Dw

Adobe Dreamweaver 2022 release

Adobe

Classroom in a Book®

The official training workbook from Adobe

Jim Maivald

CONTENTS

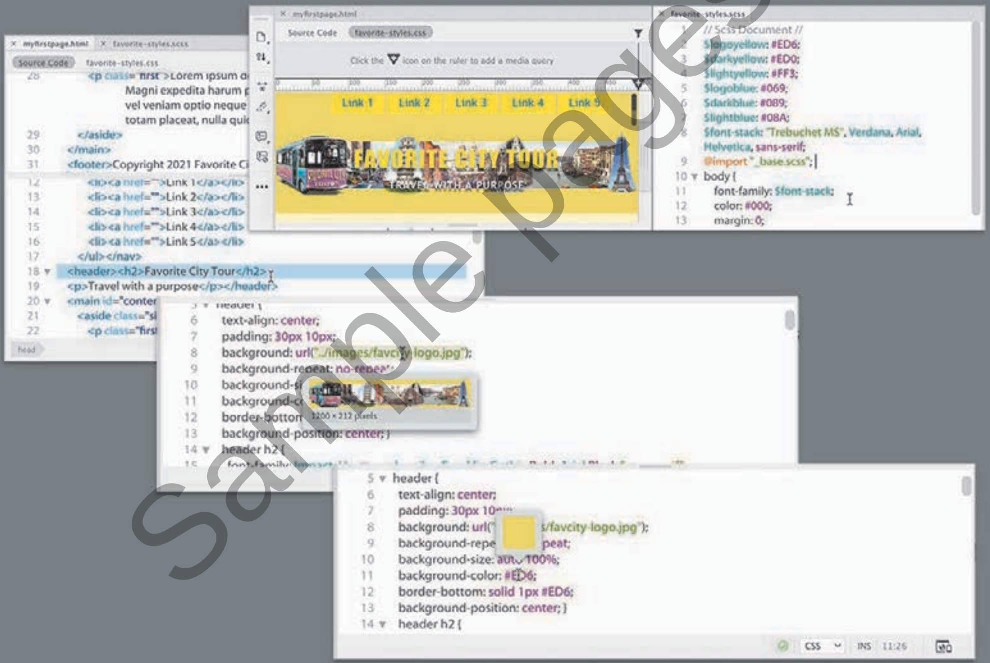
4 WORKING WITH CODE

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 adobepress.com/DreamweaverCIB2022. Define a new site for the lesson04 folder, as described in the "Getting Started" section at the beginning of this book. Name the new site **lesson04**.



```
// 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;
```

```
header {
    text-align: center;
    padding: 30px 10px;
    background: url("../images/favcity-logo.jpg");
    background-repeat: no-repeat;
    background-size: auto 100%;
    background-color: #ED6;
    border-bottom: solid 1px #ED6;
    background-position: center; }
header h2 {
```

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

Creating HTML code

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

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

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

Dreamweaver has always made it as easy to work with a page in Code view as it is in Design view or Live view. Dreamweaver can unify your entire web development team by providing a single platform that can handle almost any task.

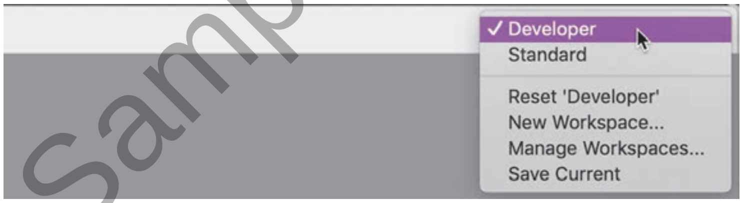
You'll often find that a specific task is actually easier to accomplish in Code view than in Live view or Design view alone. In the following exercises, you'll learn more about how Dreamweaver makes working with the code an effortless and surprisingly enjoyable task.

Writing code manually

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

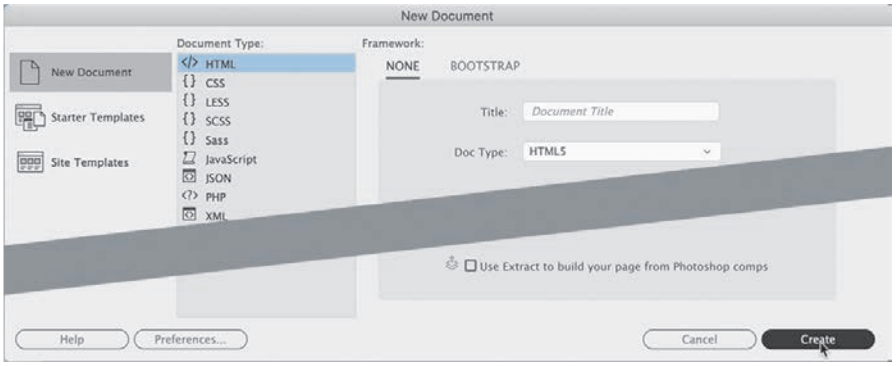
1 Define a site based on the lesson04 folder downloaded from your account page, as described in the "Getting Started" section at the beginning of the book.

2 Select Developer from the Workspace menu.



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

3 Choose File > New.



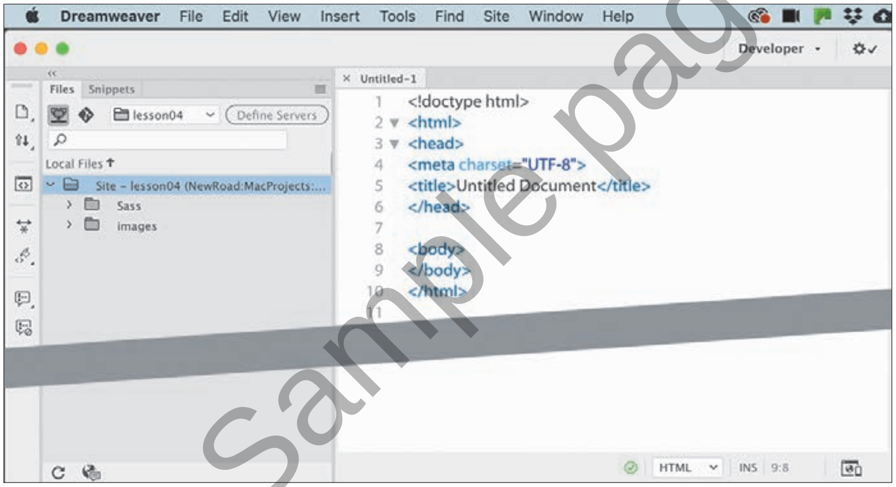
Title: Document Title

Doc Type: HTML5

The New Document dialog appears.

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

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



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

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

Dreamweaver creates the basic structure of a webpage automatically. The cursor will normally appear at the beginning of the code when you are using the Developer workspace.

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.

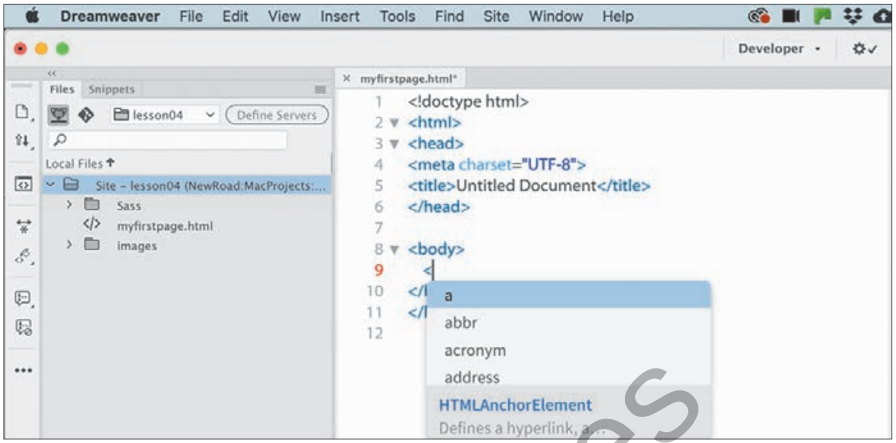
5 Choose File > Save.

6 Name the file **myfirstpage.html** and save it in the lesson04 folder.

● **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.

7 Insert the cursor after the opening `<body>` tag.

Press Enter/Return to create a new line. Type **<**



Dreamweaver File Edit View Insert Tools Find Site Window Help
Developer
Files Snippets
lesson04
Define Servers
Local Files
Site - lesson04 (NewRoad:MacProjects:...
Sass
myfirstpage.html
images
myfirstpage.html*

```
<!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.

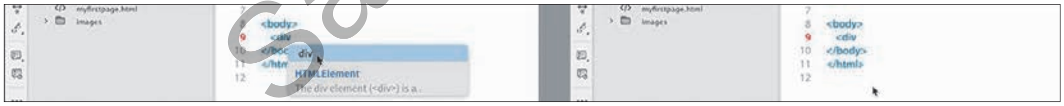
8 Type **d**

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
</body>
</html>
```

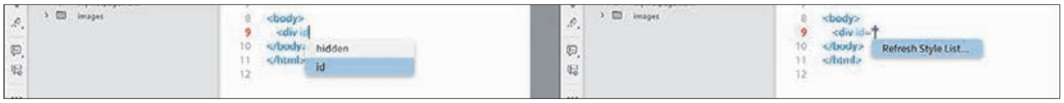
The tag name `div` is inserted in the code. The cursor remains at the end of the tag name, waiting for your next input. For example, you could complete the tag name or enter various HTML attributes. Let's add an `id` attribute to the `div` element.

● **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.

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.


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

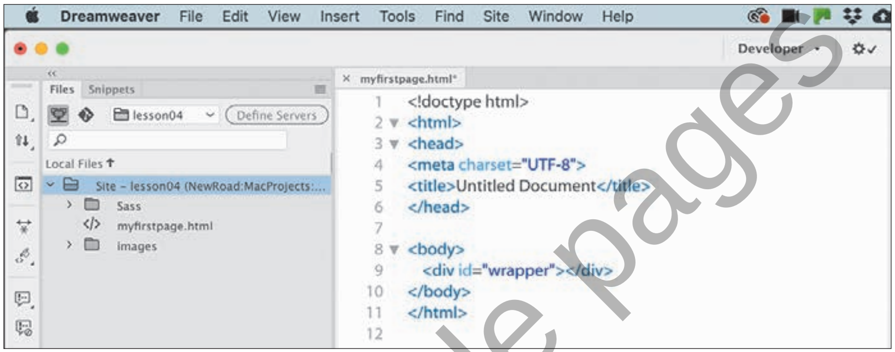
```
<body>
  <div id=""
</body>
</html>
```

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.

14 Type **>**


Dreamweaver File Edit View Insert Tools Find Site Window Help
Developer
myfirstpage.html*
Files Snippets
lesson04
Define Servers
Local Files
Site - lesson04 (NewRoad:MacProjects:...
Sass
myfirstpage.html
images

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

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

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

When you type the >, 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

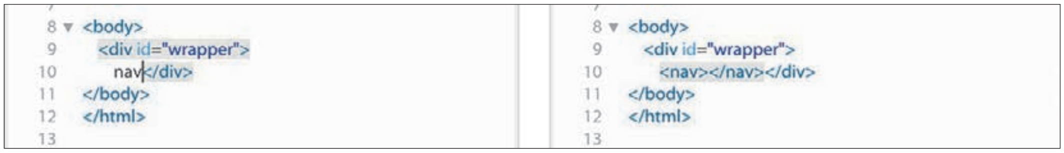
Emmet is a web-developer toolkit that was added to Dreamweaver a while ago and enables you to supercharge your code-writing tasks. When you enter shorthand characters and operators, Emmet enables you to create whole blocks of code with just a few keystrokes. In the following exercise you will experience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

1 If necessary, open **myfirstpage.html**.

2 In the Code view window, 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 most websites a navigation menu appears at the top of the page. HTML5 uses the `<nav>` element as the foundation of site navigation. You will insert the menu and learn how to populate it with menu items.

3 Type **nav** and press Tab.



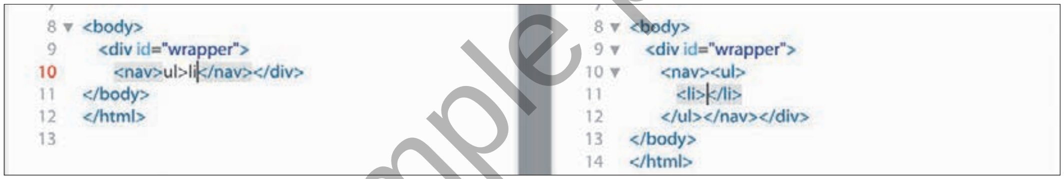
```
<body>
  <div id="wrapper">
    nav</div>
</body>
</html>
```

```
<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>
```

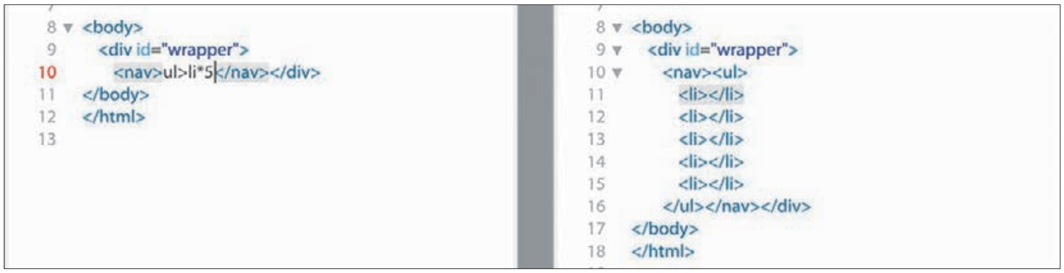
```
<body>
  <div id="wrapper">
    <nav><ul>
      <li></li>
    </ul></nav></div>
</body>
</html>
```

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`**

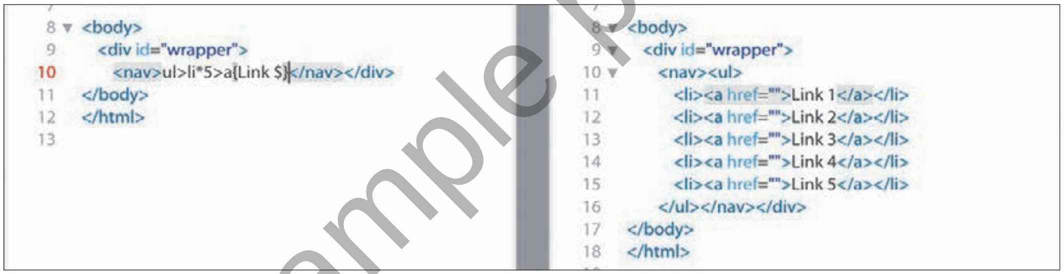
If you guessed that adding the markup `>a` would create a hyperlink child element for each link item, you are correct. Emmet can also create placeholder content. Let's use it to insert some text in each link item.

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>
```

The new menu appears fully structured, with five link items and hyperlink placeholders incremented 1 through 5. The menu is nearly complete. The only things 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.

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

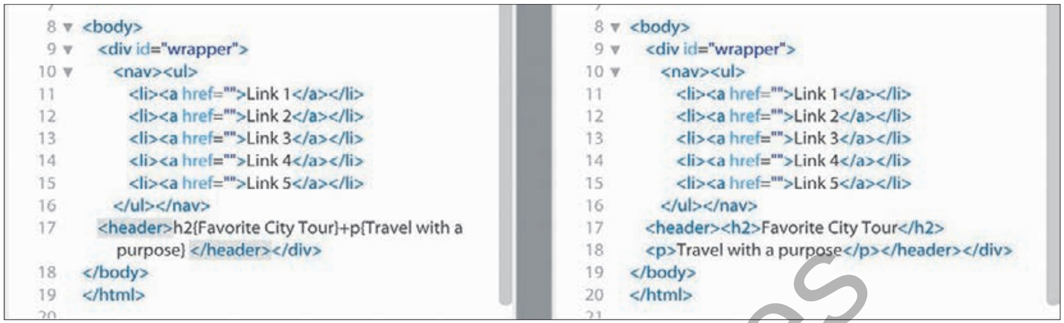
11 Insert the cursor after the closing `</nav>` tag.
Press Enter/Return to create a new line.

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.

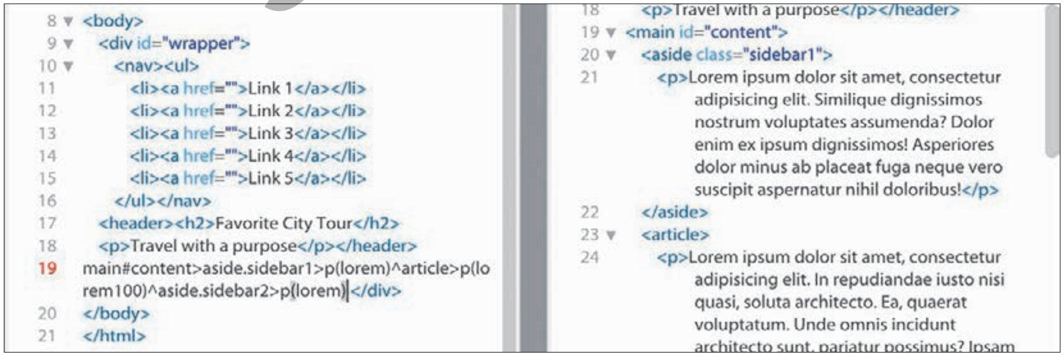
14 Insert the cursor after the closing `</header>` tag.

15 Press Enter/Return to insert a new line.

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

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

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



```
<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>
```

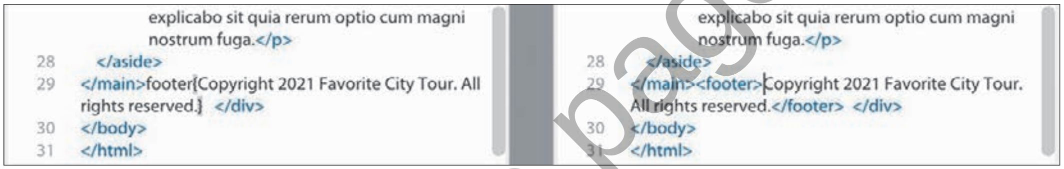
```
  <p>Travel with a purpose</p></header>
<main id="content">
  <aside class="sidebar1">
    <p>Lorem ipsum dolor sit amet, consectetur
      adipisicing elit. Similique dignissimos
      nostrum voluptates assumenda? Dolor
      enim ex ipsum dignissimos! Asperiores
      dolor minus ab placeat fuga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

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 2022 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 https://emmet.io to learn more about Emmet. Check out https://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? Dreamweaver offers 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 `nav`, `header`, `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. In HTML, the hash mark (#) is used as placeholder content until the final link destinations can be added.

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.

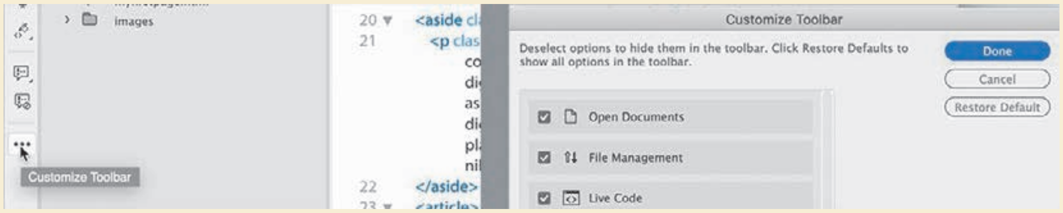
3 Hold the Alt key (Windows) or Option key (macOS) and drag the mouse down through all five links.

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.

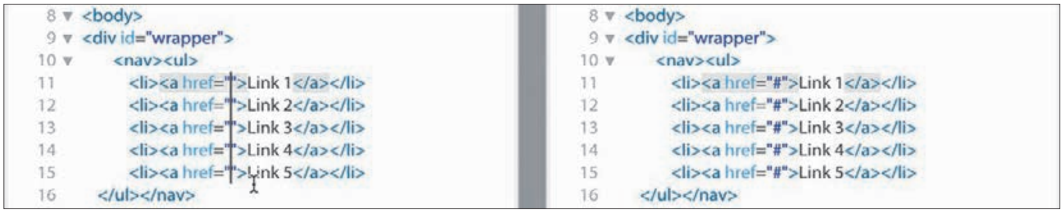
Customizing the Common toolbar

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 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.

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



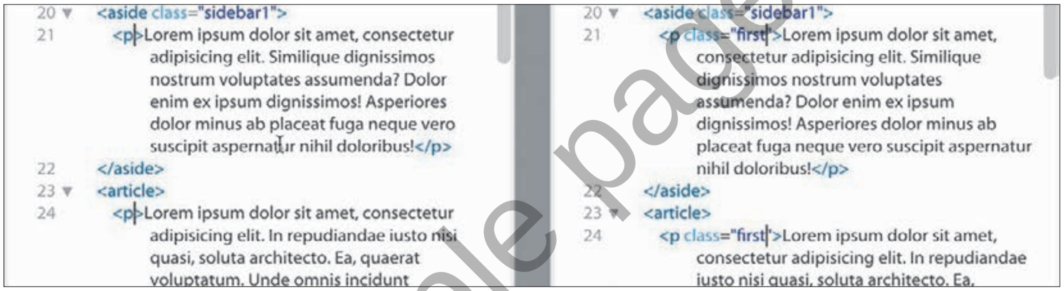
```
<body>
<div id="wrapper">
    <nav><ul>
      <li><a href="">Link 1</a></li>
      <li><a href="">Link 2</a></li>
      <li><a href="">Link 3</a></li>
      <li><a href="">Link 4</a></li>
      <li><a href="">Link 5</a></li>
  </ul></nav>
```

```
<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.

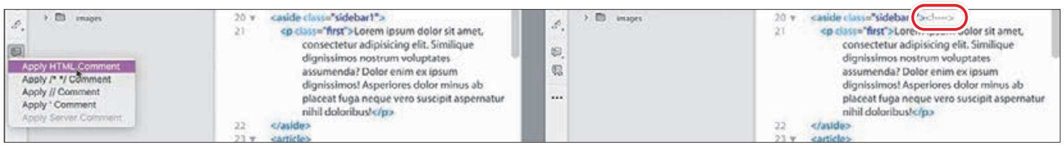
1 Open **myfirstpage.html** using the Developer workspace.

2 Insert the cursor after the opening tag `<aside class="sidebar1">`.

3 Click the Apply Comment icon .

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

4 Choose Apply HTML Comment.

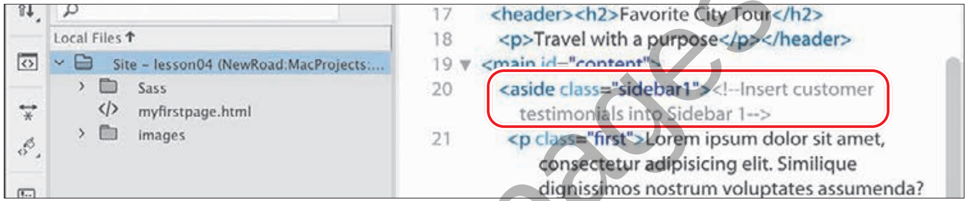


```
<aside class="sidebar1">
  <p class="first">Lorem ipsum dolor sit amet,
      consectetur adipisicing elit. Similique
      dignissimos nostrum voluptates
      assumenda? Dolor enim ex ipsum
      dignissimos! Asperiores dolor minus ab
      placeat fuga neque vero suscipit aspernatur
      nihil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

```
<aside class="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>
```

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?
```

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

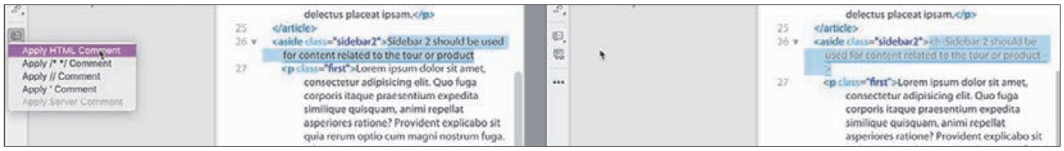
6 Insert the cursor after the opening tag `<aside class="sidebar2">`.

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 supports CSS preprocessors. 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 over the last few years was support for industry-standard CSS preprocessors. Known by the acronyms LESS (Leaner CSS), Sass (Syntactically Awesome Style Sheets), and SCSS (Sassy CSS), these 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.

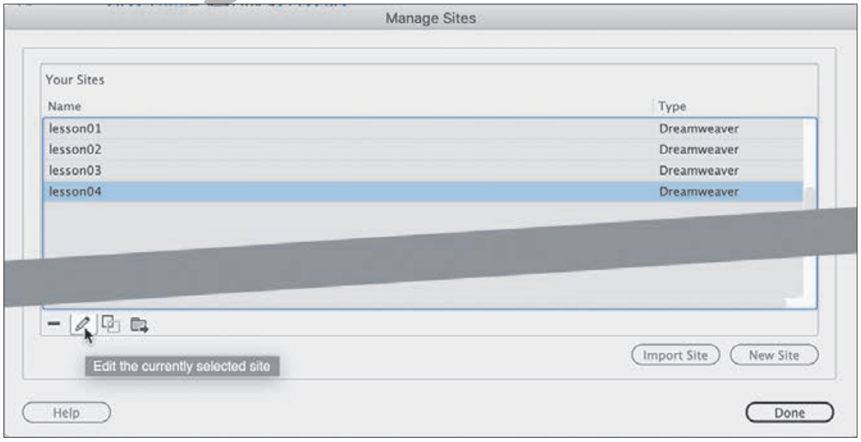
Enabling a preprocessor

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

The Site Definition dialog for lesson04 appears.

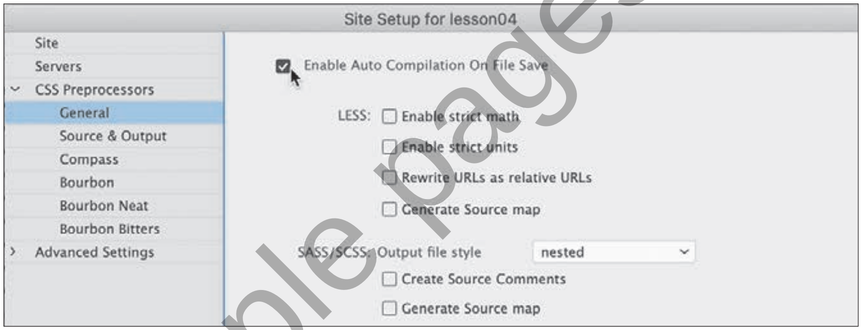
3 Select the **CSS Preprocessors** option in the Site Definition dialog.

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

4 Select the General category.

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

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



Site Setup for lesson04

- Site
- Servers
- CSS Preprocessors
  - General
  - Source & Output
  - Compass
  - Bourbon
  - Bourbon Neat
  - Bourbon Bitters
- Advanced Settings

✓ Enable Auto Compilation On File Save

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

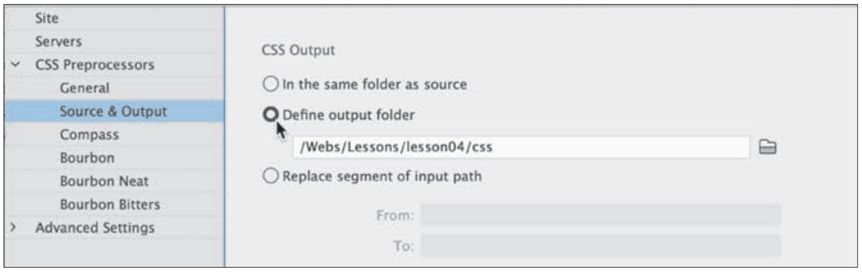
SASS/SCSS: Output file style: nested
☐ Create Source Comments
☐ Generate Source map

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.

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.



- Site
- Servers
- CSS Preprocessors
  - General
  - Source & Output
  - Compass
  - Bourbon
  - Bourbon Neat
  - Bourbon Bitters
- Advanced Settings

CSS Output

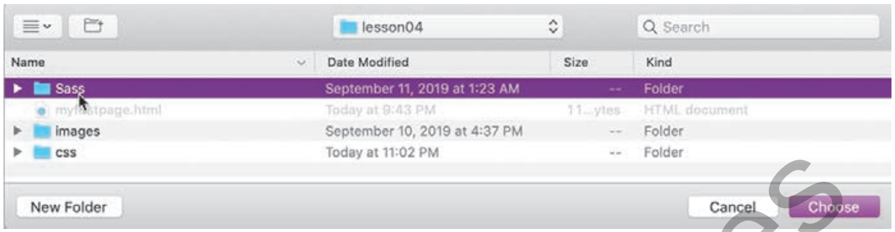
○ In the same folder as source
● Define output folder
/Webs/Lessons/lesson04/css
○ Replace segment of input path
From:
To:

When you enable this option, Dreamweaver displays a file path pointing to a folder (css). This folder doesn't exist yet, but Dreamweaver will create it automatically. If you want to use a different folder, you will have to use the Browse For Folder icon to select or create the folder.

8 Click the Browse For Folder icon beside the Source Folder field.

9 Navigate to the Site Root folder.

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



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

11 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.

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.

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.

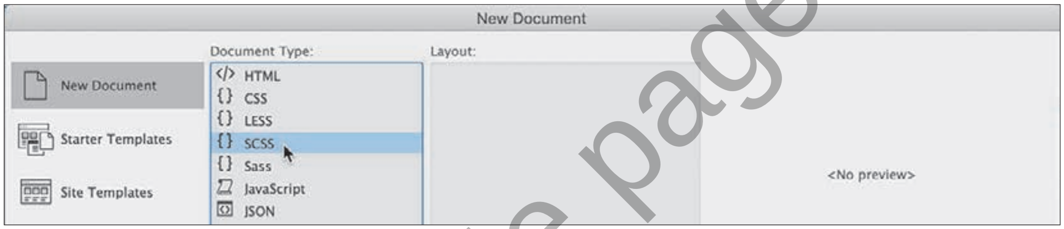
3 If necessary, open **myfirstpage.html** and switch to Split view.

The webpage is unstyled at the moment.

4 Choose File > New.

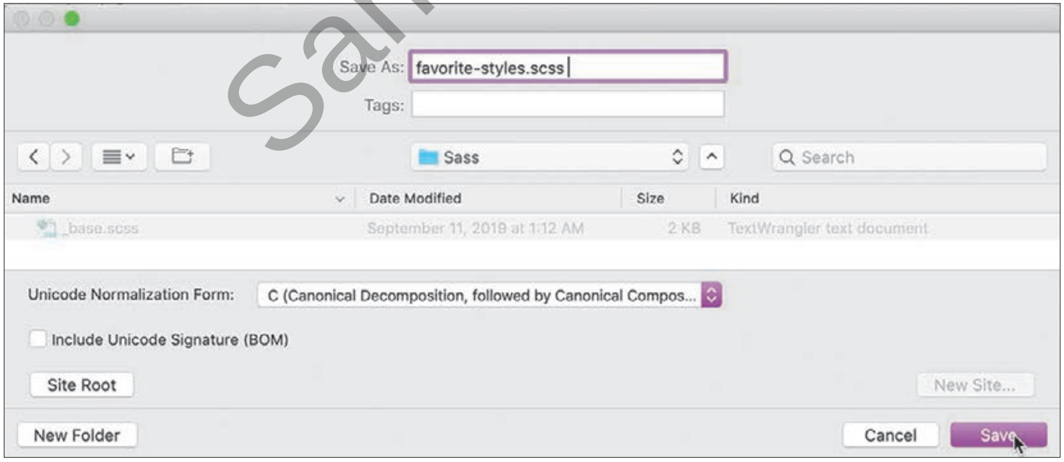
The New Document dialog appears. This dialog allows you to create all types of web-compatible documents. In the Document Type section of the dialog, you will see the LESS, Sass, and SCSS file types. We'll use SCSS in the following exercises. SCSS is a flavor of Sass that uses a syntax that is similar to regular CSS and that many users find easier to learn and work with.

5 Choose New Document > SCSS.
Click the Create button.



A new blank SCSS document appears in the document window.

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:
Sass

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)

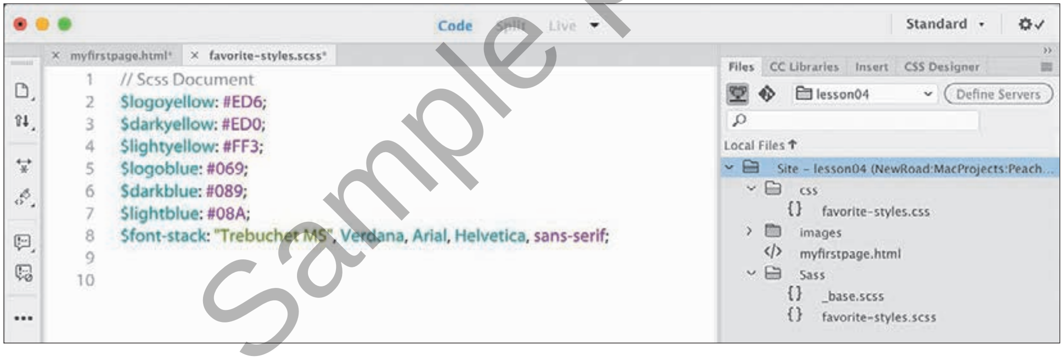
You don't need to create the CSS file; the compiler in Dreamweaver will do that for you. You're all set to start working with Sass. The first step is to define variables. *Variables* are programmatic constructs that enable you to store CSS specifications you want to use multiple times, such as colors 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 yellow 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.

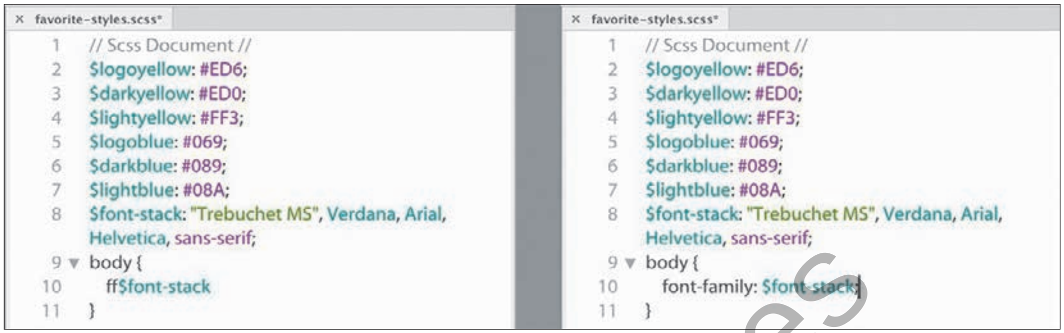
● **Note:** Your color coding may appear differently than that pictured.

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.

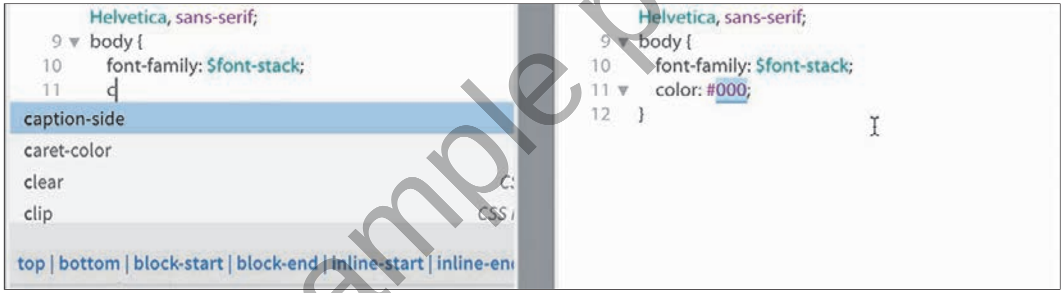


```
// Scss Document //
$logoyellow: #ED6;
$darkyellow: #ED0;
$lightyellow: #FF3;
$logoblue: #069;
$darkblue: #089;
$lightblue: #08A;
$font-stack: "Trebuchet MS", Verdana, Arial,
Helvetica, sans-serif;
body {
    ff$font-stack
}
```

```
// 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 within the body rule.
Type **`c`** and press Tab.


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

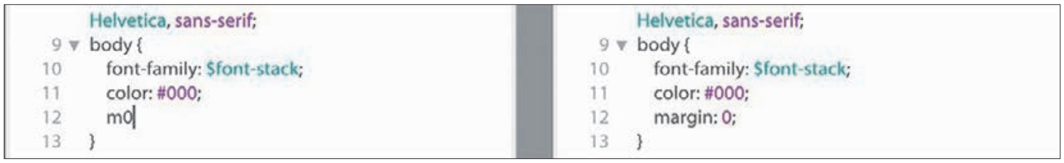
```
Helvetica, sans-serif;
body {
    font-family: $font-stack;
    c
```

```
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;
}
```

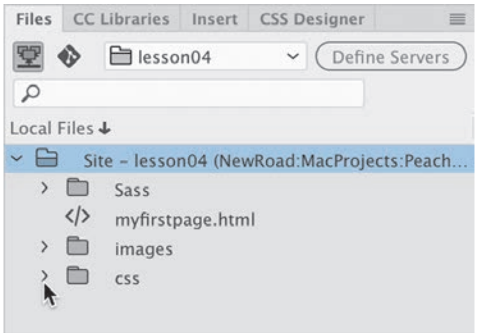
The shorthand expands to `margin: 0;` completing the basic styling for the `body` element. Before you save the file, this is a good time to see how preprocessors do their work.

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.

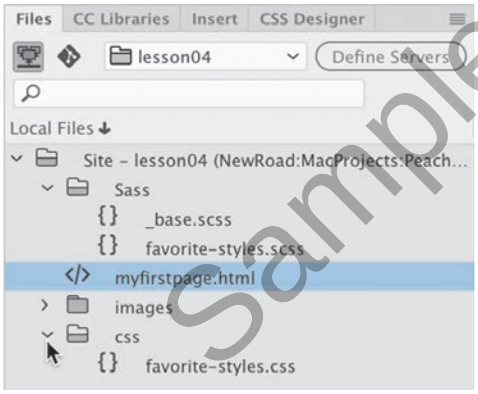
● **Note:** The order of the files on your screen may appear in a different order than pictured.

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



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

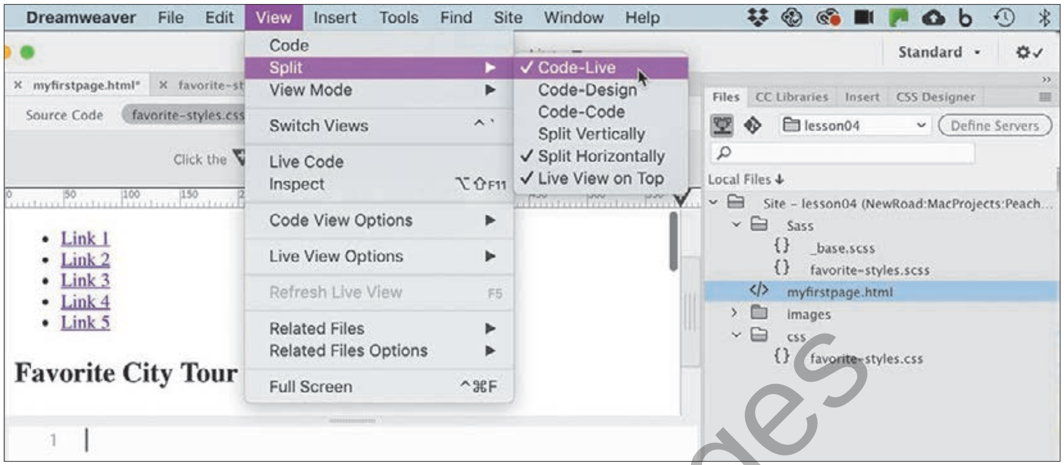
2 Expand the view of the css and Sass folders.



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

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

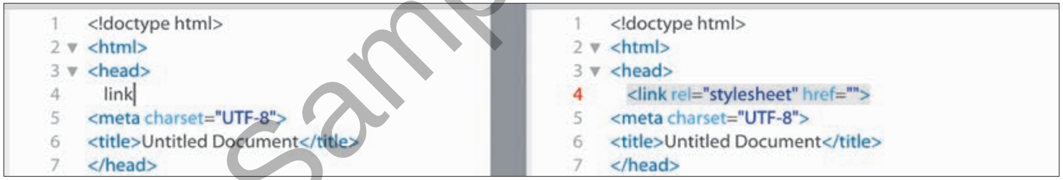
3 Select the document tab for **myfirstpage.html**.
Choose View > Split > Code-Live.
Choose View > Split > Horizontally, if necessary.



The document window is split into two windows top to bottom, showing the rendered webpage in one and the code in the other. The page shows only default HTML styling at this point.

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 Tab.



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

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

The shorthand expands to a `<link>` reference for a style sheet. It comes in with two attributes, `rel` and `href`. You'll use the `href` attribute to link the webpage to the generated CSS file.

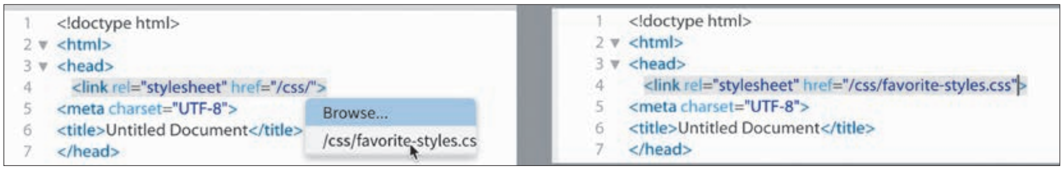
6 Insert the cursor between the quotation marks in the `href` attributes.

7 Type **`/css/`**

As you type, Dreamweaver displays a hinting menu for the file structure of the site. Once you type the second backslash, you should see the CSS file created automatically by the preprocessor.

8 Press the Down Arrow key to highlight the filename **favorite-styles.css**.

9 Press Enter/Return.



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

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

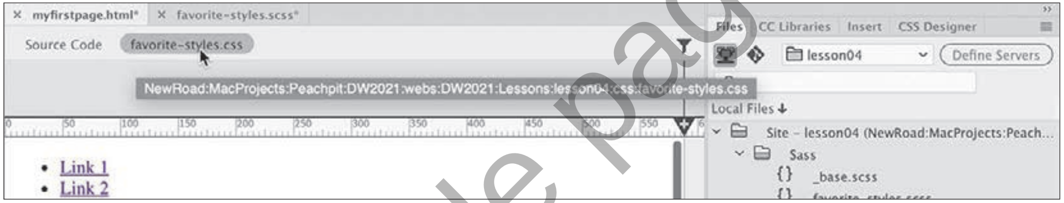
The URL to the CSS output file appears in the attribute. The link to the style sheet is now complete.

▶ **Tip:** You can also use the cursor to select the filename directly.

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.

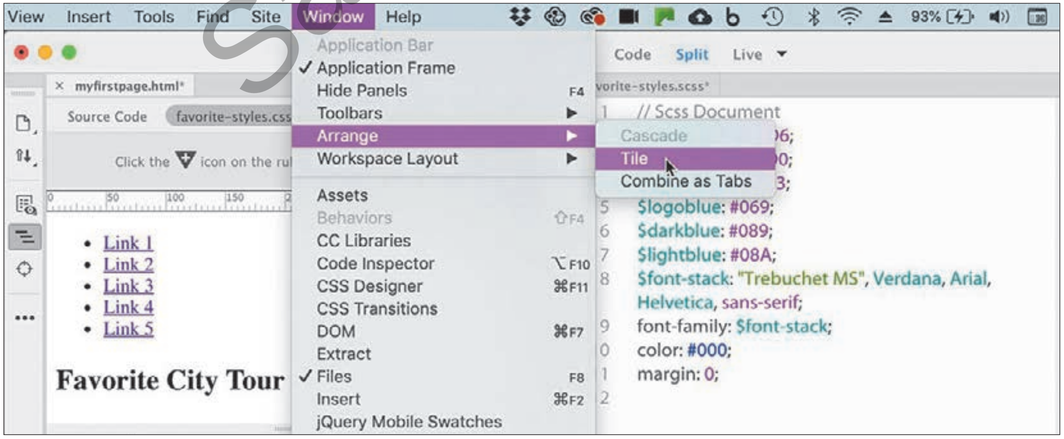
10 Select **favorite-styles.css** in the Related Files interface.



NewRoad:MacProjects:Peachpit:DW2021:webs:DW2021:Lessons:lesson04:css:favorite-styles.css

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.

11 Choose Window > Arrange > Tile.



```
// Scss Document
...6;
...0;
...3;
$logoblue: #069;
$darkblue: #089;
$lightblue: #08A;
$font-stack: "Trebuchet MS", Verdana, Arial,
Helvetica, sans-serif;
font-family: $font-stack;
color: #000;
margin: 0;
```

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