



Adobe Dreamweaver

2021
release




Classroom in a Book[®]

The official training workbook from Adobe

Jim Maivald

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

<i>Accessing image properties</i>	41
<i>Accessing table properties</i>	41
Using the Related Files interface	42
Using tag selectors	43
Using the CSS Designer	44
<i>Sources</i>	45
<i>@Media</i>	46
<i>Selectors</i>	46
<i>Properties</i>	47
<i>All and Current modes</i>	48
Using the Visual Media Query (VMQ) interface	49
Using the DOM Viewer	49
Using element dialogs, displays, and inspectors	49
<i>Position Assist dialog</i>	49
<i>Element Display</i>	50
<i>Image Display</i>	50
<i>Text Display</i>	50
<i>Setting up version control in Dreamweaver</i>	51
Exploring, experimenting, and learning	53
Review questions	54
Review answers	55

2 HTML BASICS 56



What is HTML?	58
Where did HTML begin?	58
Frequently used HTML elements	60
<i>HTML tags</i>	60
<i>HTML character entities</i>	62
What's new in HTML5	63
<i>HTML5 tags</i>	63
<i>Semantic web design</i>	64
<i>New techniques and technology</i>	65
Review questions	66
Review answers	67

3 CSS BASICS

68





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

4 WORKING WITH CODE

100



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



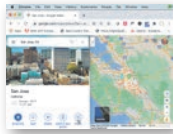
Creating Dreamweaver templates	208
Removing unneeded components	209
Modifying a Bootstrap layout	216
Modifying text formatting in a Bootstrap element	221
Adding template boilerplate and placeholders	223
Fixing semantic errors	226
Inserting metadata	229
Validating HTML code	230
Working with editable regions	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

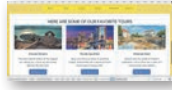
10 WORKING WITH NAVIGATION

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**

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

● **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."

Sample pages

4

WORKING WITH CODE

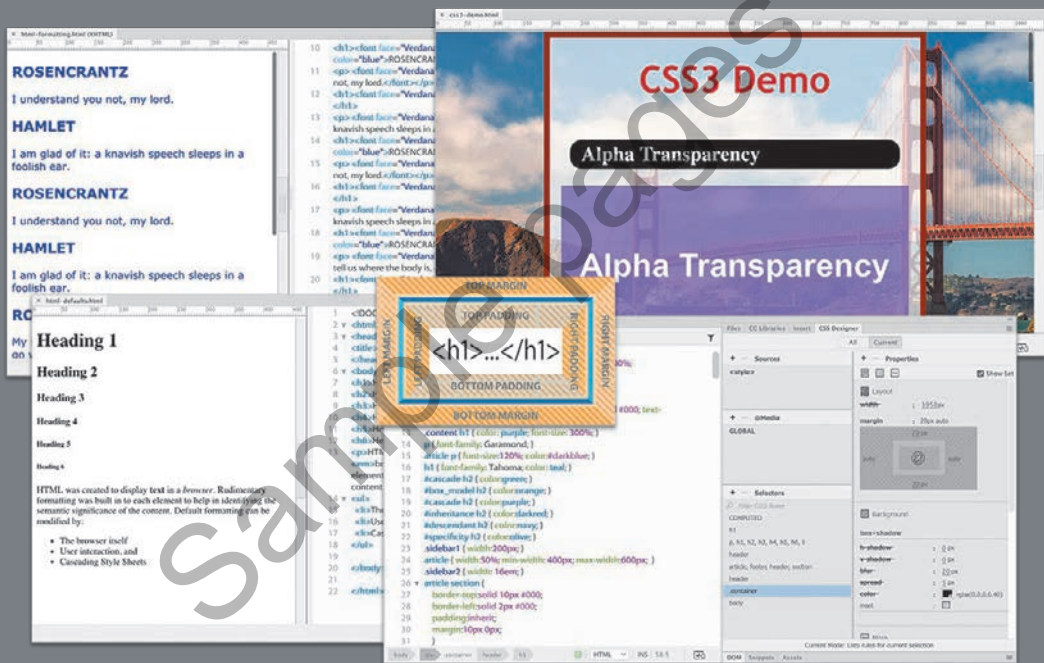
Lesson overview

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.

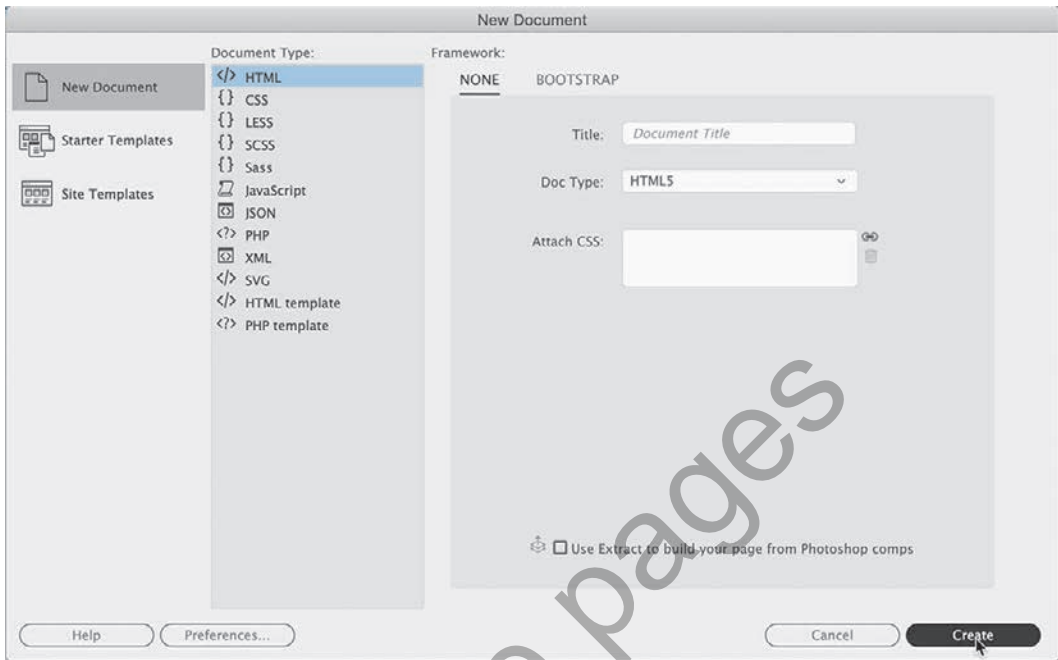
- 1 Define a site based on the lesson04 folder downloaded from your account page, as described in the “Getting Started” section at the beginning of the book.
- 2 Select Developer from the Workspace menu.



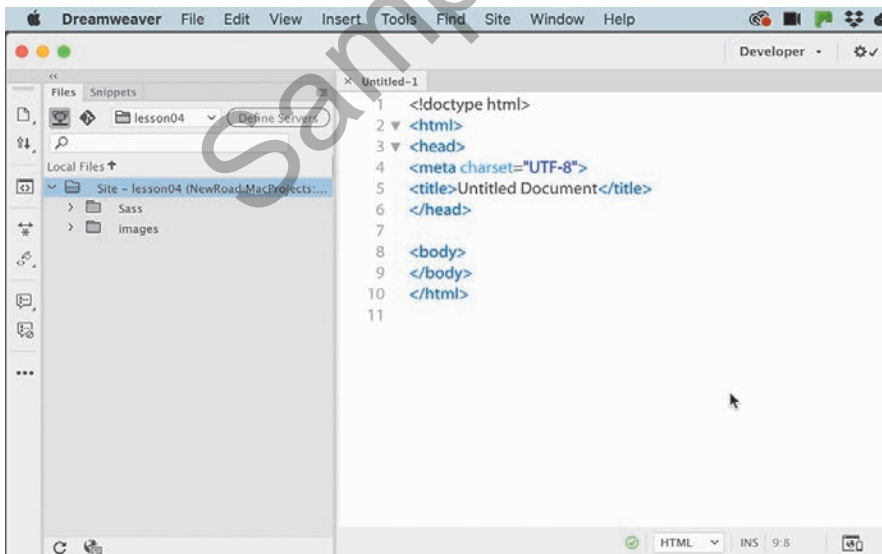
All the code-editing tools work identically in either workspace, but the Developer workspace focuses on the Code view window and provides a better experience for the following exercises.

3 Choose File > New.

The New Document dialog appears.



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

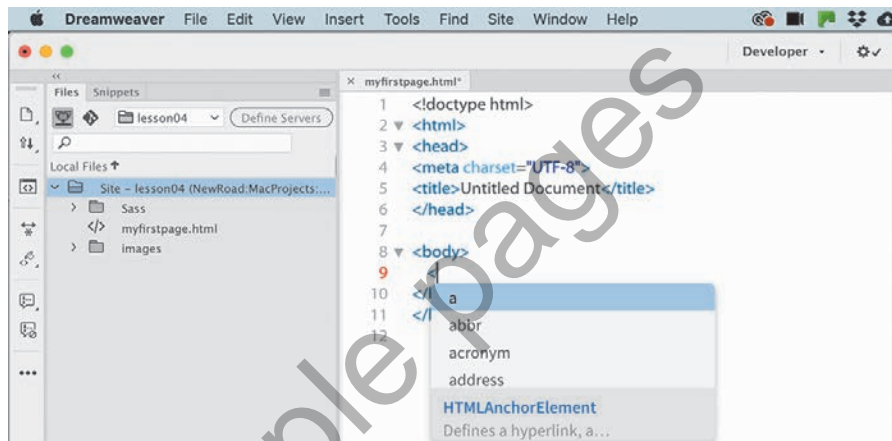


● **Note:** In all screen shots, we use the Classic color theme, which can be selected in Preferences. See the “Getting Started” section at the beginning of the book for more details.

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.

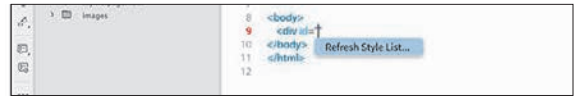


The tag name `div` is inserted in the code. The cursor remains at the end of the tag name, waiting for your next input. For example, you could complete the tag name or enter various HTML attributes. Let's add an `id` attribute to the `div` element.

- 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.



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.

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.

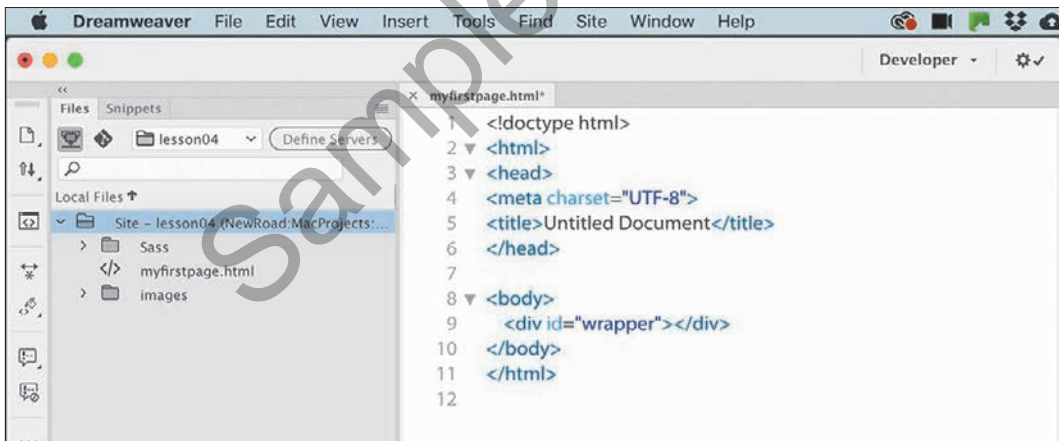
- 13 Type `wrapper` and press the Right Arrow key once.

The cursor moves outside the closing quotation mark.

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

- 14 Type `></`

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.



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.

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.

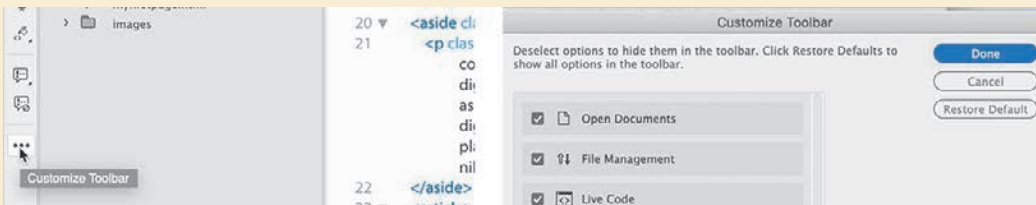
- 1 If necessary, open **myfirstpage.html** as it appears at the end of the previous exercise.

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.

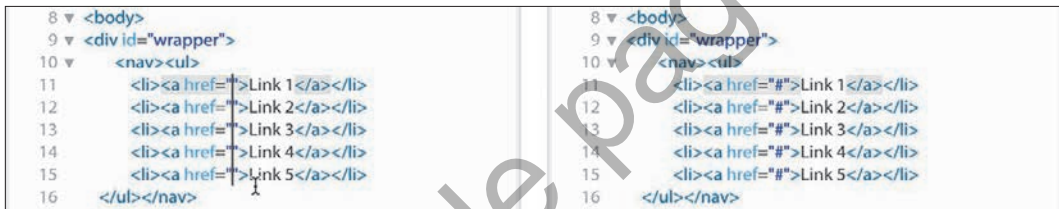
Normally, you would have to add a hash mark (#) to each attribute individually. Multicursor support makes this task much easier, but don't be surprised if it takes you a little practice. Note that all the link attributes are aligned vertically on consecutive lines.

- 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 #

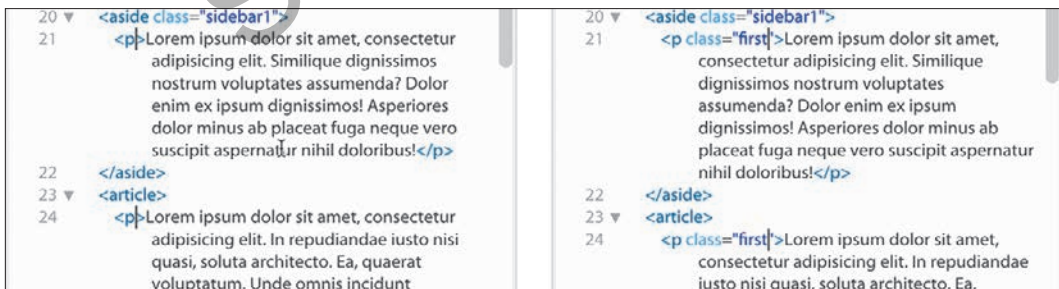
The hash mark (#) appears in all five attributes at the same time.



The screenshot shows two side-by-side views of HTML code. The left view shows five lines of code, each with a cursor (a vertical line) positioned between the double quotes of the `href=""` attribute in the `<a>` tags. The right view shows the same code after the hash mark (#) has been typed, so each `href` attribute now contains `href="#"`.

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 views of HTML code. The left view shows three lines of code, each with a cursor (a vertical line) positioned between the `p` and the `>` bracket in the opening `<p>` tags. The right view shows the same code after the spacebar has been pressed and `class="first"` has been typed, so each opening `<p>` tag now contains `<p class="first">`.


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

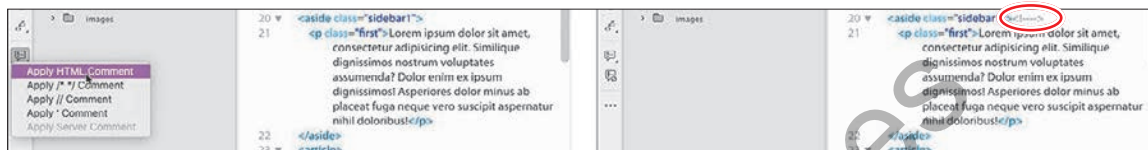
- 7 Save the file.

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 .

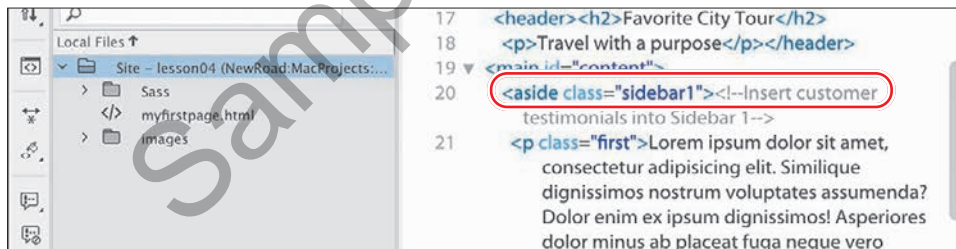


A pop-up menu appears with several comment options. Dreamweaver supports comment markup for various web-compatible languages, including HTML, CSS, JavaScript, and PHP.


- 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**

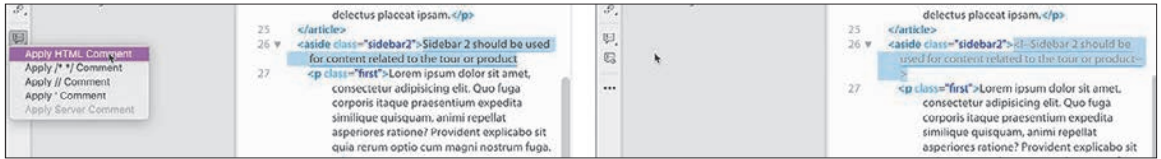



The comment appears in gray between the `<!--` and `-->` markup. The tool can also apply comment markup to existing text.

- 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 use preprocessors with Dreamweaver as well as what advantages they offer compared to a regular CSS workflow.

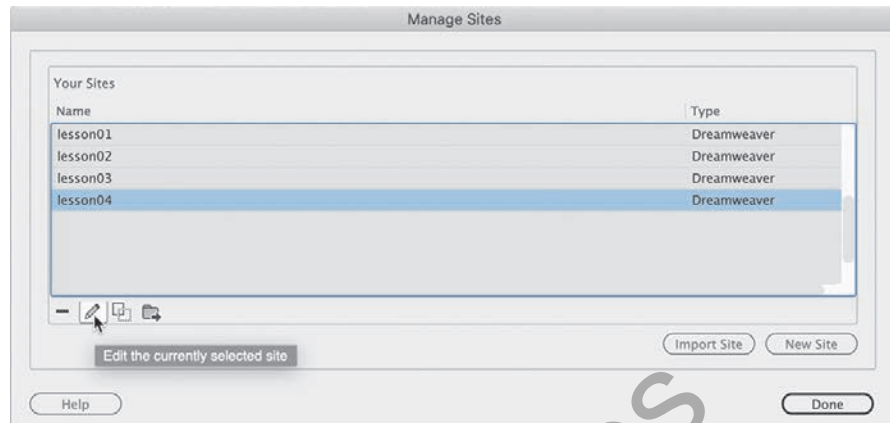
Enabling a preprocessor

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.

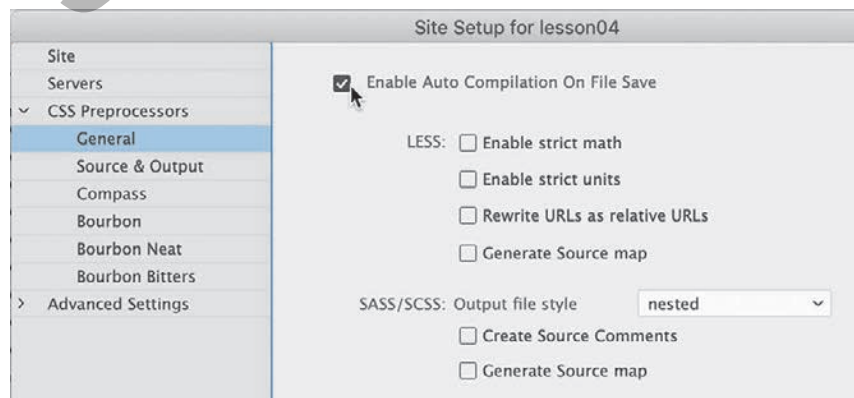
- 3 Select the **CSS Preprocessors** option in the Site Definition dialog.

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:

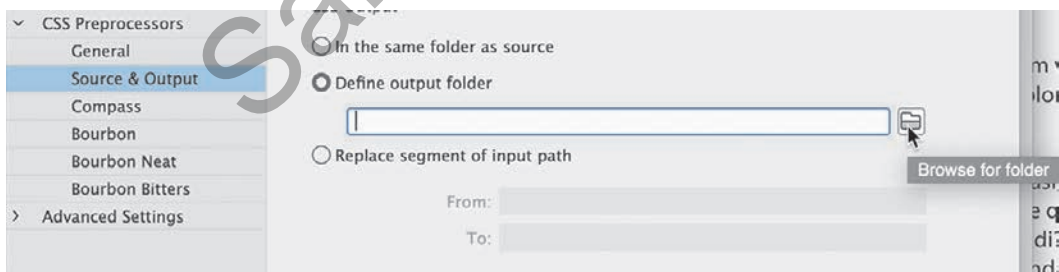
- blog.udemy.com/less-vs-sass/
- css-tricks.com/sass-vs-less/
- keycdn.com/blog/sass-vs-less

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.

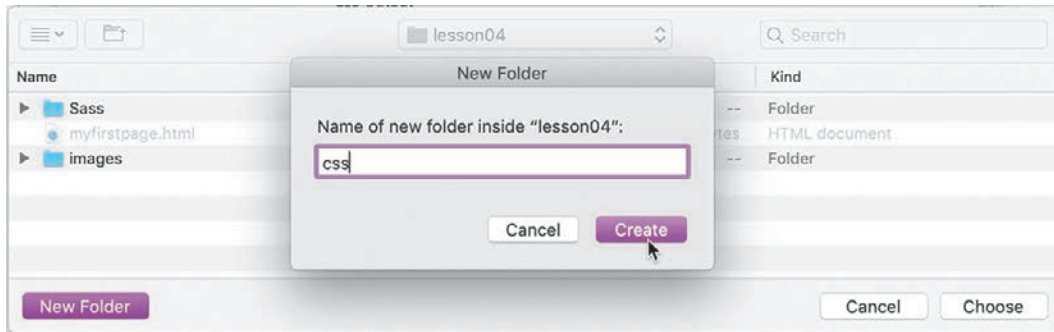



8 Click the Browse For Folder icon .

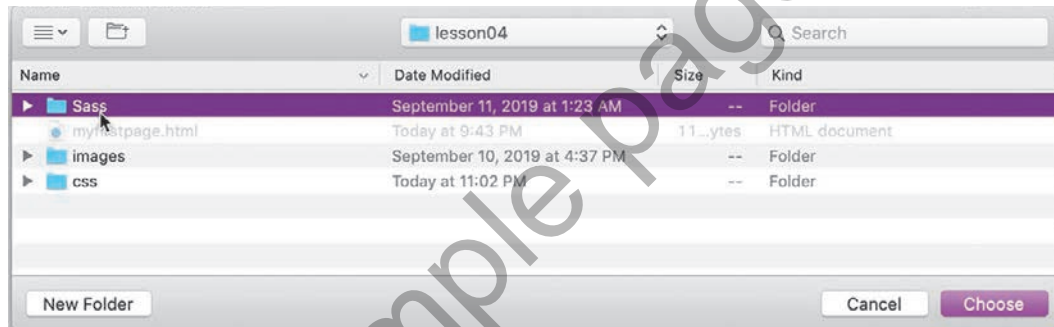
A file browser dialog appears.

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.

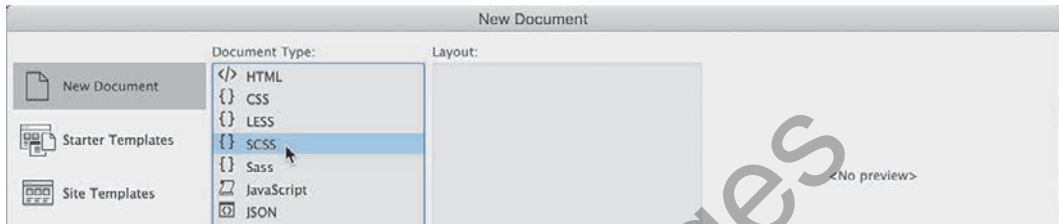
The webpage is unstyled at the moment.

- 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.

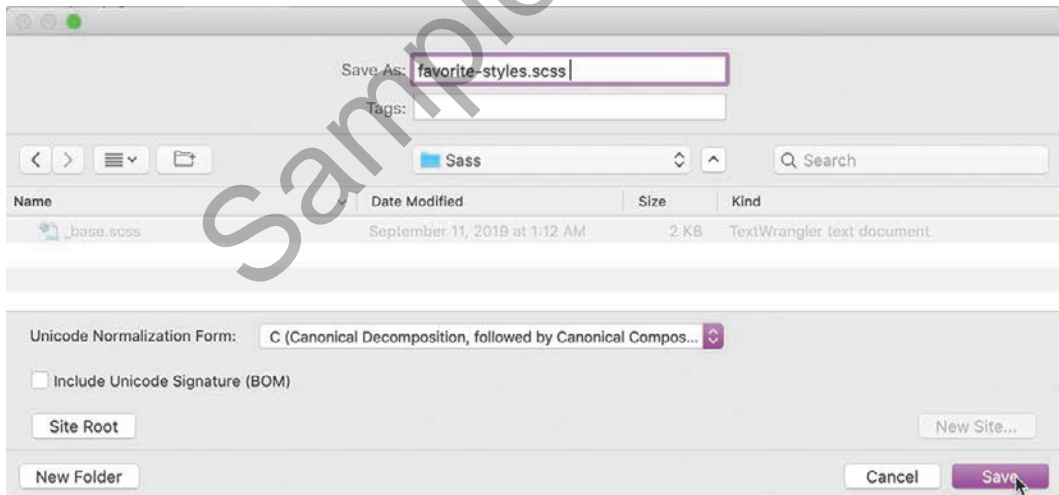
- 5 Choose New Document > SCSS.

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.



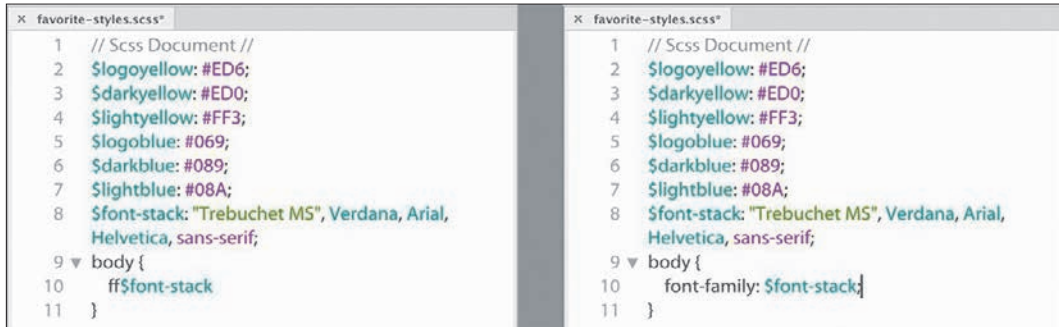
Entering the variables on separate lines makes them easier to read and edit but does not affect how they perform. Just make sure you add a semicolon (;) at the end of each variable.

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.



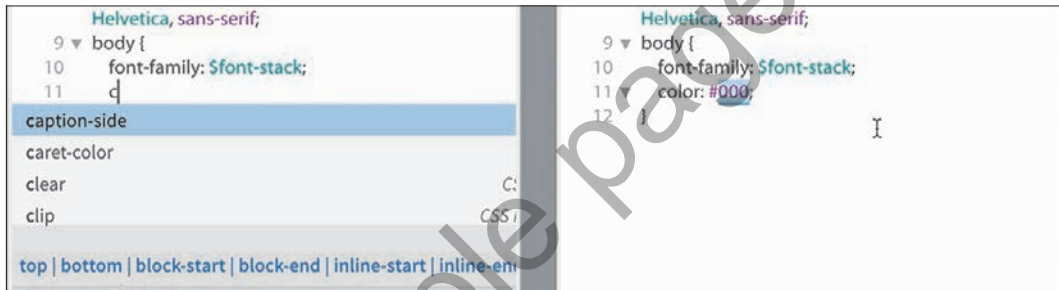
```
1 // Scss Document //
2 $logoyellow: #ED6;
3 $darkyellow: #ED0;
4 $slightyellow: #FF3;
5 $logoblue: #069;
6 $darkblue: #089;
7 $slightblue: #08A;
8 $font-stack: "Trebuchet MS", Verdana, Arial,
  Helvetica, sans-serif;
9 body {
10 ff$font-stack
11 }
```

```
1 // Scss Document //
2 $logoyellow: #ED6;
3 $darkyellow: #ED0;
4 $slightyellow: #FF3;
5 $logoblue: #069;
6 $darkblue: #089;
7 $slightblue: #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.



```
  Helvetica, sans-serif;
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
```

```
  Helvetica, sans-serif;
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.



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

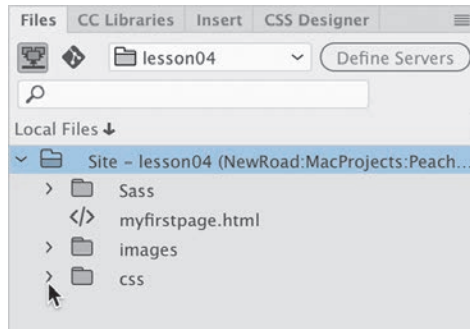
```
  Helvetica, sans-serif;
9 body {
10 font-family: $font-stack;
11 color: #000;
12 margin: 0;
13 }
```

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

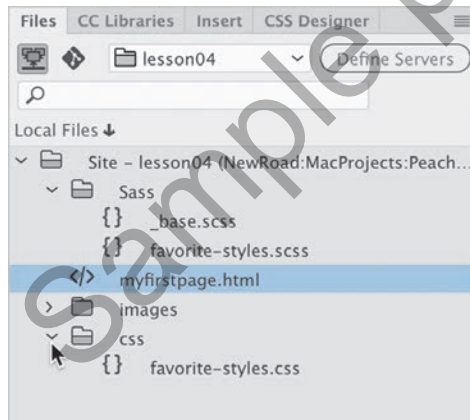
You have completed the specifications for the body element. But you have not created the styling directly in a CSS file. Your entries were made entirely in the SCSS source file. In this exercise, you will see how the compiler that is built into Dreamweaver generates the CSS output.

- 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.

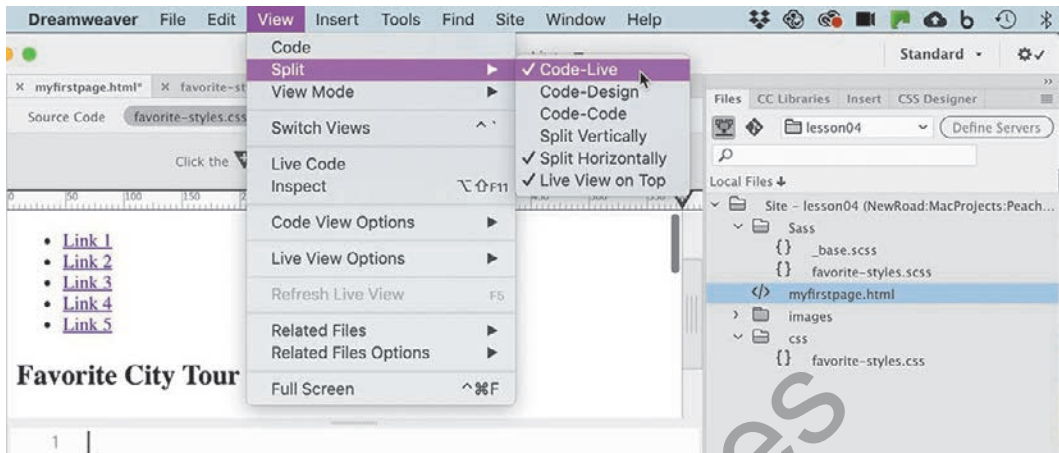
- 2 Expand the view of the css and Sass folders.



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.

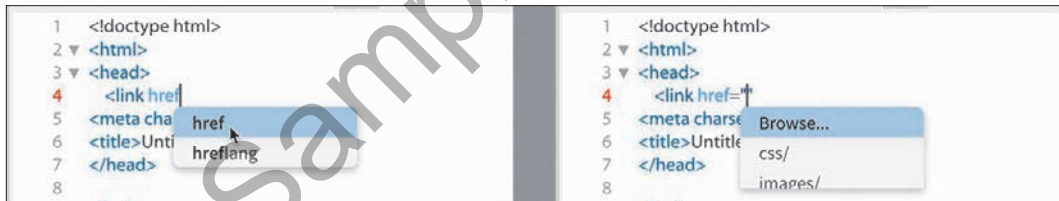
● **Note:** The **favorite-styles.css** file should have been created automatically in the previous exercise when the SCSS file was saved. If you do not see the .css file, you may need to shut down and relaunch Dreamweaver.

- 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**.