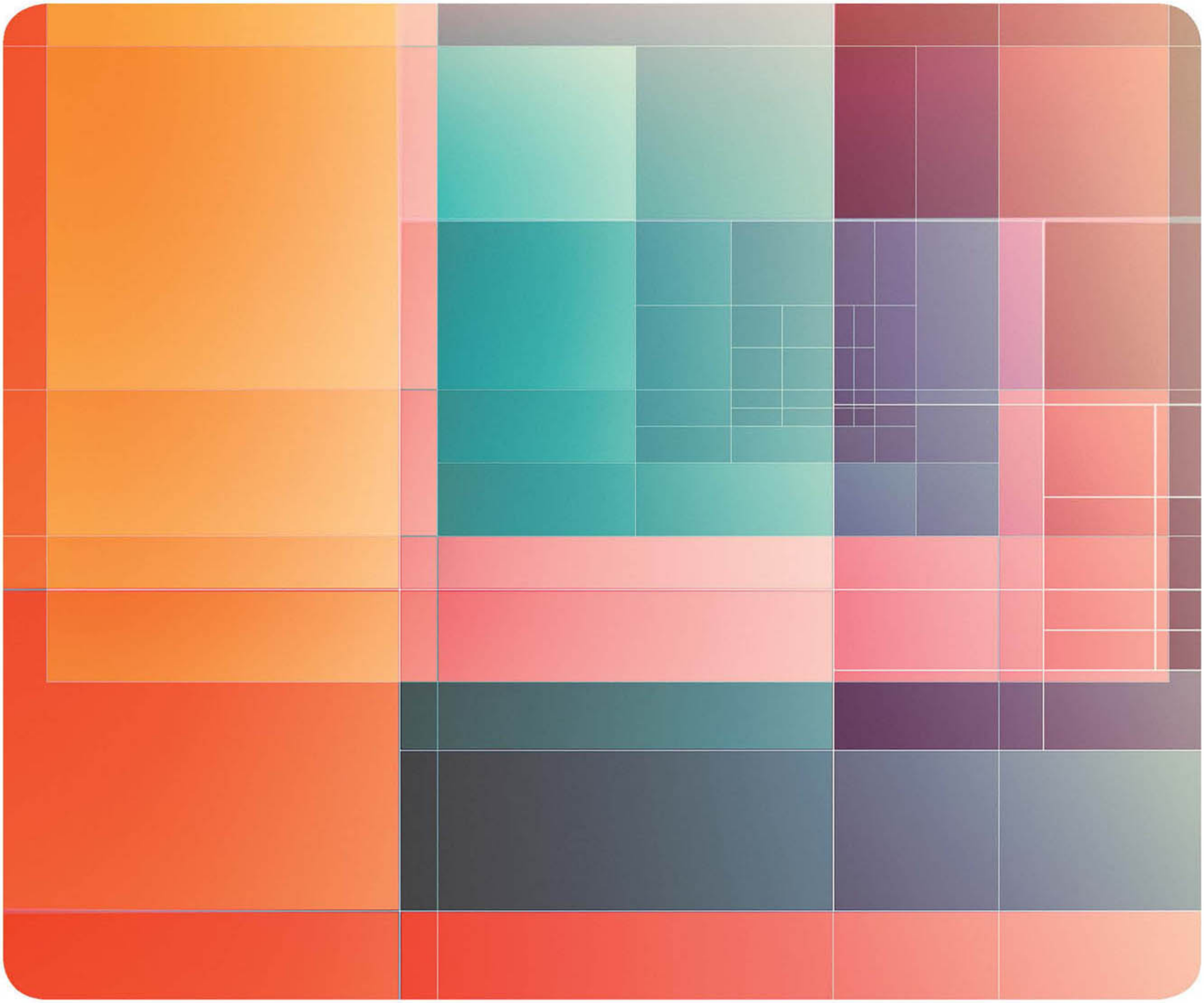
Adobe Dreamweaver

2021 release

Adobe



Classroom in a Book®

The official training workbook from Adobe

Jim Maivald

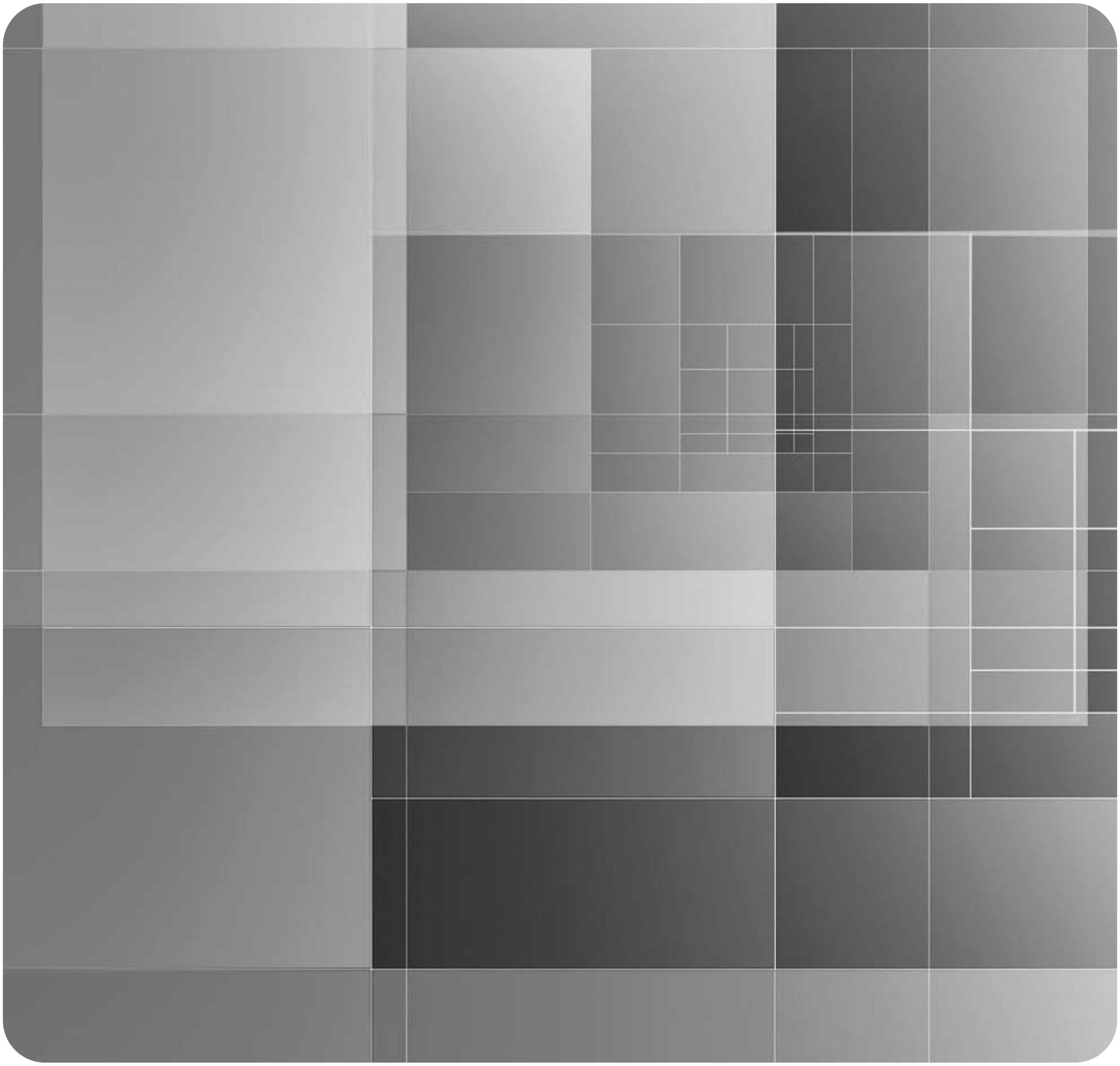
FREE SAMPLE CHAPTER

SHARE WITH OTHERS

Adobe Dreamweaver

2021 release

Adobe



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

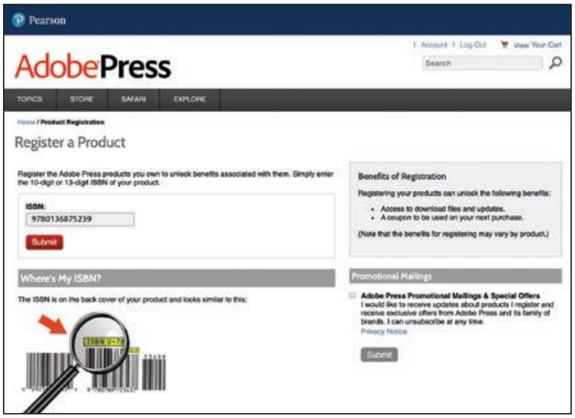
WHERE ARE THE LESSON FILES?

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.adobepress.com/DreamweaverCIB2021.

2 Sign in or create a new account.

3 Click Submit.


Pearson
AdobePress
Account | Log Out | View Your Cart
Search
TOPICS
STORE
SAFARI
EXPLORE
Home / Product Registration

Register a Product

Register the Adobe Press products you own to unlock benefits associated with them. Simply enter the 10-digit or 13-digit ISBN of your product.

ISBN: 9780136875239

Submit

Benefits of Registration

Registering your products can unlock the following benefits:

- Access to download files and updates.
- A coupon to be used on your next purchase.

(Note that the benefits for registering may vary by product.)

Where's My ISBN?

The ISBN is on the back cover of your product and looks similar to this:

Promotional Mailings

□ Adobe Press Promotional Mailings & Special Offers
I would like to receive updates about products I register and receive exclusive offers from Adobe Press and its family of brands. I can unsubscribe at any time.
Privacy Notice

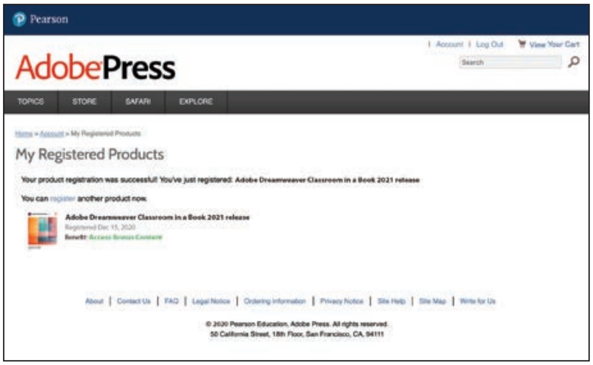
Submit

Note: If you encounter problems registering your product or accessing the lesson files or web edition, go to www.adobepress.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.


Pearson
AdobePress
Account | Log Out | View Your Cart
Search
TOPICS
STORE
SAFARI
EXPLORE
Home » Account » My Registered Products

My Registered Products

Your product registration was successful! You've just registered: Adobe Dreamweaver Classroom in a Book 2021 release

You can register another product now.

Adobe Dreamweaver Classroom in a Book 2021 release
Registered Dec 15, 2020
Benefit: Access Bonus Content

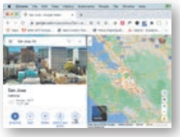
About | Contact Us | FAQ | Legal Notice | Ordering Information | Privacy Notice | Site Help | Site Map | Write for Us

© 2020 Pearson Education, Adobe Press. All rights reserved.
50 California Street, 18th Floor, San Francisco, CA, 94111

Note: If you purchased a digital product directly from www.adobepress.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

CONTACT FAVORITE CITY TOUR



● **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."

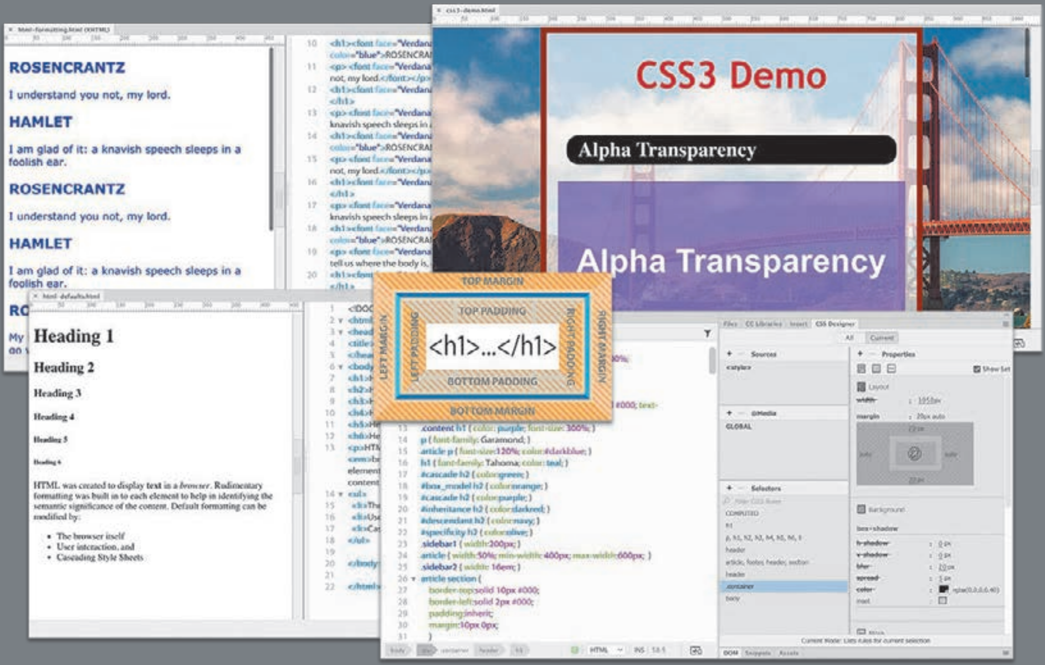
4 WORKING WITH CODE

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.



ROSENCRANTZ

I understand you not, my lord.

HAMLET

I am glad of it: a knavish speech sleeps in a foolish ear.

ROSENCRANTZ

I understand you not, my lord.

HAMLET

I am glad of it: a knavish speech sleeps in a foolish ear.

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

HTML was created to display **text** in a *browser*. Rudimentary formatting was built in to each element to help in identifying the semantic significance of the content. Default formatting can be modified by:

- The browser itself
- User interaction, and
- Cascading Style Sheets

```
.content h1 { color: purple; font-size: 300%; }
p { font-family: Garamond; }
article p { font-size:120%; color:#darkblue; }
h1 { font-family: Tahoma; color: teal; }
#cascade h2 { color:green; }
#box_model h2 { color:orange; }
#cascade h2 { color:purple; }
#inheritance h2 { color:darkred; }
#descendant h2 { color:navy; }
#specificity h2 { color:olive; }
.sidebar1 { width:200px; }
article { width:50%; min-width: 400px; max-width:600px; }
.sidebar2 { width: 16em; }
article section {
  border-top:solid 10px #000;
  border-left:solid 2px #000;
  padding:inherit;
  margin:10px 0px;
  }
```

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.

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.

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

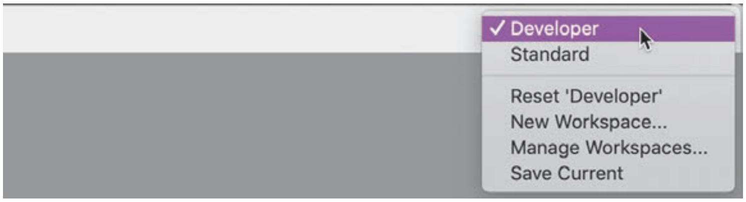
You'll often find that a specific task is actually easier to accomplish in Code view than in Live view or Design view alone. In the following exercises, you'll learn more about how Dreamweaver makes working with the code an effortless and surprisingly enjoyable task.

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.

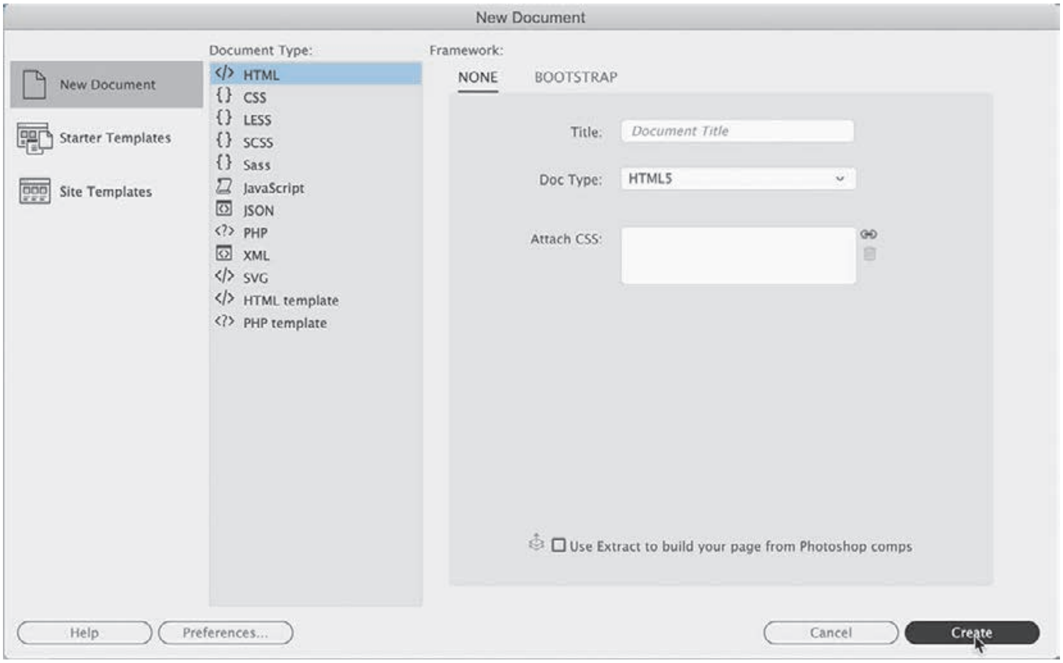
2 Select Developer from the Workspace menu.



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.

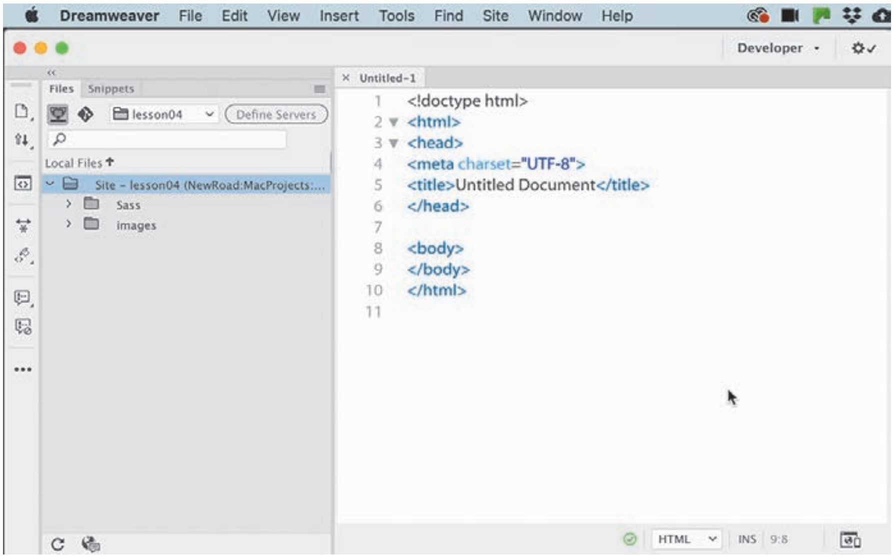


Title: Document Title

Doc Type: HTML5

☐ Use Extract to build your page from Photoshop comps

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



```
<!doctype html>
<html>
<head>
<meta charset="UTF-8">
<title>Untitled Document</title>
</head>

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

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.

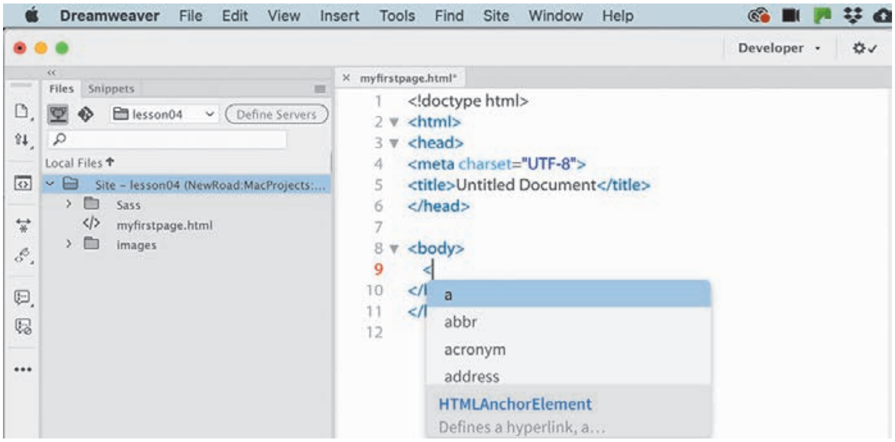
Dreamweaver creates the basic structure of a webpage automatically. The cursor will probably appear at the beginning of the code.

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 **<**



```
<!doctype html>
<html>
<head>
<meta charset="UTF-8">
<title>Untitled Document</title>
</head>

<body>
  <
</
</
```

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

● **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.

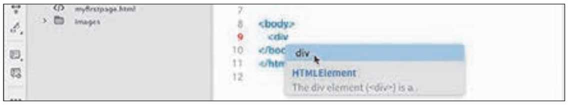
8 Type **d**

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.

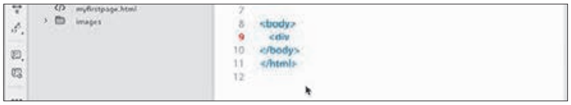
9 Press the Down Arrow key.

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.



```
<body>
  <div
</boc
</htm
```



```
<body>
  <div
</body>
</html>
```

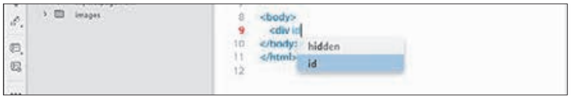
The tag name `div` is inserted in the code. The cursor remains at the end of the tag name, waiting for your next input. For example, you could complete the tag name or enter various HTML attributes. Let's add an `id` attribute to the `div` element.

11 Press the spacebar to insert a space.

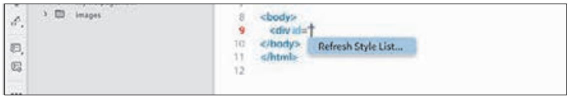
The hinting menu opens again, displaying a different list; this time the list contains various appropriate HTML attributes.

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.



```
<body>
  <div id
</body>
</html>
```



```
<body>
  <div id=
</body>
</html>
```

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.

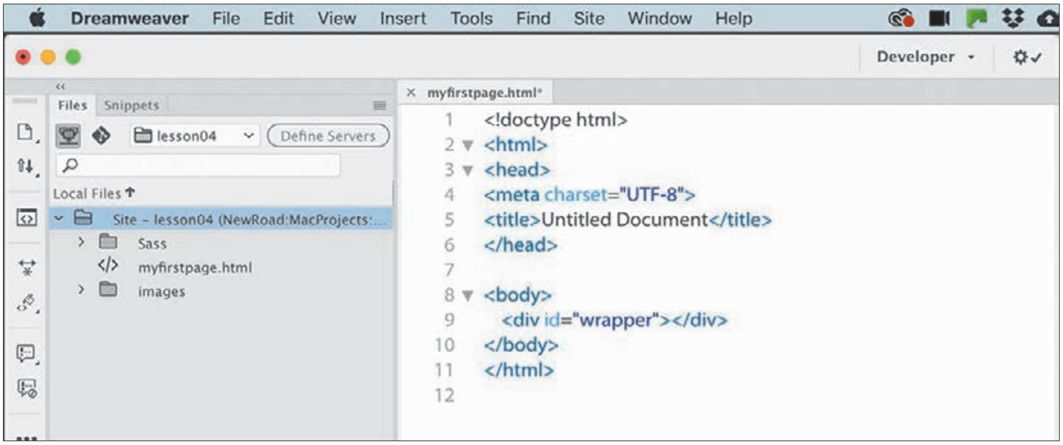
13 Type **`wrapper`** and press the Right Arrow key once.

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.



```
<!doctype html>
<html>
<head>
<meta charset="UTF-8">
<title>Untitled Document</title>
</head>

<body>
  <div id="wrapper"></div>
</body>
</html>
```

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

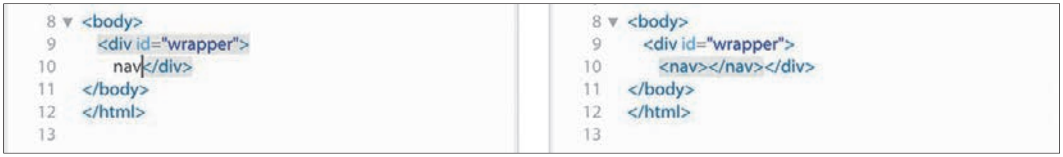
Emmet is a web-developer toolkit that was added to a previous version of Dreamweaver and enables you to supercharge your code-writing tasks. When you enter shorthand characters and operators, Emmet enables you to create whole blocks of code with just a few keystrokes. To experience the power of Emmet,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



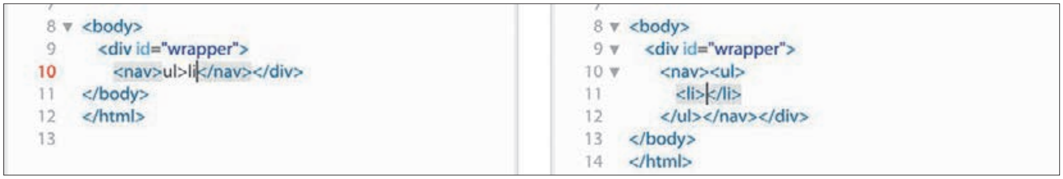
```
<body>
  <div id="wrapper">
    nav</div>
</body>
</html>
```

```
<body>
  <div id="wrapper">
    <nav></nav></div>
</body>
</html>
```

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 `<ul>` element with one or more child `<li>` 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.



```
<body>
  <div id="wrapper">
    <nav>ul>li</nav></div>
</body>
</html>
```

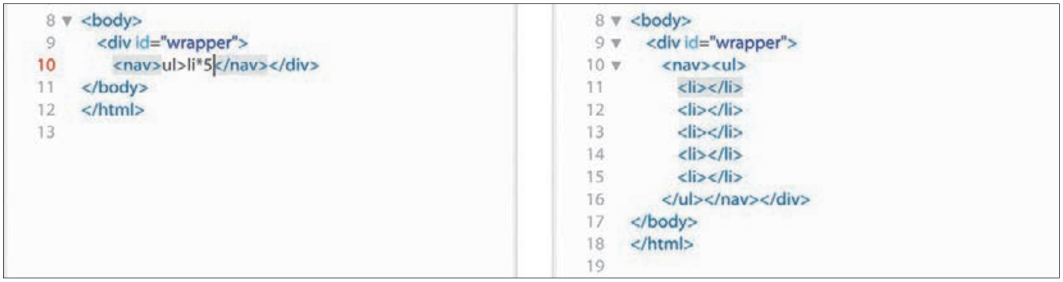
```
<body>
  <div id="wrapper">
    <nav><ul>
      <li></li>
    </ul></nav></div>
</body>
</html>
```

A `<ul>` 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.



```
<body>
  <div id="wrapper">
    <nav>ul>li*5</nav></div>
</body>
</html>
```

```
<body>
  <div id="wrapper">
    <nav><ul>
      <li></li>
      <li></li>
      <li></li>
      <li></li>
      <li></li>
    </ul></nav></div>
</body>
</html>
```

A new unordered list appears, this time with five `<li>` elements. The asterisk (*) is the mathematical symbol for multiplication, so this latest change says "`<li>` 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`

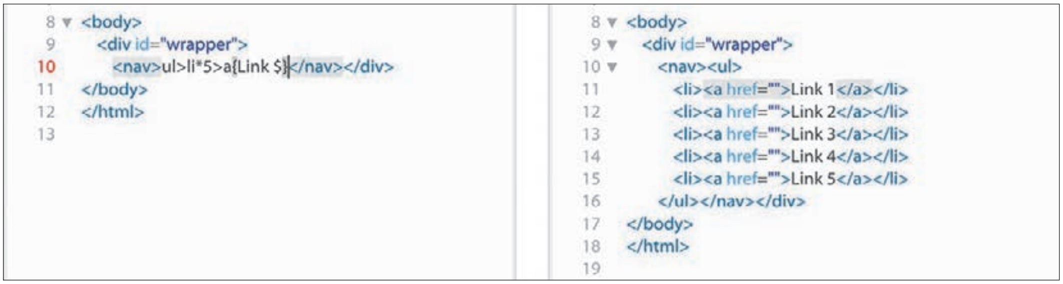
If you guessed that adding the markup `>a` would create a hyperlink child element for each link item, you are correct. Emmet can also create placeholder content. Let's use it to insert some text in each link item.

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.



```
<body>
  <div id="wrapper">
    <nav>ul>li*5>a{Link $}</nav></div>
</body>
</html>
```

```
<body>
  <div id="wrapper">
    <nav><ul>
      <li><a href="">Link 1</a></li>
      <li><a href="">Link 2</a></li>
      <li><a href="">Link 3</a></li>
      <li><a href="">Link 4</a></li>
      <li><a href="">Link 5</a></li>
    </ul></nav></div>
</body>
</html>
```

● **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.

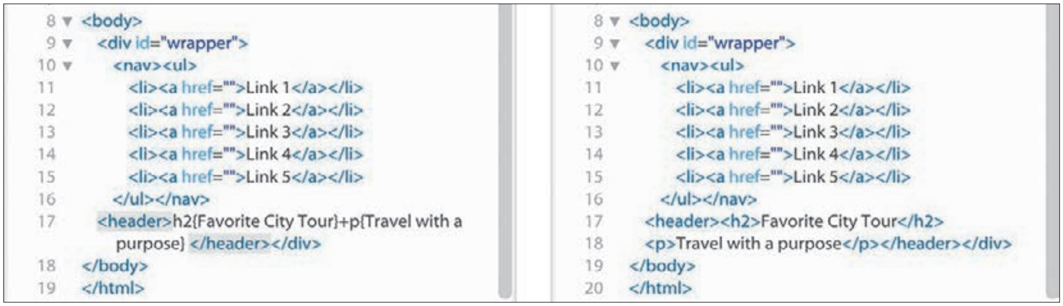
Press Enter/Return to create a new line.

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.



```
<body>
  <div id="wrapper">
    <nav><ul>
      <li><a href="">Link 1</a></li>
      <li><a href="">Link 2</a></li>
      <li><a href="">Link 3</a></li>
      <li><a href="">Link 4</a></li>
      <li><a href="">Link 5</a></li>
    </ul></nav>
  <header>h2{Favorite City Tour}+p{Travel with a
   purpose} </header></div>
</body>
</html>
```

```
<body>
  <div id="wrapper">
    <nav><ul>
      <li><a href="">Link 1</a></li>
      <li><a href="">Link 2</a></li>
      <li><a href="">Link 3</a></li>
      <li><a href="">Link 4</a></li>
      <li><a href="">Link 5</a></li>
    </ul></nav>
  <header><h2>Favorite City Tour</h2>
  <p>Travel with a purpose</p></header></div>
</body>
</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.

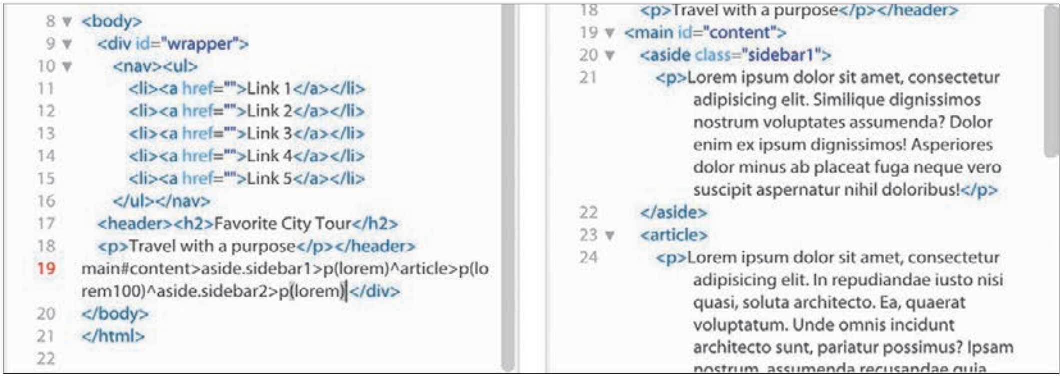
14 Insert the cursor after the closing `</header>` tag.

● **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.

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.

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



```
<body>
  <div id="wrapper">
    <nav><ul>
      <li><a href="">Link 1</a></li>
      <li><a href="">Link 2</a></li>
      <li><a href="">Link 3</a></li>
      <li><a href="">Link 4</a></li>
      <li><a href="">Link 5</a></li>
    </ul></nav>
  <header><h2>Favorite City Tour</h2>
  <p>Travel with a purpose</p></header>
main#content>aside.sidebar1>p(lorem)^article>p(lo
rem100)^aside.sidebar2>p(lorem)</div>
</body>
</html>
```

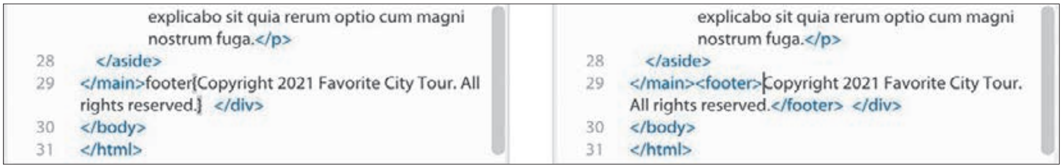
```
  <p>Travel with a purpose</p></header>
<main id="content">
  <aside class="sidebar1">
    <p>Lorem ipsum dolor sit amet, consectetur
       adipisicing elit. Similique dignissimos
       nostrum voluptates assumenda? Dolor
       enim ex ipsum dignissimos! Asperiores
       dolor minus ab placeat fuga neque vero
       suscipit aspernatur nihil doloribus!</p>
  </aside>
  <article>
    <p>Lorem ipsum dolor sit amet, consectetur
       adipisicing 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.



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

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

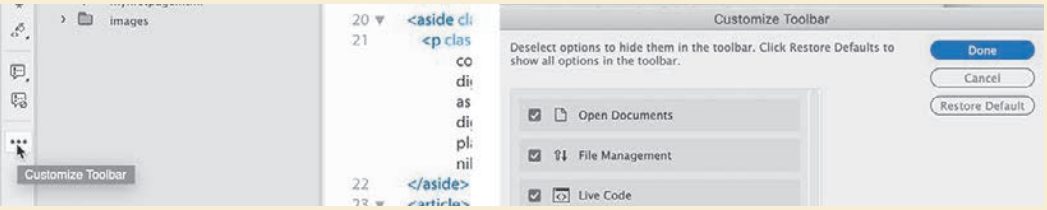
1 If necessary, open **myfirstpage.html** as it appears at the end of the previous exercise.

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.



Deselect options to hide them in the toolbar. Click Restore Defaults to show all options in the toolbar.

☑ Open Documents

☑ File Management

☑ Live Code

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

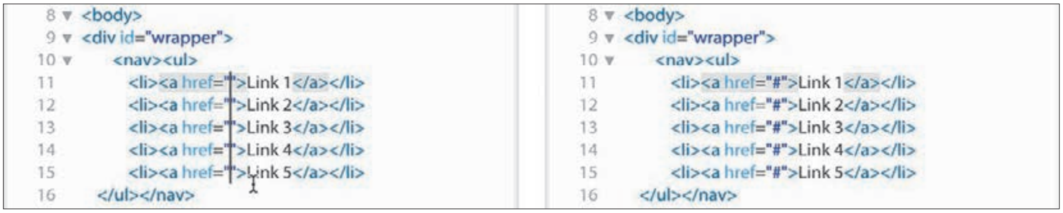
Normally, you would have to add a hash mark (#) to each attribute individually. Multicursor support makes this task much easier, but don't be surprised if it takes you a little practice. Note that all the link attributes are aligned vertically on consecutive lines.

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.



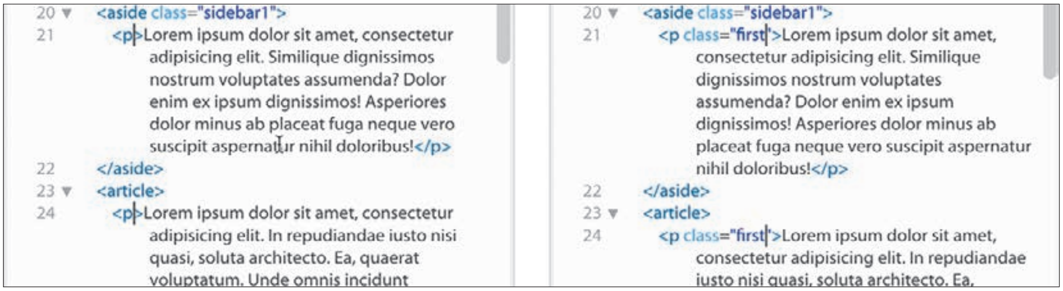
```
<body>
<div id="wrapper">
    <nav><ul>
        <li><a href="">Link 1</a></li>
        <li><a href="">Link 2</a></li>
        <li><a href="">Link 3</a></li>
        <li><a href="">Link 4</a></li>
        <li><a href="">Link 5</a></li>
    </ul></nav>
```

```
<body>
<div id="wrapper">
    <nav><ul>
        <li><a href="#">Link 1</a></li>
        <li><a href="#">Link 2</a></li>
        <li><a href="#">Link 3</a></li>
        <li><a href="#">Link 4</a></li>
        <li><a href="#">Link 5</a></li>
    </ul></nav>
```

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"`**



```
<aside class="sidebar1">
    <p>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>
</aside>
<article>
    <p>Lorem ipsum dolor sit amet, consectetur
        adipisicing elit. In repudiandae iusto nisi
        quasi, soluta architecto. Ea, quaerat
        voluptatum. Unde omnis incidunt
```

```
<aside class="sidebar1">
    <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>
</aside>
<article>
    <p class="first">Lorem ipsum dolor sit amet,
        consectetur adipisicing elit. In repudiandae
        iusto nisi quasi, soluta architecto. Ea,
```

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.

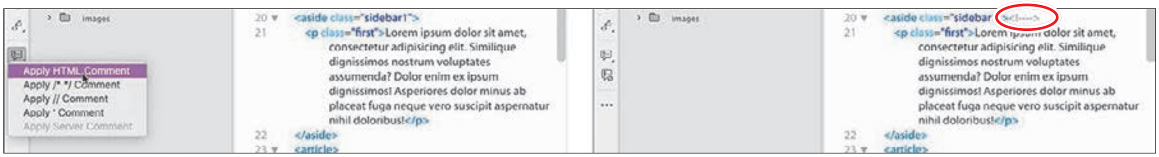
Commenting your code

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 .



```
<aside class="sidebar1">
  <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>
</aside>
<article>
```

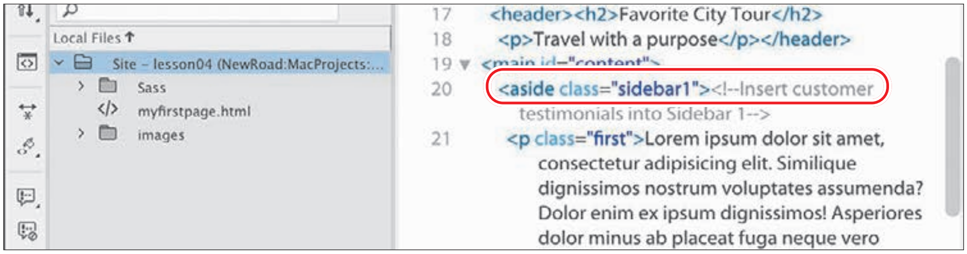
```
<aside class="sidebar<!---->
  <p class="first">Lorem ipsum dolor sit amet,
     consectetur adipisicing elit. Similique
     dignissimos nostrum voluptates
     assumenda? Dolor enim ex ipsum
     dignissimos! Asperiores dolor minus ab
     placeat fuga neque vero suscipit aspernatur
     nihil doloribus!</p>
</aside>
<article>
```

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`**



```
<header><h2>Favorite City Tour</h2>
 <p>Travel with a purpose</p></header>
<main id="content">
 <aside class="sidebar1"><!--Insert customer
   testimonials into Sidebar 1-->
  <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
```

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

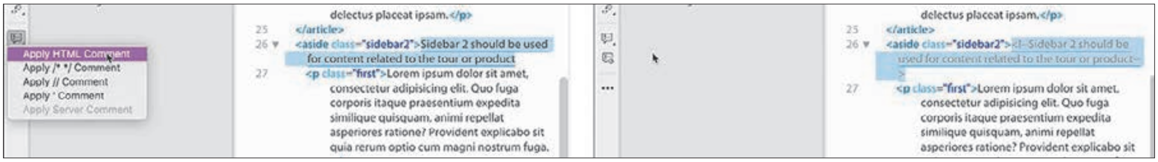
6 Insert the cursor after the opening tag `<aside class="sidebar2">`.

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.



```
        delectus placeat ipsam.</p>
    </article>
    <aside class="sidebar2">Sidebar 2 should be used
      for content related to the tour or product
      <p class="first">Lorem ipsum dolor sit amet,
        consectetur adipisicing elit. Quo fuga
        corporis itaque praesentium expedita
        similique quisquam, animi repellat
        asperiores ratione? Provident explicabo sit
        quia rerum optio cum magni nostrum fuga.
```

```
        delectus placeat ipsam.</p>
    </article>
    <aside class="sidebar2"><!--Sidebar 2 should be
      used for content related to the tour or product--
      >
      <p class="first">Lorem ipsum dolor sit amet,
        consectetur adipisicing elit. Quo fuga
        corporis itaque praesentium expedita
        similique quisquam, animi repellat
        asperiores ratione? Provident explicabo sit
```

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.

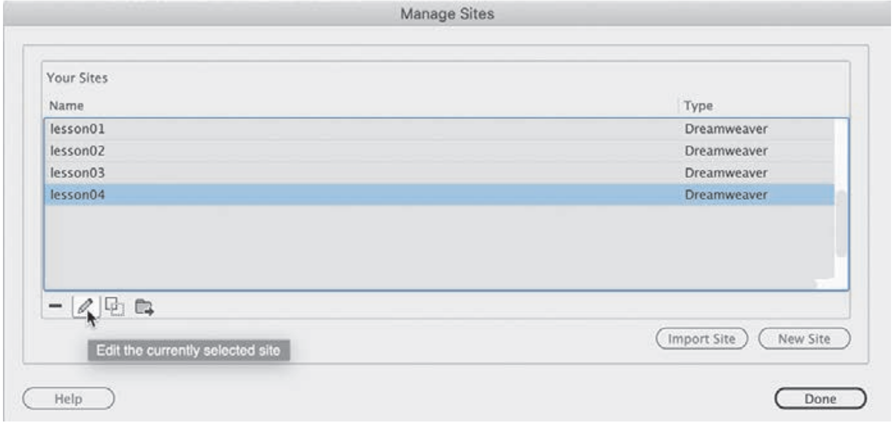
In this exercise, you'll get a taste of how easy it is use preprocessors with Dreamweaver as well as what advantages they offer compared to a regular CSS workflow.

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.


Manage Sites

Your Sites

Name	Type
lesson01	Dreamweaver
lesson02	Dreamweaver
lesson03	Dreamweaver
lesson04	Dreamweaver

Edit the currently selected site

The Site Definition dialog for lesson04 appears.

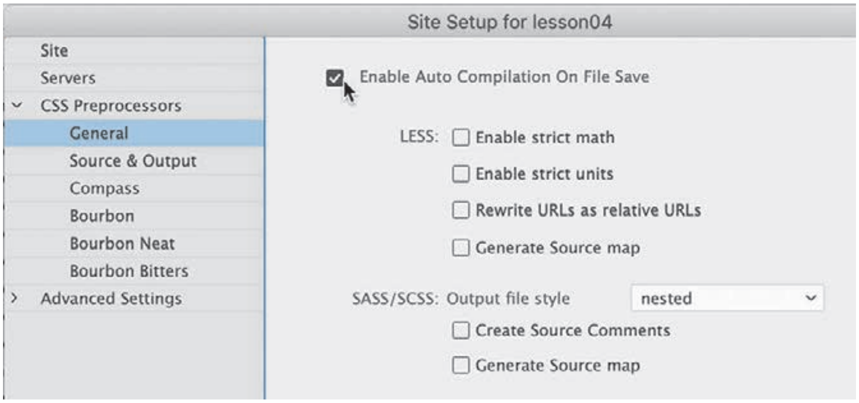
3 Select the **CSS Preprocessors** option in the Site Definition dialog.

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 preprocessor compiler, if necessary.


Site Setup for lesson04

☑ Enable Auto Compilation On File Save

LESS: ☐ Enable strict math
☐ Enable strict units
☐ Rewrite URLs as relative URLs
☐ Generate Source map

SASS/SCSS: Output file style: nested
☐ Create Source Comments
☐ Generate Source map

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:

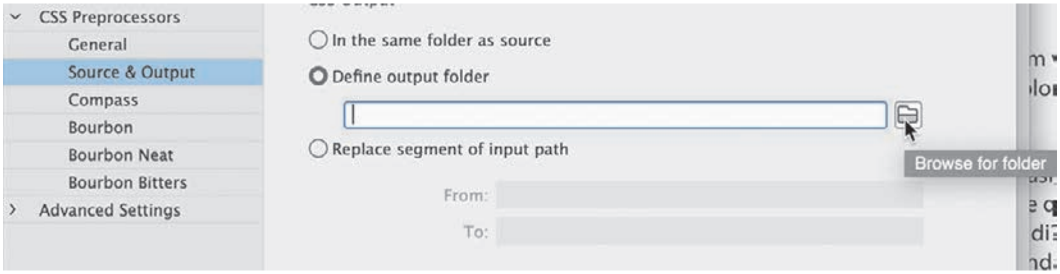
- blog.udemy.com/less-vs-sass/
- css-tricks.com/sass-vs-less/
- keycdn.com/blog/sass-vs-less

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.



⌄ CSS Preprocessors
General
Source & Output
Compass
Bourbon
Bourbon Neat
Bourbon Bitters
› Advanced Settings

○ In the same folder as source
● Define output folder
○ Replace segment of input path
From:
To:

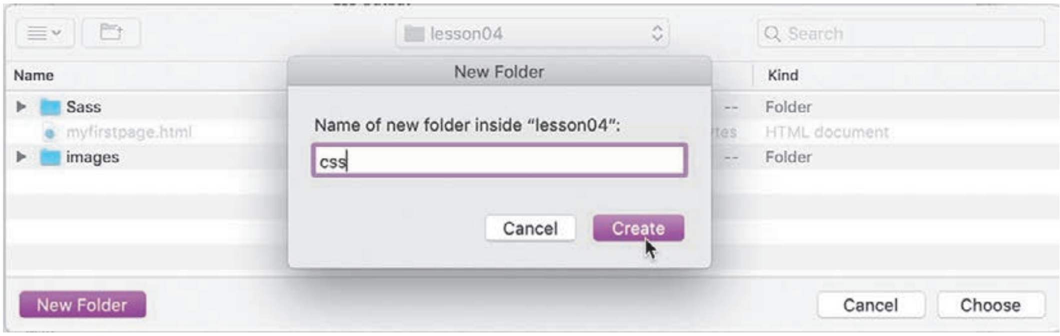
8 Click the Browse For Folder icon 📁.

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.



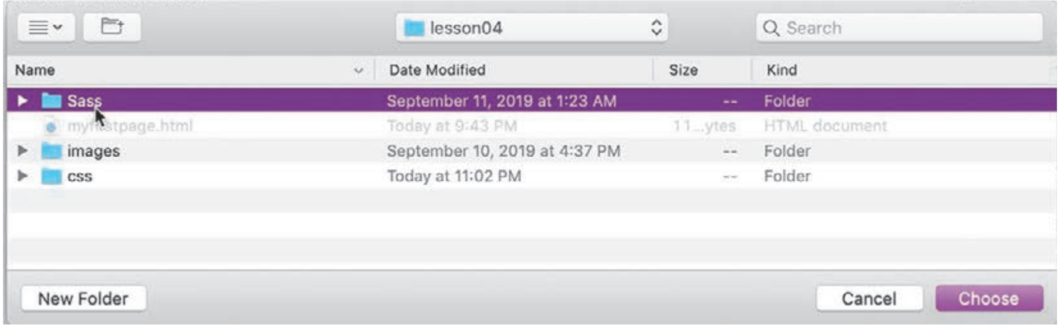
Name of new folder inside “lesson04”:

css

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.



Name	Date Modified	Size	Kind
Sass	September 11, 2019 at 1:23 AM	--	Folder
myfirstpage.html	Today at 9:43 PM	11...ytes	HTML document
images	September 10, 2019 at 4:37 PM	--	Folder
css	Today at 11:02 PM	--	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.

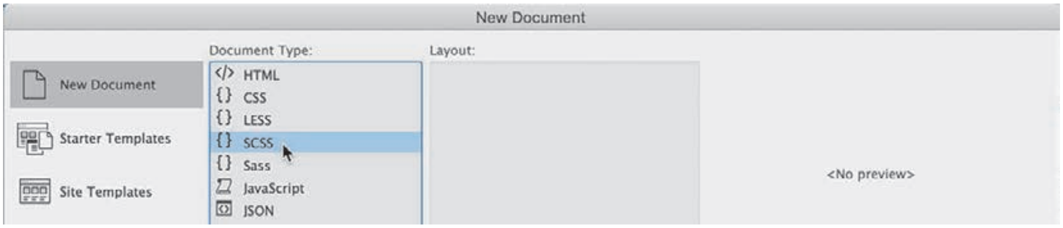
3 If necessary, open **myfirstpage.html** and switch to Split view.

The webpage is unstyled at the moment.

4 Choose File > New.

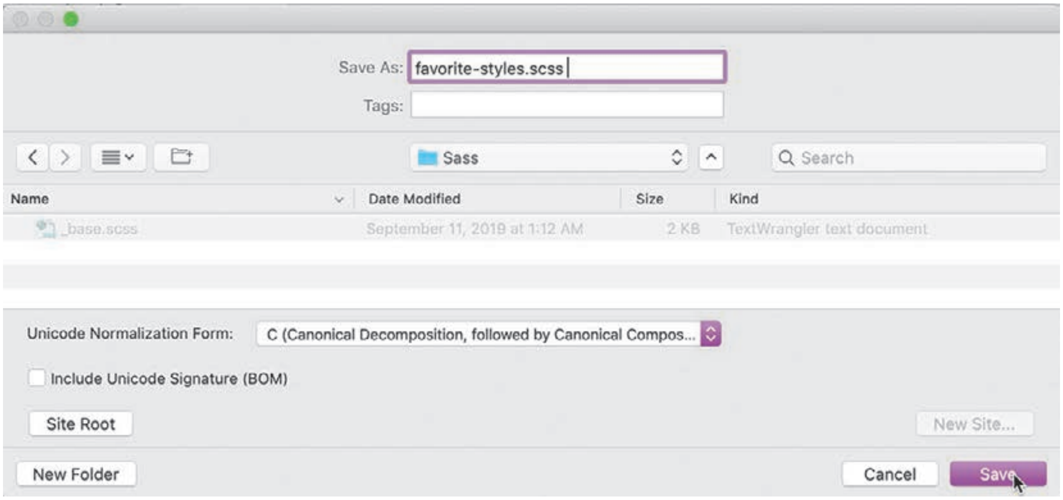
The New Document dialog appears. This dialog allows you to create all types of web-compatible documents. In the Document Type section of the dialog, you will see the LESS, Sass, and SCSS file types.

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.


Save As: favorite-styles.scss

Tags:

Name	Date Modified	Size	Kind
_base.scss	September 11, 2019 at 1:12 AM	2 KB	TextWrangler text document

Unicode Normalization Form: C (Canonical Decomposition, followed by Canonical Compos...

□ Include Unicode Signature (BOM)

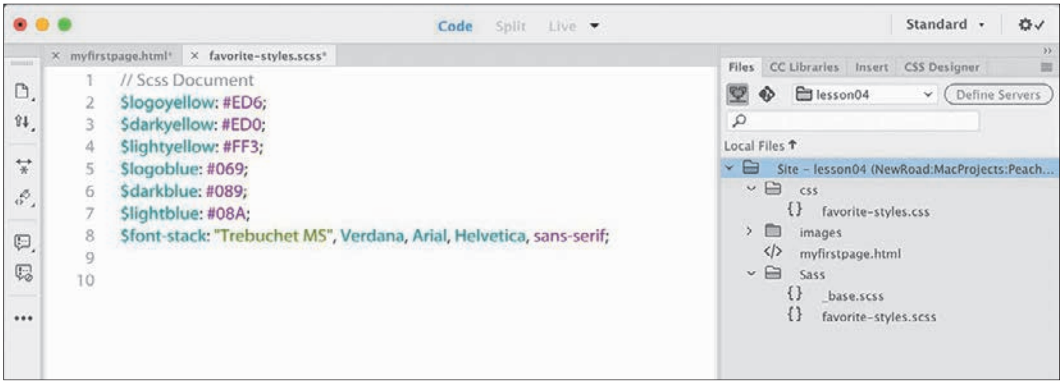
There's no need to create the CSS file; the compiler in Dreamweaver will do that for you. You're all set to start working with Sass. The first step is to define variables. Variables are programmatic constructs that enable you to store CSS specifications you want to use multiple times, such as colors 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 **`$darkyellow: #ED0;`**
`$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.



```
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
10
```

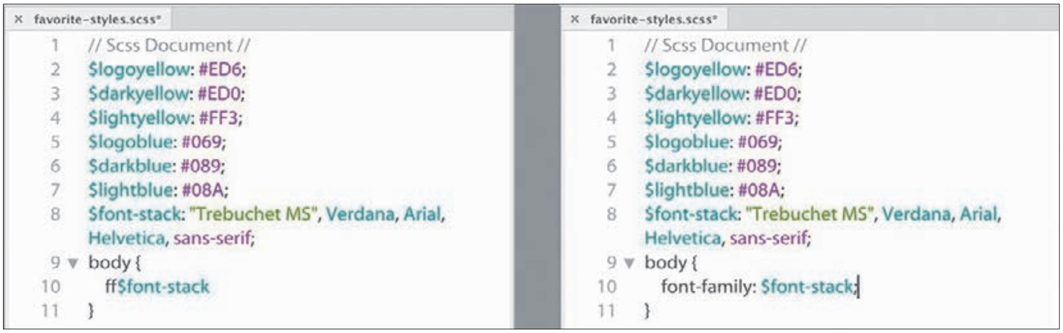
Entering the variables on separate lines makes them easier to read and edit but does not affect how they perform. Just make sure you add a semicolon (;) at the end of each variable.

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.

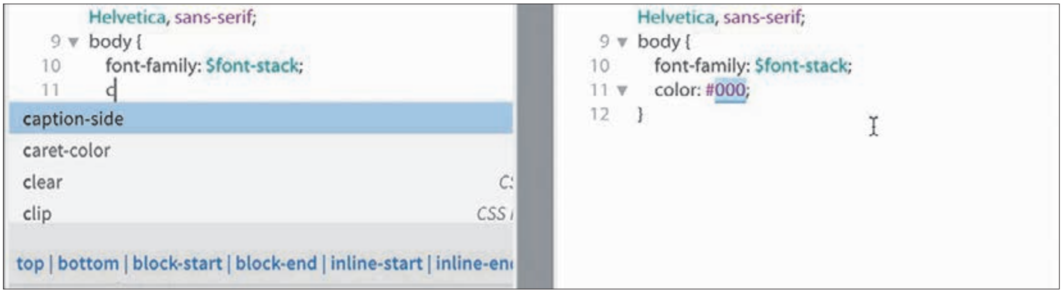


```
// Scss Document //
$logoyellow: #ED6;
$darkyellow: #ED0;
$lightyellow: #FF3;
$logoblue: #069;
$darkblue: #089;
$lightblue: #08A;
$font-stack: "Trebuchet MS", Verdana, Arial,
Helvetica, sans-serif;
body {
   ff$font-stack
}
```

```
// Scss Document //
$logoyellow: #ED6;
$darkyellow: #ED0;
$lightyellow: #FF3;
$logoblue: #069;
$darkblue: #089;
$lightblue: #08A;
$font-stack: "Trebuchet MS", Verdana, Arial,
Helvetica, sans-serif;
body {
   font-family: $font-stack;
}
```

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;
body {
   font-family: $font-stack;
   c
}
```

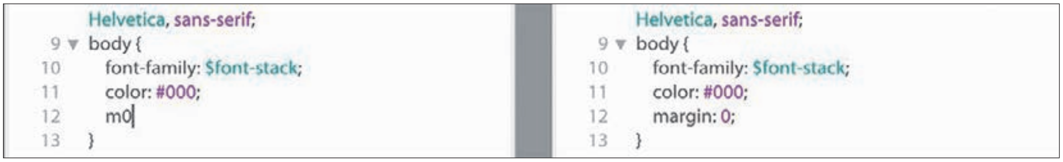
caption-side
caret-color
clear
clip
top | bottom | block-start | block-end | inline-start | inline-en

```
Helvetica, sans-serif;
body {
   font-family: $font-stack;
   color: #000;
}
```

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;
body {
   font-family: $font-stack;
   color: #000;
   m0
}
```

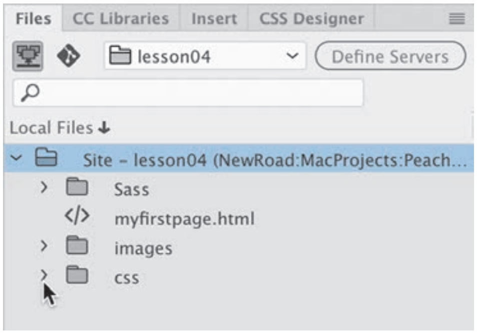
```
Helvetica, sans-serif;
body {
   font-family: $font-stack;
   color: #000;
   margin: 0;
}
```

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.

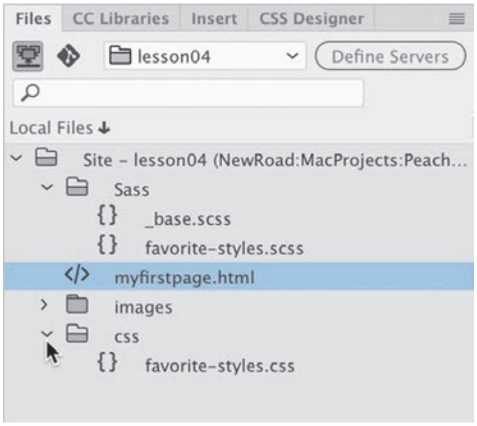
1 Display the Files panel, if necessary, and expand the list of site files.



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

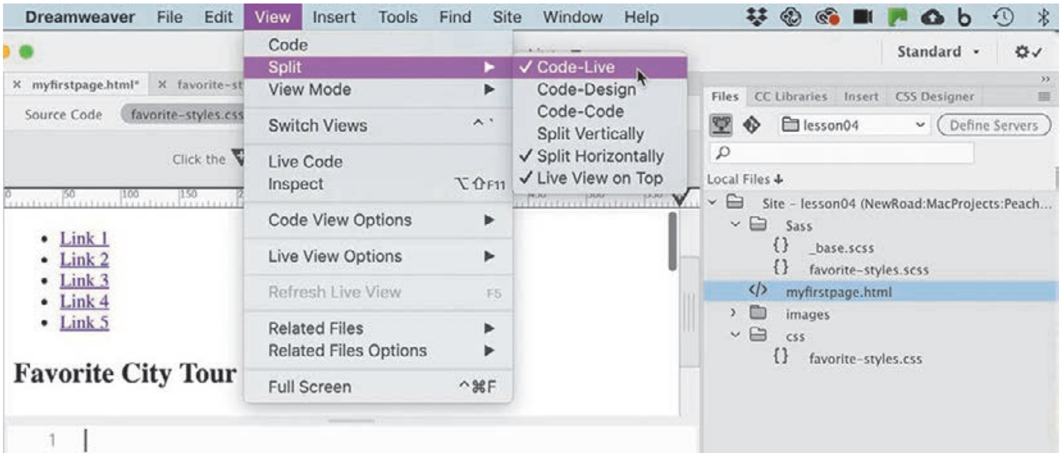
● **Note:** The **favorite-styles.css** file should have been created automatically in the previous exercise when the SCSS file was saved. If you do not see the .css file, you may need to shut down and relaunch Dreamweaver.

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.

3 Select the document tab for **myfirstpage.html**.
Select View > Split > Code-Live.



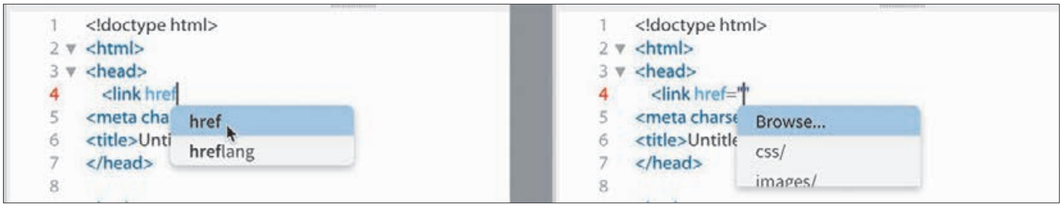
The page shows only default HTML styling.

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.



```
<!doctype html>
<html>
<head>
  <link href
<meta cha
<title>Unti
</head>
```

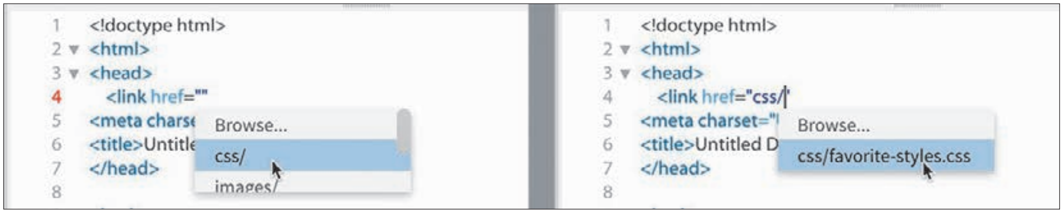
```
<!doctype html>
<html>
<head>
  <link href="
<meta charse
<title>Untitle
</head>
```

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**.

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

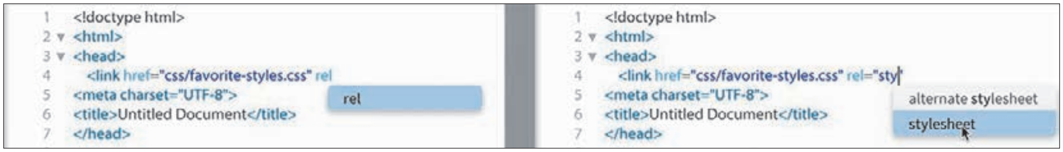


```
<!doctype html>
<html>
<head>
  <link href=""
<meta charse
<title>Untitle
</head>
```

```
<!doctype html>
<html>
<head>
  <link href="css/
<meta charset="
<title>Untitled D
</head>
```

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.

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

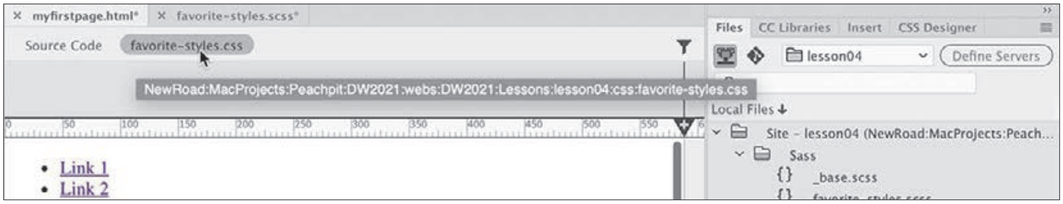


```
<!doctype html>
<html>
<head>
  <link href="css/favorite-styles.css" rel
<meta charset="UTF-8">
<title>Untitled Document</title>
</head>
```

```
<!doctype html>
<html>
<head>
  <link href="css/favorite-styles.css" rel="sty
<meta charset="UTF-8">
<title>Untitled Document</title>
</head>
```

Select **`stylesheet`** from the hinting menu.

10 Move the cursor outside the closing quotation mark. Type **>** to close the link.



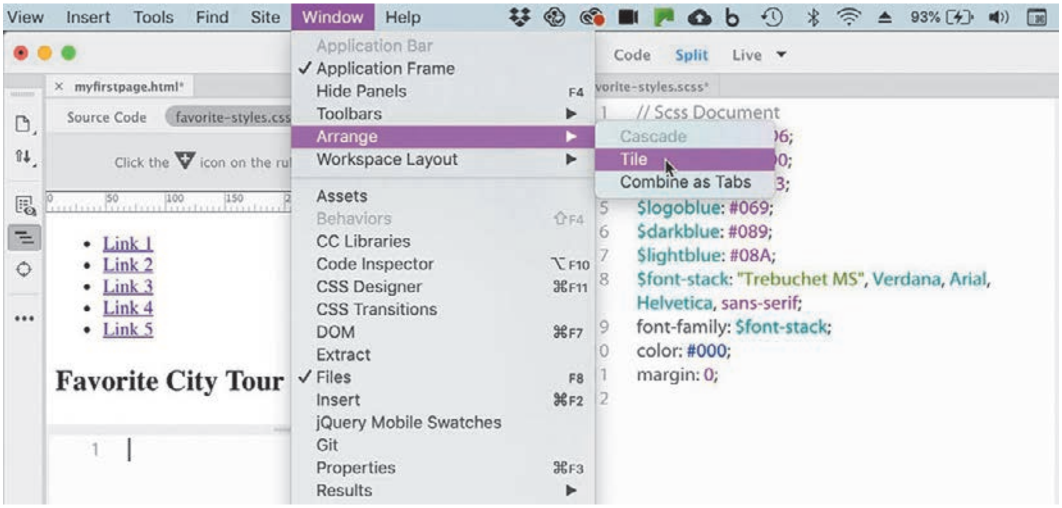
The CSS output file is now referenced by the webpage. In the Live view window, there should be no difference in the styling, but you should now see **favorite-styles.css** displayed in the Related Files interface.

● **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.

11 Select **favorite-styles.css** in the Related Files interface.

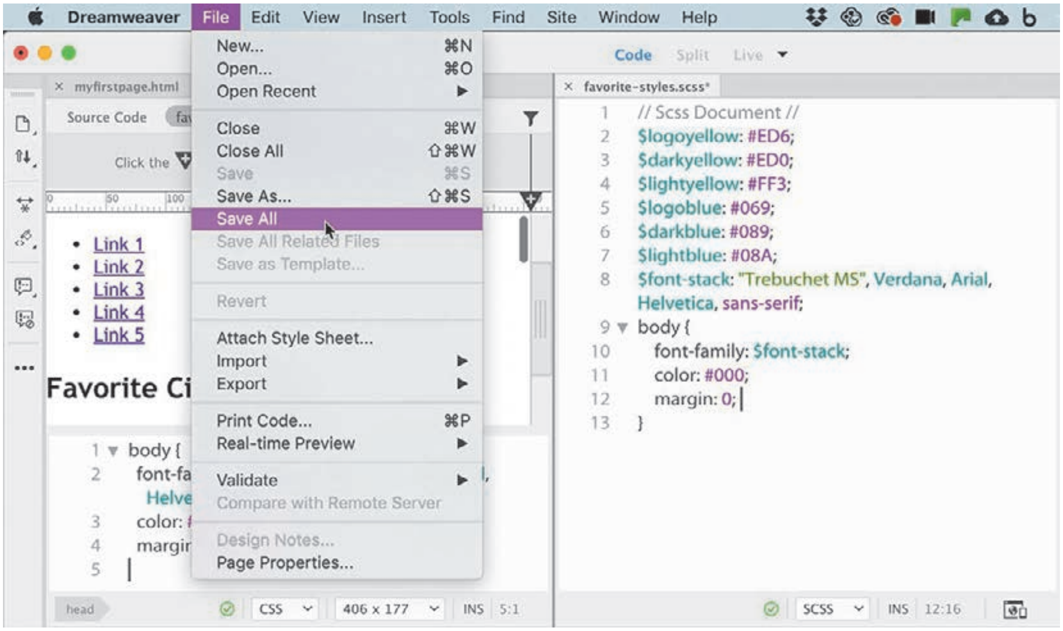
Code view displays the contents of **favorite-styles.css,** which is empty at the moment. An asterisk appears next to the filename in the document tab for **favorite-styles.scss**, indicating that the file has been changed but not saved.

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.



```
// Scss Document //
$logoyellow: #ED6;
$darkyellow: #ED0;
$lightyellow: #FF3;
$logoblue: #069;
$darkblue: #089;
$lightblue: #08A;
$font-stack: "Trebuchet MS", Verdana, Arial,
Helvetica, sans-serif;
body {
   font-family: $font-stack;
   color: #000;
   margin: 0;
}
```

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.

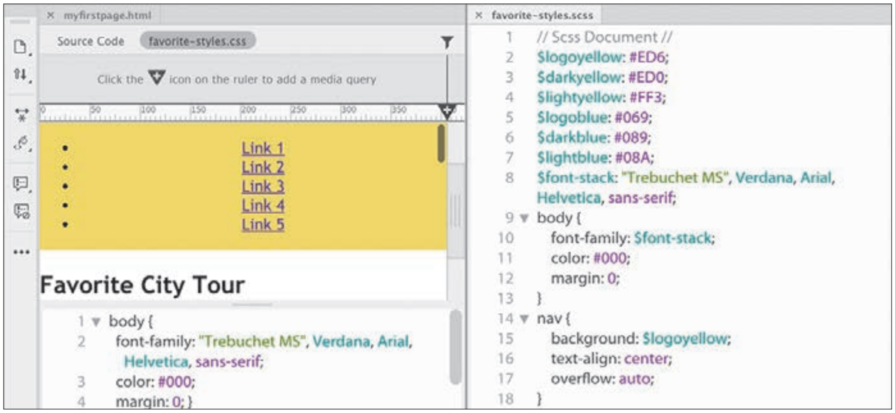
4 Type **`ta:c`** and press Tab. Press Enter/Return.

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.



```
1 ▼ body {
2     font-family: "Trebuchet MS", Verdana, Arial,
       Helvetica, sans-serif;
3     color: #000;
4     margin: 0; }
```

```
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      color: #000;
12      margin: 0;
13    }
14 ▼  nav {
15      background: $logoyellow;
16      text-align: center;
17      overflow: auto;
18    }
```

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

<ul> 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.

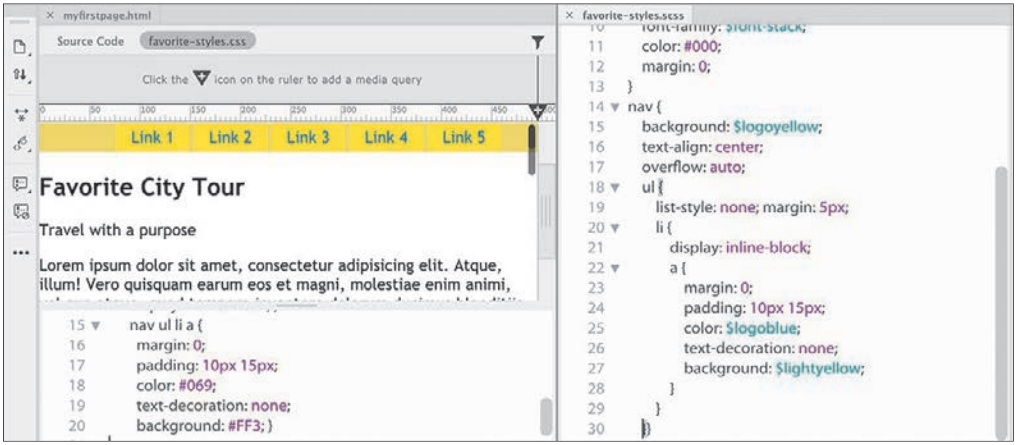
This property will display all the links in a single row, side by side. The last element to style is the <a> for the link itself.

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.



Favorite City Tour

Travel with a purpose

Lorem ipsum dolor sit amet, consectetur adipisicing elit. Atque, illum! Vero quisquam earum eos et magni, molestiae enim animi,

```
nav ul li a {
  margin: 0;
  padding: 10px 15px;
  color: #069;
  text-decoration: none;
  background: #FF3; }
```

```
    font-family: $font-stack;
    color: #000;
    margin: 0;
  }
  nav {
    background: $logoyellow;
    text-align: center;
    overflow: auto;
    ul {
      list-style: none; margin: 5px;
      li {
        display: inline-block;
        a {
          margin: 0;
          padding: 10px 15px;
          color: $logoblue;
          text-decoration: none;
          background: $lightyellow;
        }
      }
    }
```

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.

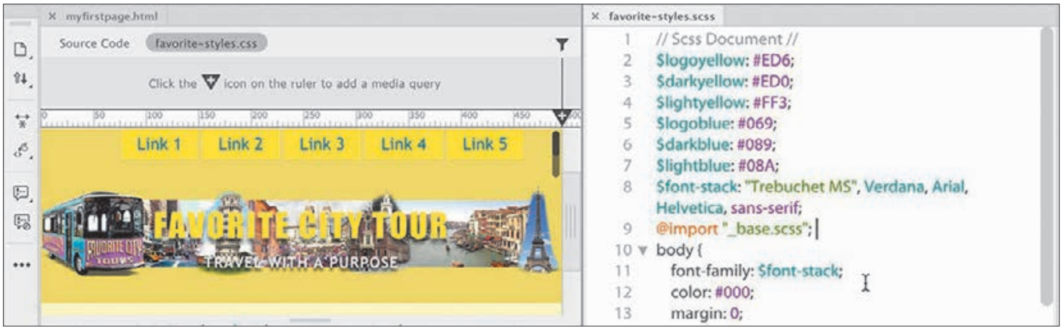
2 In **myfirstpage.html,** click **favoritestyles.css** in the Related Files interface.

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**.



```
// Scss Document //
$logoyellow: #ED6;
$darkyellow: #ED0;
$lightyellow: #FF3;
$logoblue: #069;
$darkblue: #089;
$lightblue: #08A;
$font-stack: "Trebuchet MS", Verdana, Arial,
Helvetica, sans-serif;
@import "_base.scss";
body {
  font-family: $font-stack;
  color: #000;
  margin: 0;
```

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 preprocessor. 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.

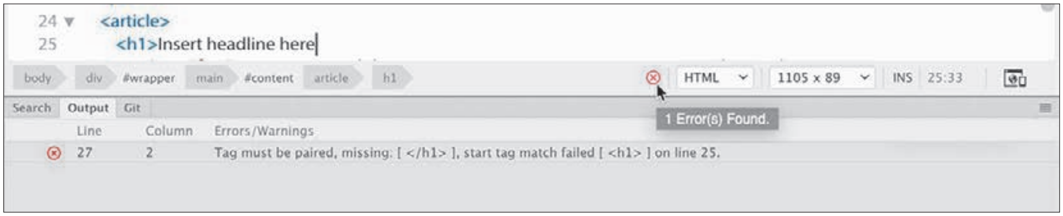
● **Note:** Dreamweaver may create the opening and closing tags at once. If so, delete the closing `</h1>` tag before proceeding to step 4.

3 Type **`<h1>Insert headline here`**

4 Save the file.

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.



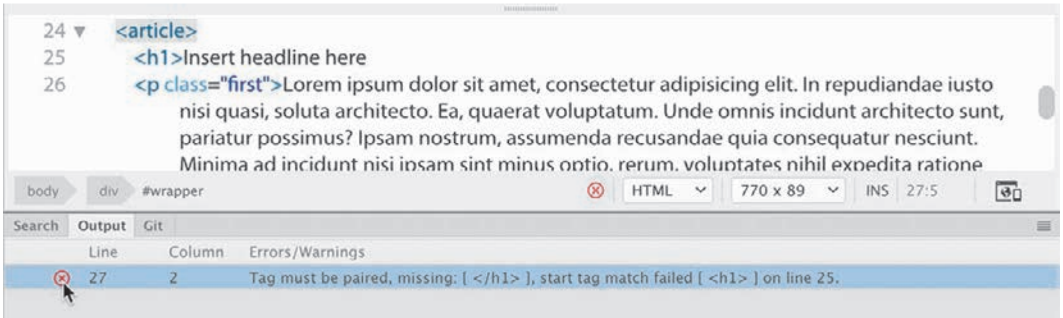
```
24 ▼  <article>
25      <h1>Insert headline here
```

Line	Column	Errors/Warnings
⊗ 27	2	Tag must be paired, missing: [</h1>], start tag match failed [<h1>] on line 25.

● **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.



```
24 ▼  <article>
25      <h1>Insert headline here
26      <p class="first">Lorem ipsum dolor sit amet, consectetur adipisicing elit. In repudiandae iusto
           nisi quasi, soluta architecto. Ea, quaerat voluptatum. Unde omnis incidunt architecto sunt,
           pariatur possimus? Ipsam nostrum, assumenda recusandae quia consequatur nesciunt.
           Minima ad incidunt nisi ipsam sint minus optio, rerum, voluptates nihil expedita ratione
```

Line	Column	Errors/Warnings
⊗ 27	2	Tag must be paired, missing: [</h1>], start tag match failed [<h1>] on line 25.

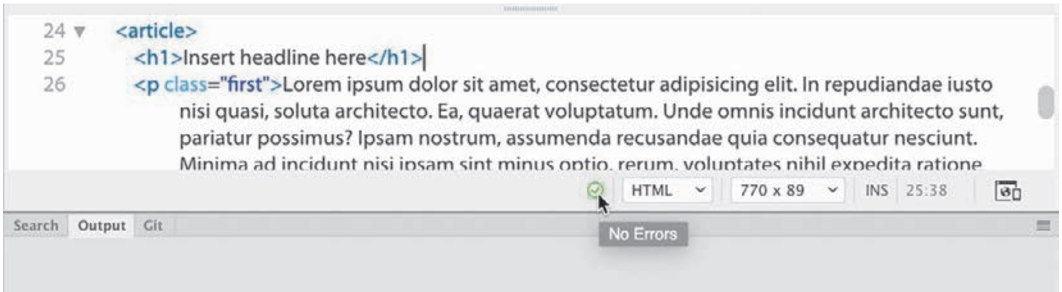
Dreamweaver focuses on the article in the Code view window that it identifies as containing the error. Since Dreamweaver is looking for the closing tag for the `<h1>` element, the first closing tag it encounters is `</article>` and flags it, which is incorrect. This behavior will get you close to the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



```
<article>
  <h1>Insert headline here</h1>
  <p class="first">Lorem ipsum dolor sit amet, consectetur adipisicing elit. In repudiandae iusto
      nisi quasi, soluta architecto. Ea, quaerat voluptatum. Unde omnis incidunt architecto sunt,
      pariatur possimus? Ipsam nostrum, assumenda recusandae quia consequatur nesciunt.
      Minima ad incidunt nisi ipsam sint minus optio, rerum, voluptates nihil expedita ratione
```

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).

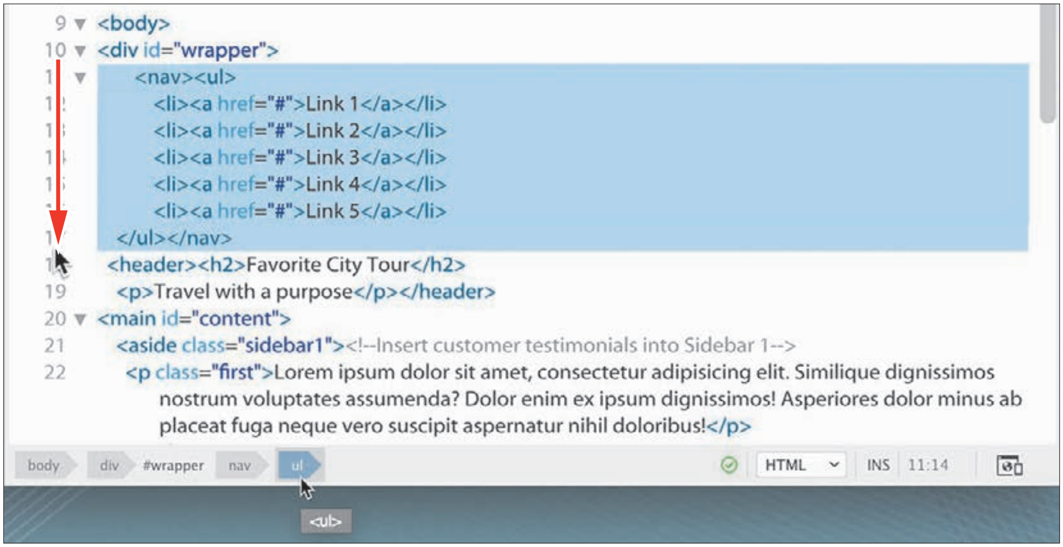
3 Drag the cursor across the entire element, including the menu items.

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.



```
<body>
<div id="wrapper">
    <nav><ul>
      <li><a href="#">Link 1</a></li>
      <li><a href="#">Link 2</a></li>
      <li><a href="#">Link 3</a></li>
      <li><a href="#">Link 4</a></li>
      <li><a href="#">Link 5</a></li>
  </ul></nav>
 <header><h2>Favorite City Tour</h2>
  <p>Travel with a purpose</p></header>
<main id="content">
  <aside class="sidebar1"><!--Insert customer testimonials into Sidebar 1-->
   <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>
```

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.

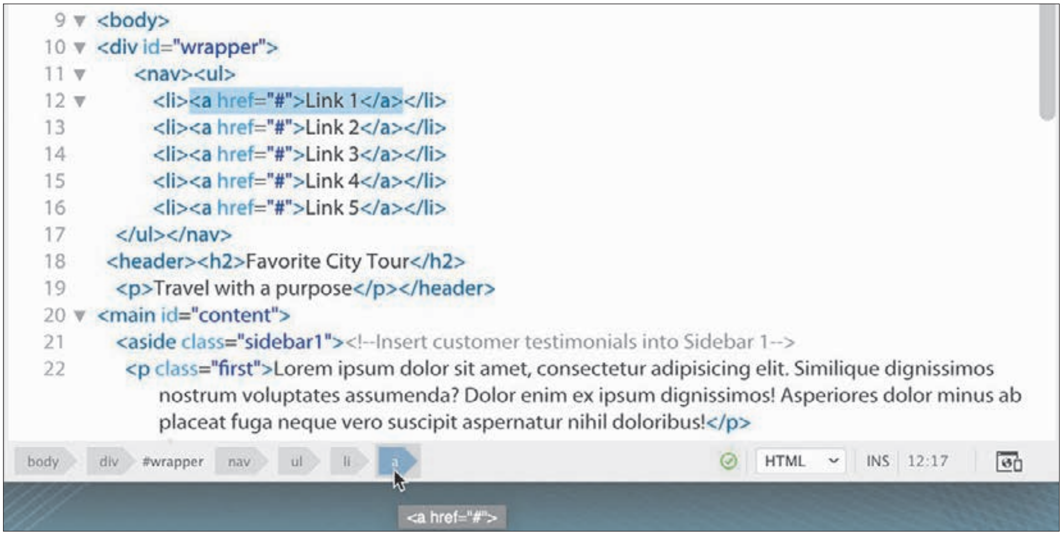
1 Scroll down and locate the following code:

```
<a href="#">Link 1</a>
```

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.

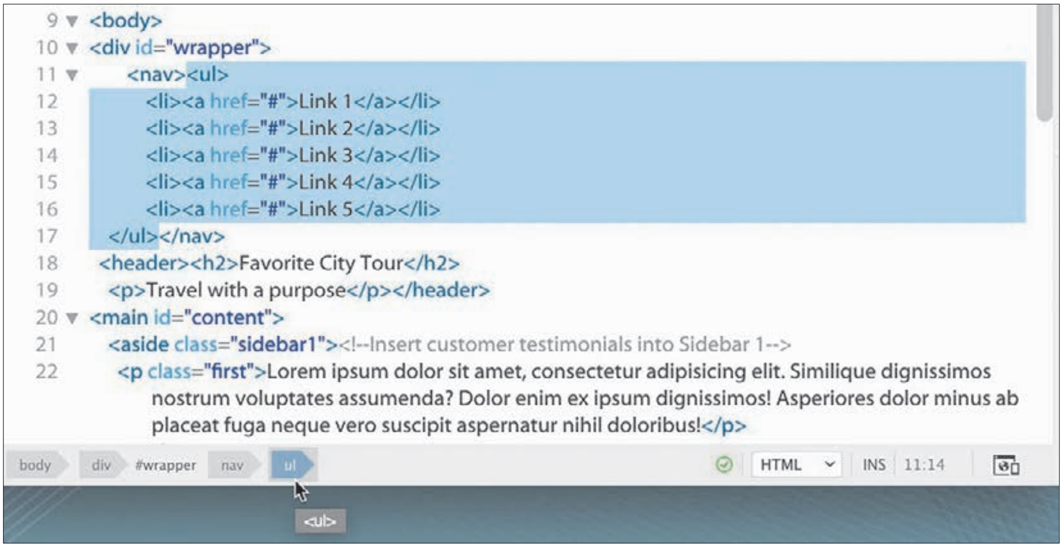


```
<body>
<div id="wrapper">
    <nav><ul>
      <li><a href="#">Link 1</a></li>
      <li><a href="#">Link 2</a></li>
      <li><a href="#">Link 3</a></li>
      <li><a href="#">Link 4</a></li>
      <li><a href="#">Link 5</a></li>
  </ul></nav>
 <header><h2>Favorite City Tour</h2>
  <p>Travel with a purpose</p></header>
<main id="content">
  <aside class="sidebar1"><!--Insert customer testimonials into Sidebar 1-->
   <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 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 `<li>` element, which is a child of `<ul>`, 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 `<ul>` tag selector.



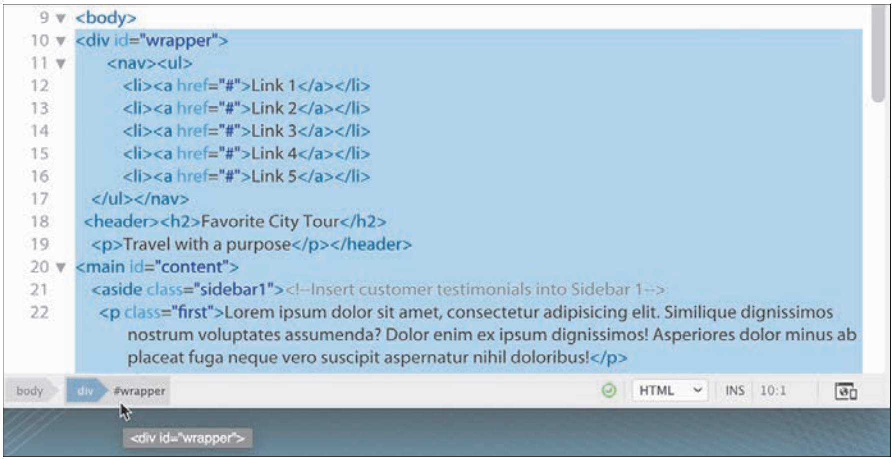
```
<body>
<div id="wrapper">
    <nav><ul>
      <li><a href="#">Link 1</a></li>
      <li><a href="#">Link 2</a></li>
      <li><a href="#">Link 3</a></li>
      <li><a href="#">Link 4</a></li>
      <li><a href="#">Link 5</a></li>
  </ul></nav>
 <header><h2>Favorite City Tour</h2>
  <p>Travel with a purpose</p></header>
<main id="content">
  <aside class="sidebar1"><!--Insert customer testimonials into Sidebar 1-->
   <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 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.



```
<body>
<div id="wrapper">
    <nav><ul>
      <li><a href="#">Link 1</a></li>
      <li><a href="#">Link 2</a></li>
      <li><a href="#">Link 3</a></li>
      <li><a href="#">Link 4</a></li>
      <li><a href="#">Link 5</a></li>
  </ul></nav>
 <header><h2>Favorite City Tour</h2>
  <p>Travel with a purpose</p></header>
<main id="content">
  <aside class="sidebar1"><!--Insert customer testimonials into Sidebar 1-->
   <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 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

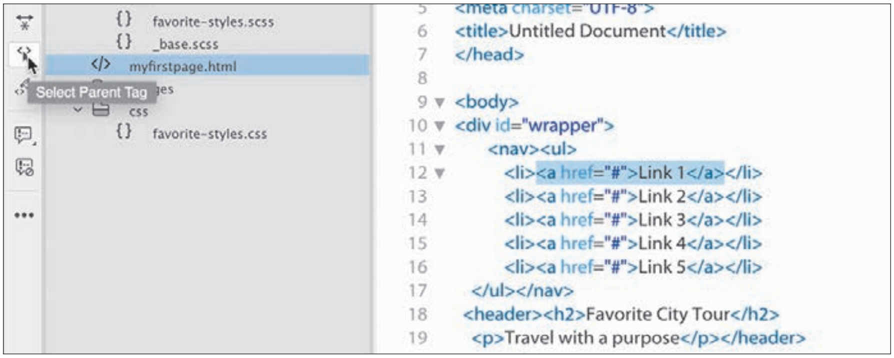
Using the parent tag selector in the Code view window makes the job of selecting the hierarchical structure of your page even simpler.

1 Choose Window > Toolbars > Common to display the Common toolbar, if necessary.

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.

2 Insert the cursor anywhere in the text `Link 1`.

3 In the Common toolbar, click the Select Parent Tag icon .



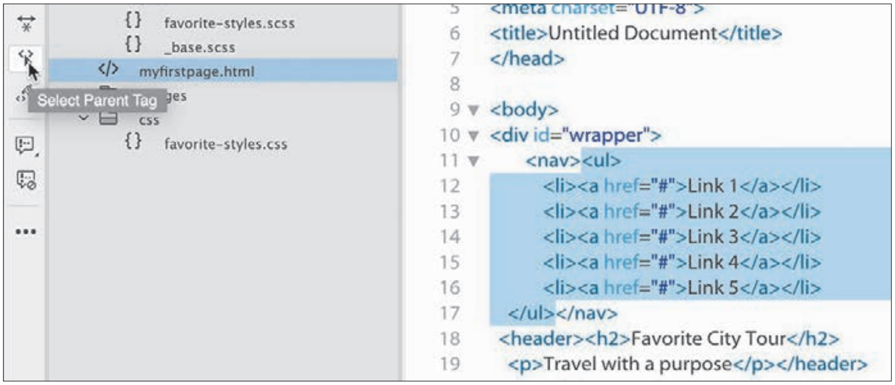
```
<meta charset="UTF-8">
<title>Untitled Document</title>
</head>

<body>
<div id="wrapper">
    <nav><ul>
      <li><a href="#">Link 1</a></li>
      <li><a href="#">Link 2</a></li>
      <li><a href="#">Link 3</a></li>
      <li><a href="#">Link 4</a></li>
      <li><a href="#">Link 5</a></li>
  </ul></nav>
 <header><h2>Favorite City Tour</h2>
  <p>Travel with a purpose</p></header>
```

The entire `<a>` element is highlighted.

4 Click the Select Parent Tag icon again or press Ctrl+[/Cmd+[(left bracket).

The entire `<li>` element is selected.

5 Click the Select Parent Tag icon.



```
<meta charset="UTF-8">
<title>Untitled Document</title>
</head>

<body>
<div id="wrapper">
    <nav><ul>
        <li><a href="#">Link 1</a></li>
        <li><a href="#">Link 2</a></li>
        <li><a href="#">Link 3</a></li>
        <li><a href="#">Link 4</a></li>
        <li><a href="#">Link 5</a></li>
  </ul></nav>
 <header><h2>Favorite City Tour</h2>
  <p>Travel with a purpose</p></header>
```

The entire `<ul>` element is selected.

6 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.

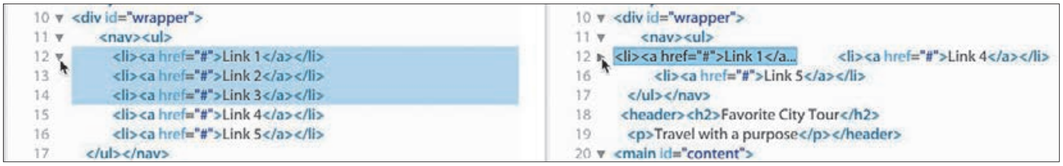
Collapsing code

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.

1 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.

2 Click the Collapse icon ▼ to collapse the selection.



```
<div id="wrapper">
    <nav><ul>
      <li><a href="#">Link 1</a></li>
      <li><a href="#">Link 2</a></li>
      <li><a href="#">Link 3</a></li>
      <li><a href="#">Link 4</a></li>
      <li><a href="#">Link 5</a></li>
  </ul></nav>
```

```
<div id="wrapper">
    <nav><ul>
<li><a href="#">Link 1</a...        <li><a href="#">Link 4</a></li>
      <li><a href="#">Link 5</a></li>
  </ul></nav>
 <header><h2>Favorite City Tour</h2>
  <p>Travel with a purpose</p></header>
<main id="content">
```

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

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

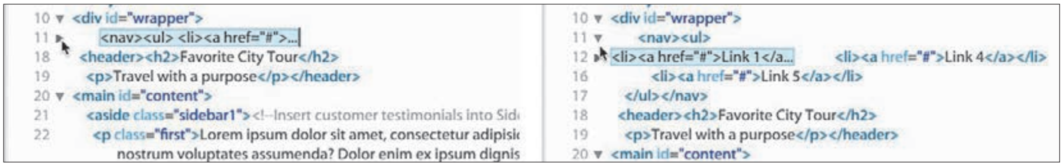
3 Click the Collapse icon ▼ beside the line for the `<nav>` element.

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.



```
<div id="wrapper">
    <nav><ul> <li><a href="#">...
  <header><h2>Favorite City Tour</h2>
    <p>Travel with a purpose</p></header>
<main id="content">
    <aside class="sidebar1"><!--Insert customer testimonials into Sid
      <p class="first">Lorem ipsum dolor sit amet, consectetur adipisi
          nostrum voluptates assumenda? Dolor enim ex ipsum dignis
```

```
<div id="wrapper">
      <nav><ul>
<li><a href="#">Link 1</a...        <li><a href="#">Link 4</a></li>
          <li><a href="#">Link 5</a></li>
    </ul></nav>
  <header><h2>Favorite City Tour</h2>
    <p>Travel with a purpose</p></header>
<main id="content">
```

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

2 Click the Expand icon ▶ beside the line for the `<li>` elements.

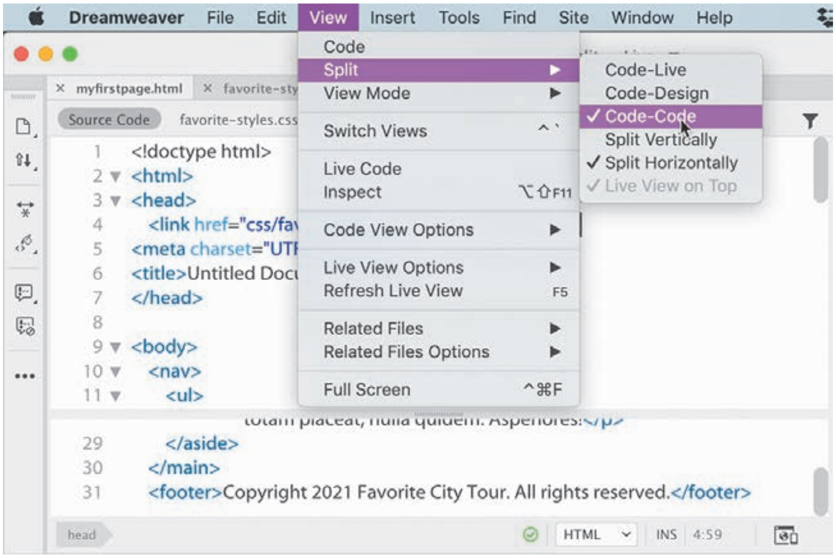
All collapsed elements are now expanded. Note that the Expand icon for the three `<li>` elements disappears altogether.

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.



```
<!doctype html>
<html>
<head>
  <link href="css/fa
<meta charset="UTF
<title>Untitled Doc
</head>

<body>
  <nav>
    <ul>
```

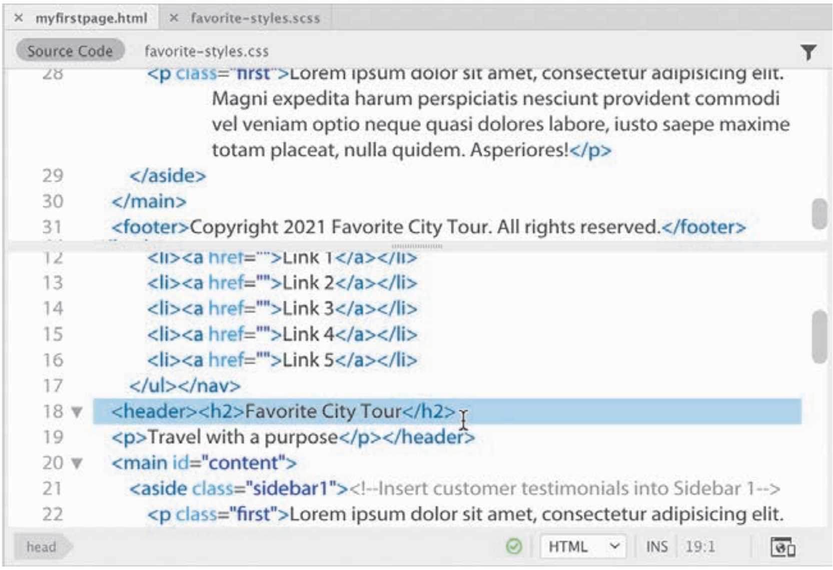
```
      totam placeat, nulla quidem. Asperiores!</p>
      </aside>
    </main>
    <footer>Copyright 2021 Favorite City Tour. All rights reserved.</footer>
```

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.



```
      <p class="first">Lorem ipsum dolor sit amet, consectetur adipisicing elit.
          Magni expedita harum perspiciatis nesciunt provident commodi
          vel veniam optio neque quasi dolores labore, iusto saepe maxime
          totam placeat, nulla quidem. Asperiores!</p>
    </aside>
  </main>
  <footer>Copyright 2021 Favorite City Tour. All rights reserved.</footer>
```

```
      <li><a href="">Link 1</a></li>
      <li><a href="">Link 2</a></li>
      <li><a href="">Link 3</a></li>
      <li><a href="">Link 4</a></li>
      <li><a href="">Link 5</a></li>
    </ul></nav>
  <header><h2>Favorite City Tour</h2>
  <p>Travel with a purpose</p></header>
  <main id="content">
    <aside class="sidebar1"><!--Insert customer testimonials into Sidebar 1-->
      <p class="first">Lorem ipsum dolor sit amet, consectetur adipisicing elit.
```

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**.

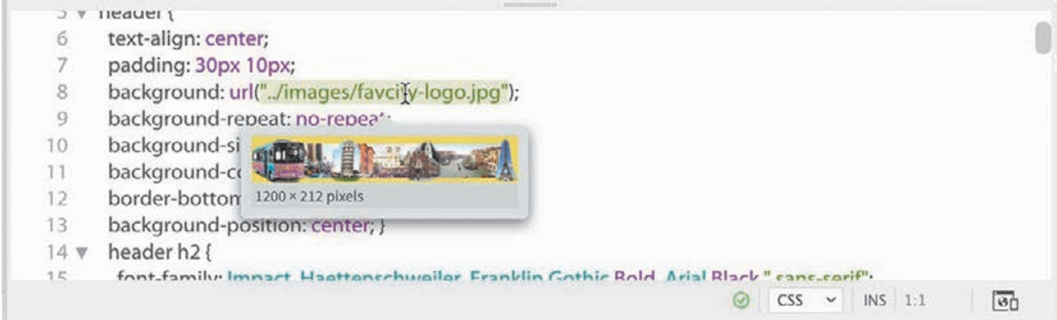
2 Click **favoritestyles.css** in the Related Files interface.

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.

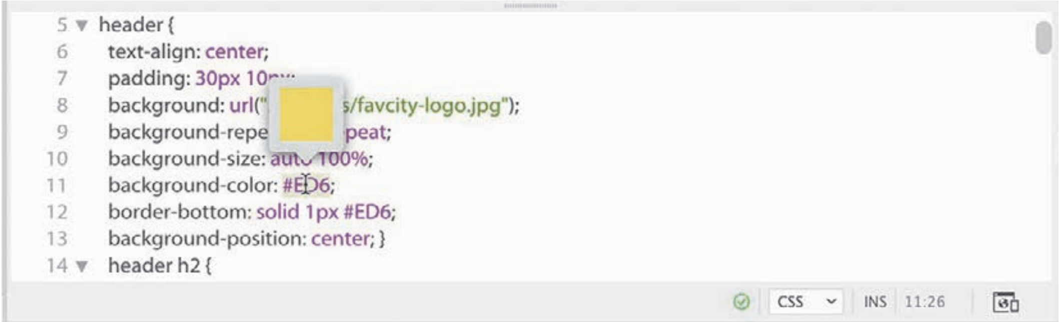
4 Position the cursor over the markup `url(../images/favcity-logo.jpg)` in the `background` property (line 8).



```
5  header {
6    text-align: center;
7    padding: 30px 10px;
8    background: url("../images/favcity-logo.jpg");
9    background-repeat: no-repea
10   background-si
11   background-c
12   border-botton
13   background-position: center; }
14   header h2 {
15     font-family: Impact, Haettenschweiler, Franklin Gothic Bold, Arial Black, "sans-serif";
```

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.



```
header {
  text-align: center;
  padding: 30px 10p
  background: url("          s/favcity-logo.jpg");
  background-repe            peat;
  background-size: auto 100%;
  background-color: #ED6;
  border-bottom: solid 1px #ED6;
  background-position: center; }
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