, 54
 - saving custom workspace, 35
 - Start Screen, 19–21
 - switching and splitting views, 25–29
 - tag selectors, 43–44
 - toolbars, 37–38
 - touring workspace, 18–19
 - version control setup, 51–52
 - Visual Media Query (VMQ) interface, 49
- Workspace, setting up, 9–10
- Wrapping things up, 416–417
- Writing code
 - automatically, 106–110
 - manually, 103–105
- WYSIWYG editor, Design view and, 25–26