

# 2

## HTML BASICS BONUS

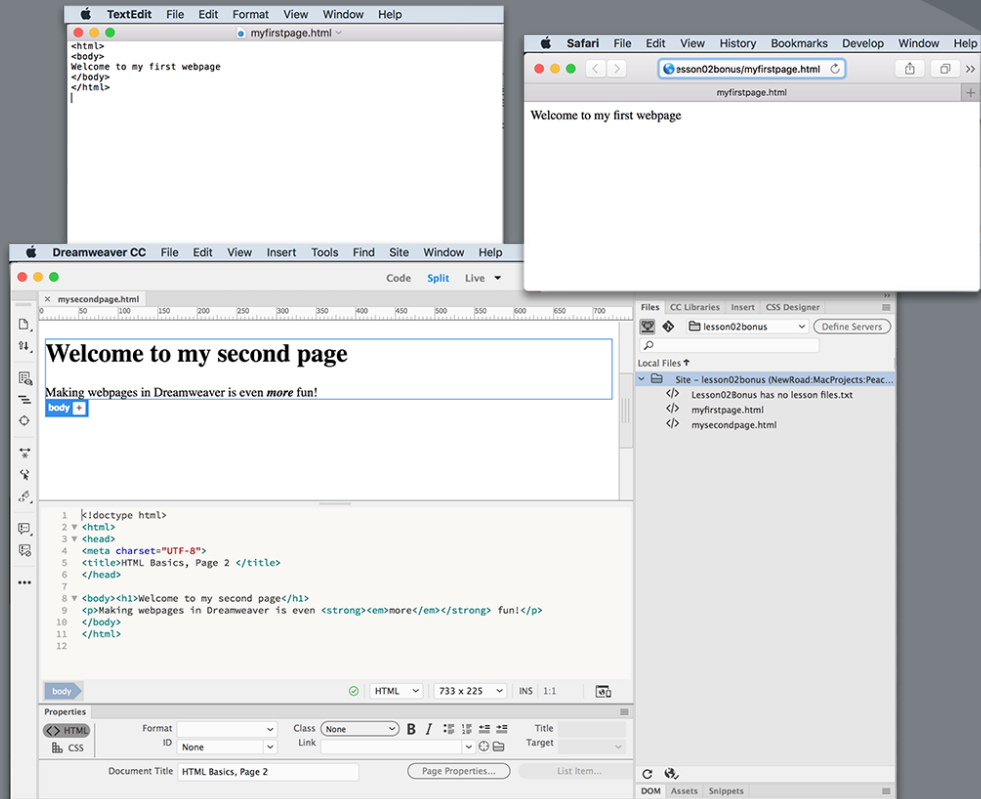
### Lesson overview

In this lesson, you will gain valuable hands-on skills and experience in writing HTML code and learn the following:

- How to write HTML code in a text editor
- How to construct the basic structure of a webpage, including the root, head, and body elements
- How to write code by hand in Dreamweaver
- How to use code hinting and other productivity enhancements in Dreamweaver
- How to preview HTML in a browser and in Dreamweaver's Live view



This lesson will take about 45 minutes to complete. If you have not already done so, please log in to your account on [peachpit.com](http://peachpit.com) to download the project files for this lesson as described in the “Getting Started” section at the beginning of this book and follow the instructions under “Accessing the Lesson Files and Web Edition.” Define a site based on the `lesson02bonus` folder.



Dreamweaver provides many productivity enhancements for writing and proofing HTML code.

# Writing your own HTML code

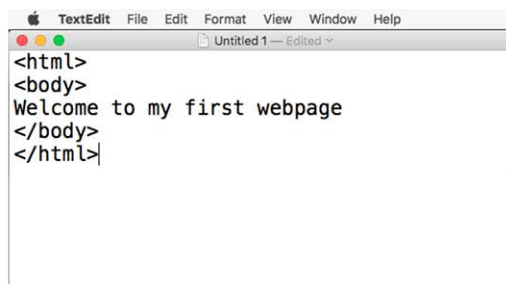
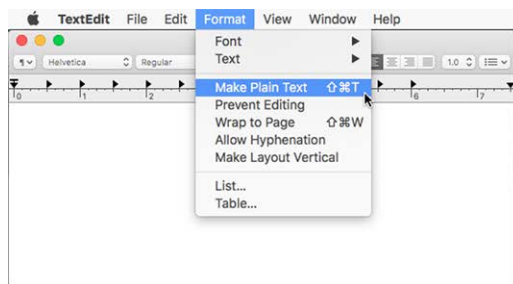
● **Note:** Feel free to use any text editor for these exercises, but be sure to save your files as plain text or text only.

● **Note:** TextEdit may default to saving the file as rich text (.rtf); in this case, you need to choose Format > Make Plain Text before you can save the file as HTML (.html).

Writing code may sound difficult or at least tedious, but creating a webpage is actually much easier than you think. In the next few exercises, you will learn how HTML works by creating a basic webpage and adding and formatting simple text content.

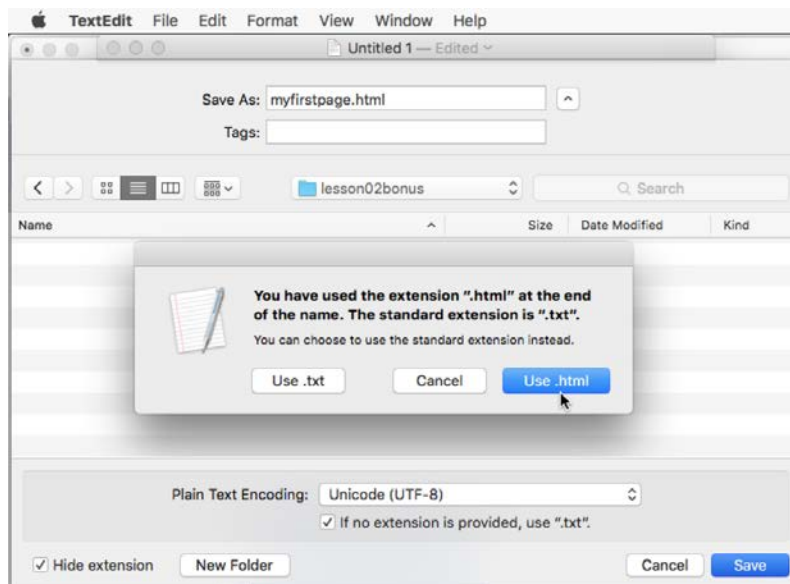
- 1 Launch Notepad (Windows) or TextEdit (Mac).
- 2 Enter the following code in the empty document window:

```
<html>
<body>
Welcome to my first webpage
</body>
</html>
```

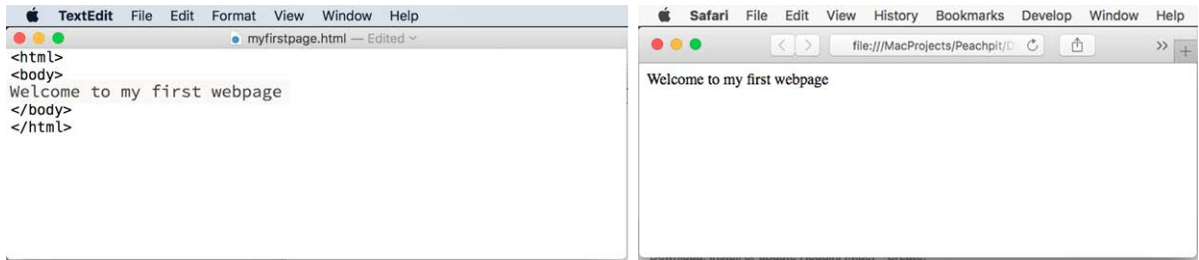


● **Note:** In text editors, you will have to type the .html extension yourself. Some text editors may try to change the .html extension or prompt you to confirm the choice.

- 3 Save the file to the lesson02bonus folder as **myfirstpage.html**.



- 4 Launch Chrome, Firefox, Internet Explorer, Safari, or another installed web browser.
- 5 Open **myfirstpage.html** in the browser.



Arrange the two programs so you can see the code in the text editor side by side with the display in the browser.

Congratulations! You just created your first webpage. As you can see, it doesn't take much code to create a serviceable webpage.

## Understanding HTML syntax

Next, you'll add content to your new webpage to learn some important aspects of HTML code syntax.

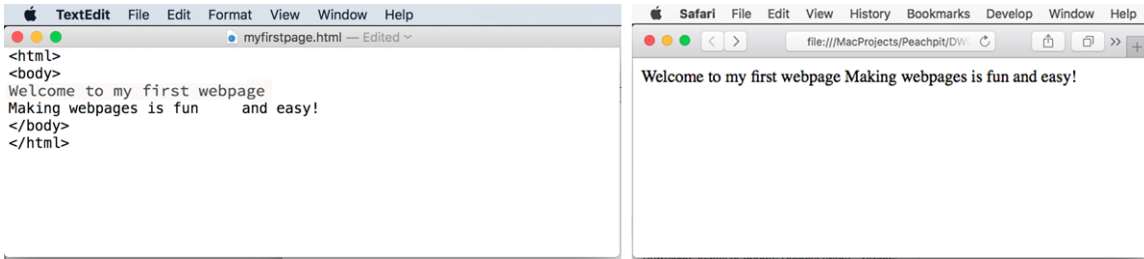
- 1 Switch back to the text editor, but don't close the browser.
- 2 Insert your cursor at the end of the text "Welcome to my first webpage" and press Enter/Return to insert a paragraph return.
- 3 Type **Making webpages is fun** on the new line.  
Press the spacebar five times to insert five spaces.  
Finish by typing **and easy!** on the same line.
- 4 Save the file.
- 5 Switch to the browser.

Although you saved the changes, you'll notice that the new text doesn't appear in the browser. That's because you never see a webpage "live" on the Internet. It must first be downloaded to your computer and saved, or *cached*, on the hard drive. The browser is actually displaying the page that it downloaded originally. To see the latest version of the webpage, you'll have to reload it.

This is important to remember as a web designer. People frequently miss changes in a website because they are looking at the cached versions of a page instead of the most current version.

- 6 Refresh the window to load the updated page.

► **Tip:** In most browsers, you can press Ctrl+R/Cmd+R to refresh the page view.



As you can see, the browser is displaying the new text, but it's ignoring the paragraph return between the two lines as well as the extra spaces. In fact, you could add hundreds of paragraph returns between the lines and dozens of spaces between each word, and the browser display would be no different. That's because the browser is programmed to ignore extra whitespace and honor only HTML code elements. By inserting a tag here and there, you can easily create the desired text display.

## Inserting HTML code

In this exercise, you will insert HTML tags to produce the correct text display.

- 1 Switch back to the text editor.
- 2 Add the highlighted tags to the text as follows:

**<p>**Making webpages is fun and easy!**</p>**

► **Tip:** Feel free to copy and paste the entity code to save time.

To add extra spacing or other special characters within a line of text, HTML provides code elements known as *entities*. Entities are entered into the code differently than tags. For example, the method for inserting a nonbreaking space is to type the `&nbsp;` entity.

► **Tip:** Another method for creating a nonbreaking space is by typing the numbered entity `&#160;` (which will generate the nonbreaking space in the browser). In most cases, the two entities are identical in use and performance; however, `&nbsp;` is not a valid entity for some applications, such as EPUB 2.0. Before you use a specific entity, make sure it is compatible with your workflow.

- 3 Replace the five spaces in the text with five nonbreaking spaces so the code looks like the following sample:

**<p>**Making webpages is fun**&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;**easy!**</p>**

- 4** Save the file. Switch to the browser and reload or refresh the page display.



The browser is now showing the paragraph return and the desired spacing.

Because you added the tags and entities, the browser can display the paragraph structure and spacing exactly as desired.

Although line breaks, extra spacing, and even indentation are ignored by the browser, web designers and coders frequently add such whitespace to make the code easier to read and edit. But don't go crazy. Although whitespace doesn't affect the display of a page in a browser, it can contribute to the time it takes to download and render a webpage overall. Whitespace and extraneous code contribute to what web developers call the overall *weight* of a page. When a page has too much weight, it downloads, renders, and operates more slowly than it should.

## Formatting text with HTML

Tags often serve multiple purposes. Besides creating paragraph structures and creating whitespace, as demonstrated earlier, tags can impart basic text formatting, as well as identify the relative importance of the page content.

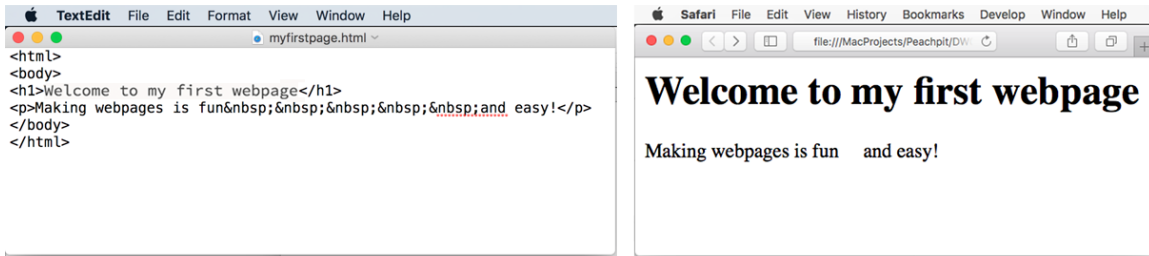
For example, HTML provides six heading tags (<h1> to <h6>) that you can use to set headings off from normal paragraph text. Not only do the tags format the heading text differently than the paragraph text, but they also impart additional meaning. Heading tags are automatically formatted in bold and often at a larger relative size. The number of the heading (1 through 6) also plays a role. Using the <h1> tag identifies the heading as being the highest in importance for search engines and other applications. The other headings are given less importance in descending order.

In this exercise, you will add a heading tag to the first line.

- 1 Switch back to the text editor.
- 2 Add the highlighted tags to the text as follows:

# Welcome to my first webpage

- 3** Save the file. Switch to the browser and reload or refresh the page display.



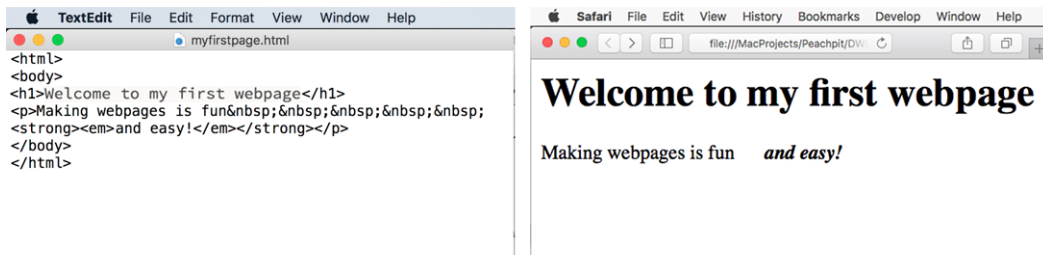
Note how the text display changed. It is now larger and formatted in boldface.

Web designers use heading tags to identify the importance of specific content and to improve their site rankings on Google, Yahoo!, and other search engines. Headings also help individuals who use screen readers and other assistive devices that allow them to access web-based content.

## Applying inline formatting

So far, all the tags you have used work as paragraph or standalone elements. These are referred to as *block* elements. HTML also provides the ability to apply formatting and structure to content that's contained within the flow of another tag, or *inline*. A typical use of inline code would be to apply bold or italic styling to a word or a portion of a paragraph. In this exercise, you will learn how to apply inline formatting.

- 1 Switch back to the text editor.
- 2 Add the highlighted tags to the text as follows:

[illegible]

- 3** Save the file. Switch to the browser and reload or refresh the page display.

*Strong* and *em* (emphasis) are used in place of the tags for bold (<b>) and italic (<i>) because they provide enhanced semantic meaning for visitors with disabilities or visual impairment, but the visual result is basically identical.

Most formatting, both inline and otherwise, is properly applied using cascading style sheets (CSS). The <strong> and <em> tags are among the few still acceptable ways to apply inline formatting using specific HTML code elements. Technically speaking, these elements are intended more to add semantic meaning to text content than to add styling, but the result is the same. Today, modern browsers still display <strong> and <em> as bold and italics, respectively.

However, this may change in the near future. There has been a move to separate the content from its presentation, or formatting, over the last decade. Although most browsers and HTML readers currently apply default formatting based on specific tags, this may not always be the case. See Lesson 3, “CSS Basics,” for a full explanation of the strategy and application of CSS in standards-based web design.

● **Note:** The <b> and <i> tags were resurrected in HTML5 to provide bold and italic styling when no semantic value is needed.

## Adding structure

Most webpages feature at least three fundamental elements: root (typically <html>), <head>, and <body>. These elements create the essential underlying structure of the webpage. The root element contains all the code and content, with the exception of any dynamic code that must load before the page content itself. This dynamic code is usually written in web programming languages such as ASP, Cold Fusion, or PHP. The <head> element holds code that performs vital background tasks, including styling, external links, and other metadata. The <body> element holds all the visible content for the webpage, such as text, tables, images, movies, and so on.

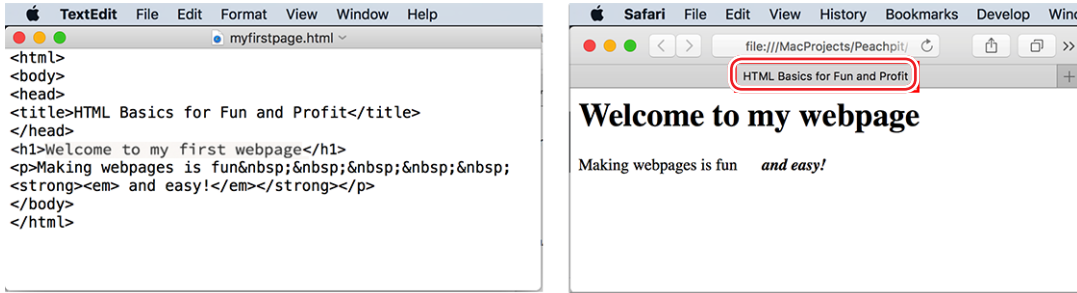
The sample page you created doesn’t have a <head> element. A webpage can exist without this section, but adding any advanced functionality to this page without one would be difficult. In this exercise, you will add <head> and <title> elements to your webpage.

- 1 Switch back to the text editor.
- 2 Add the highlighted tags and content as shown here:

```
<html>
<head>
<title>HTML Basics for Fun and Profit</title>
</head>
<body>
```



- 3 Save the file. Switch to the browser and reload or refresh the page display.



● **Note:** In some instances, tabs may be turned off by default when single pages are displayed. If you don't see the page title, check the view settings in the browser.

Did you notice what changed? It may not be obvious at first. Look at the title bar or window tab of the browser. The words “HTML Basics for Fun and Profit” now magically appear in the tab or title bar above your webpage. By adding the `<title>` element, you have created this display yourself. But it's not just a cool trick; it's good for your business too.

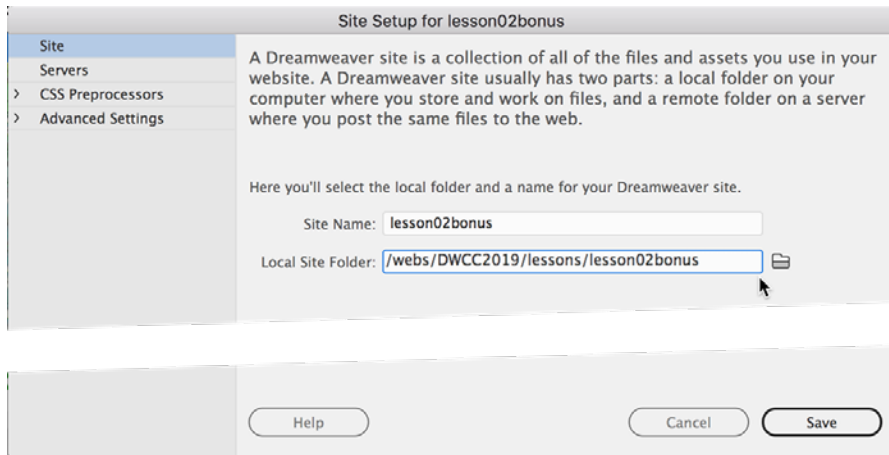
Google, Yahoo!, and the other search engines catalog the `<title>` element of each page and use it, among other criteria, to index and rank webpages. The content of the title is one of the items typically displayed within the results of a search. It also appears automatically when you create a bookmark for that page in your browser.

A well-titled page could be ranked higher than one with a bad title or one with none at all. Keep your titles short but meaningful. For example, the title “ABC Home Page” doesn't really convey any useful information. A better title might be “Welcome to the Home Page of ABC Corporation.” Check out other websites (especially peers or competitors) to see how they title their own pages.

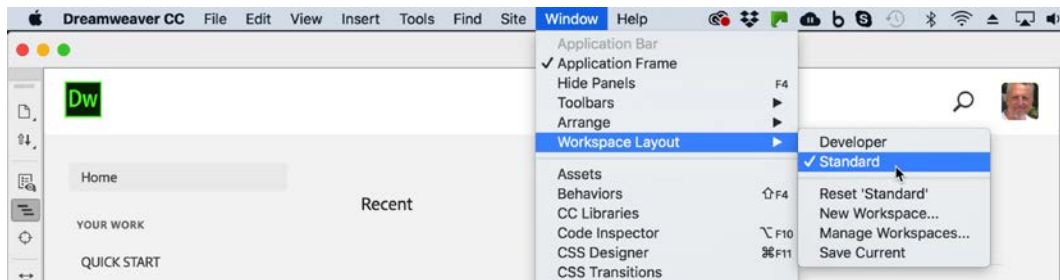
## Writing HTML in Dreamweaver

So the inevitable question is, “If I can write HTML in any text editor, why do I need to use Dreamweaver?” Although a complete answer awaits you in the 13 lessons in the book and the seven online bonus lessons, the question begs a quick demonstration. In this exercise, you will re-create the same sample webpage using Dreamweaver.

- 1 Launch Dreamweaver CC (2019 release) or later.
- 2 Define a new site based on the **lesson02bonus** folder as described in the “Getting Started” section at the beginning of the book.

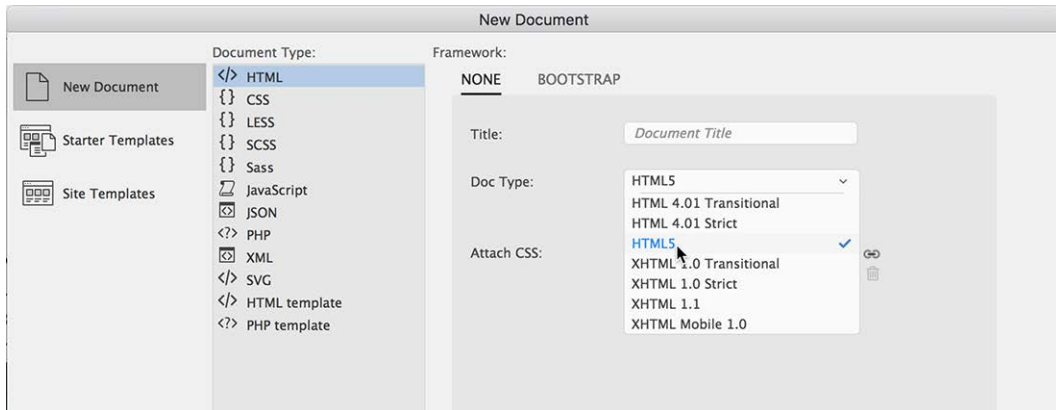


- 3 If necessary, select the Standard workspace from the Workspace menu, or choose Window > Workspace Layout > Standard.



- 4 Choose File > New.
- 5 In the New Document dialog, select the New Document category.
- 6 Select HTML from the Document Type column, if necessary.
- 7 In the Framework section, choose the None tab.

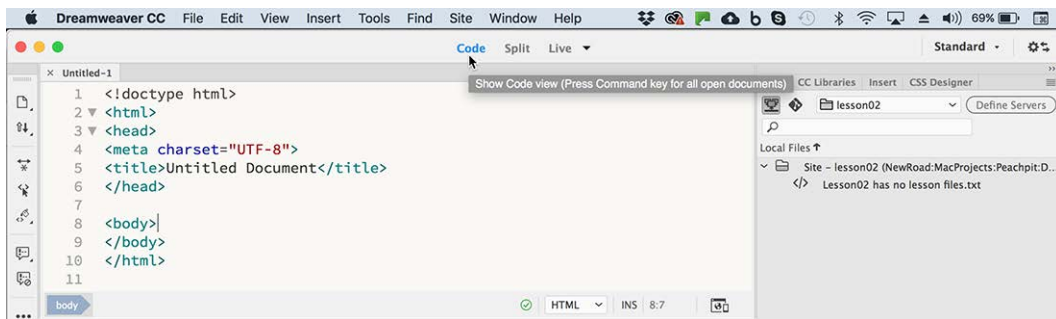
- 8 If necessary, in the Doc Type menu, select HTML5.



- 9 Click Create.

A new empty HTML document appears in Dreamweaver. The document window may default to one of four displays: Live view, Code view, Design view, or Split view.

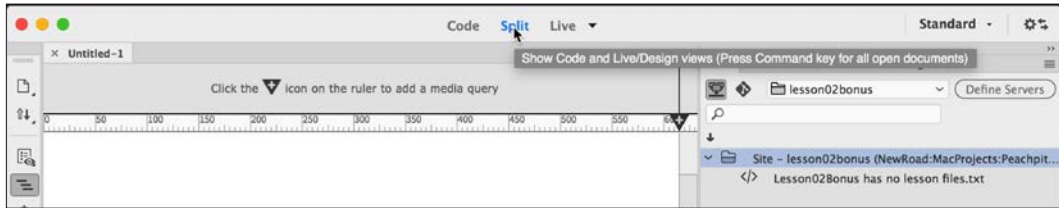
- 10 If it's not already selected, click Code view at the top of the document window.



The first thing you should notice in Code view is that Dreamweaver gives you a huge head start over the text editor. The basic structure of the page is already in place, including the root, head, body, and title elements, among others.

Another advantage is evident when you need to view the results of your coding efforts. The text editor required the use of a separate application to preview the HTML code. Dreamweaver provides a built-in method.

- 11 Click the Split view button.



In Split view, the interface is divided into two windows. One will show the HTML code; the other can be used to provide an accurate preview of the finished webpage. This preview alone can save you hours of time loading and previewing pages in a separate application.

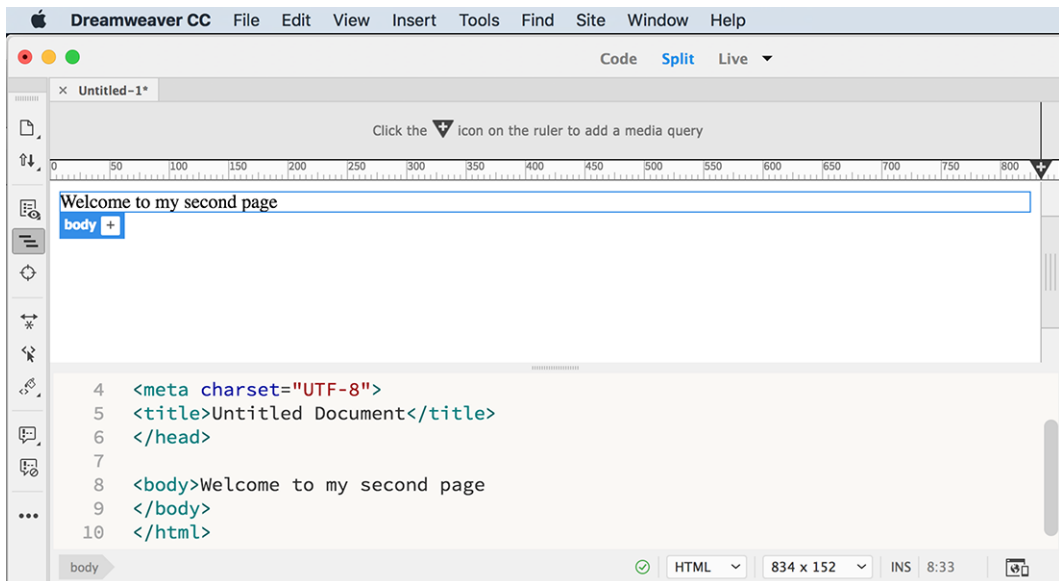
- 12 If necessary, activate Live view in Split view.

The program interface should now be divided in half, displaying Code view in one window and Live view in another. Dreamweaver also makes it easy to write HTML code.

- 13 In the Code view window, insert the cursor after the `<body>` tag and enter the following text:

**Welcome to my second page**

**Note:** Split view may use either Live view or Design view to display side by side or top to bottom with the Code window. For simple pages like this, either one will do. See Lesson 1, "Customizing Your Workspace," for more information on how you can tailor your workspace to fit your needs.



Live view renders the HTML code as it would appear in a web browser. The current version is based on the same engine as Apple's Safari. After you enter the text, it may appear immediately in the Live view window. However, sometimes you may need to refresh the preview to see the changes.

**14** Click in the Live view window to refresh the preview, if necessary.

The text appears in Live view without any special formatting.

**15** In the Code view window, insert the cursor at the beginning of the text “Welcome to my second page.”

**16** Type **<**



A drop-down menu appears next to the cursor. This is Dreamweaver’s *code-hinting* feature displaying a list of compatible HTML, CSS, JavaScript, and other supported code elements.

**17** Type **h**

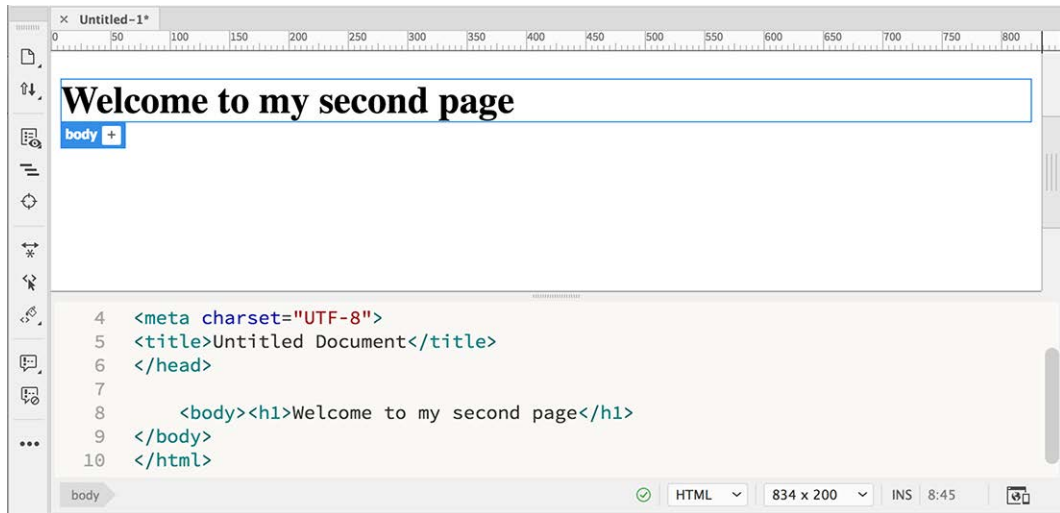


The list filters as you type, showing only elements that match the entered characters. You can continue typing the tag name manually or select it using the mouse or keyboard.

**18** Double-click **h1** from the list to insert it in the code. Type **>** to close the opening tag, if necessary.

**Note:** Depending on your preference settings, Dreamweaver may create only the opening tag or the entire element at once. The following steps assume that only the opening tag is created. Feel free to adjust the code-completion preferences to your liking.

**19** Move the cursor to the end of the text. Type `</` at the end of the sentence.



Note that Dreamweaver closes the `<h1>` element automatically. In this case, you entered the `<h1>` tags after the fact, but many web designers add the tags as they write.

**Note:** If the `<h1>` and `<p>` elements are closed automatically in step 18 and 21, you may need to move the closing tags to the end of the text manually.

**20** In Code view, press Enter/Return to insert a line break.

Type `<p` and press Enter/Return.

Type `>` to close the tag.

**21** Type **Making webpages in Dreamweaver is even more fun!** and then type `</` to close the `<p>` element.



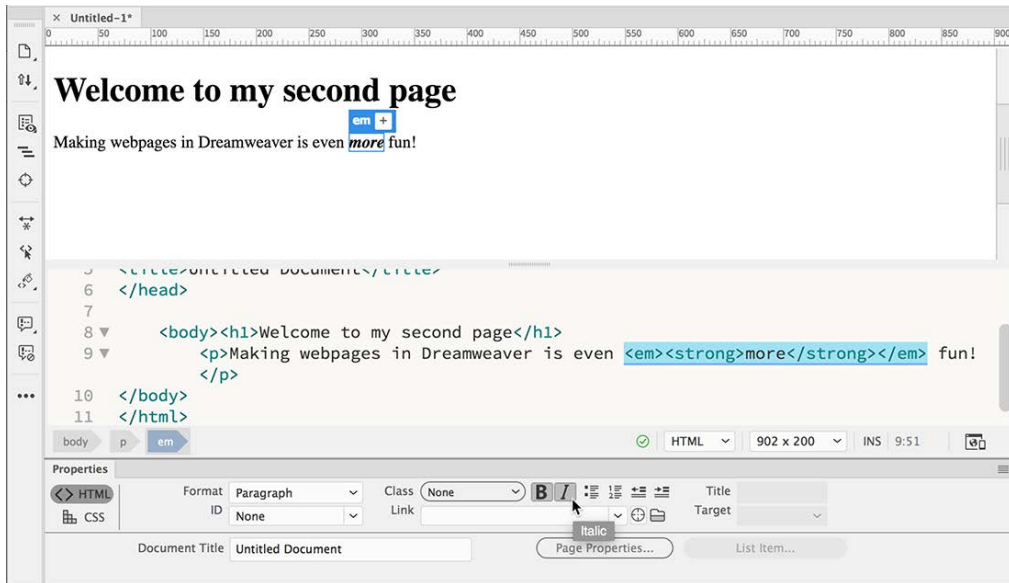
Tired of hand-coding yet? Dreamweaver offers multiple ways to write code automatically.

**22** Select the word “more.”

**23** Choose Window > Properties to display the Property inspector, if necessary.

This panel is an important component to many workflows in Dreamweaver. If it appears as a floating panel, you can dock it to the bottom of the document window so that it will be handy when you need it. See Lesson 1 for more information on how to customize the Dreamweaver interface.

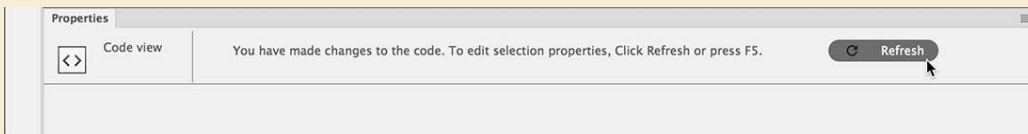
**24** In the HTML tab of the Property inspector, click the **B** button and the **I** button to apply <strong> and <em> tags to the selected text.



These tags produce the appearance of bold and italic formatting on the selected text.

## Something missing?

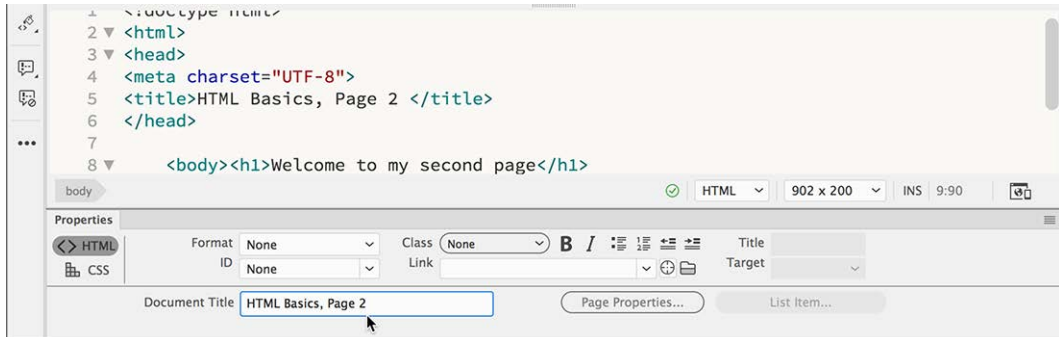
When you reached for the B and I buttons in step 24, were they missing? When you make changes in Code view, the Property inspector occasionally needs to be refreshed before you can access the formatting commands and metadata fields featured there. Simply click the Refresh button to make the formatting commands and other tools reappear.



If you do not see the Property inspector, you can display it by choosing Window > Properties.

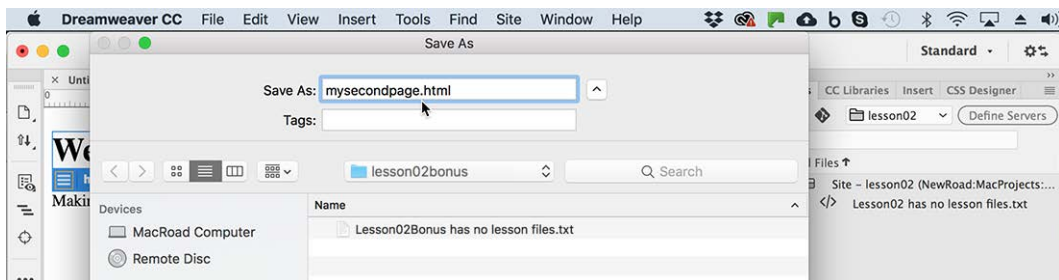
Only two more tasks remain before your new page is complete. Note that Dreamweaver created the `<title>` element and inserted the text “Untitled Document” within it. You could select the text within the code window and enter a new title, or you could change it using another built-in feature.

- 25 Locate the Document Title field in the Property inspector, and select the Untitled Document placeholder text.
- 26 Enter **HTML Basics, Page 2** in the Document Title field.  
Press Enter/Return to complete the title.



The new title text appears in the code, replacing the original content. It's time to save the file and preview it in the browser.

- 27 Choose File > Save.  
Navigate to the folder lesson02bonus.  
Name the file **mysecondpage** and click Save.

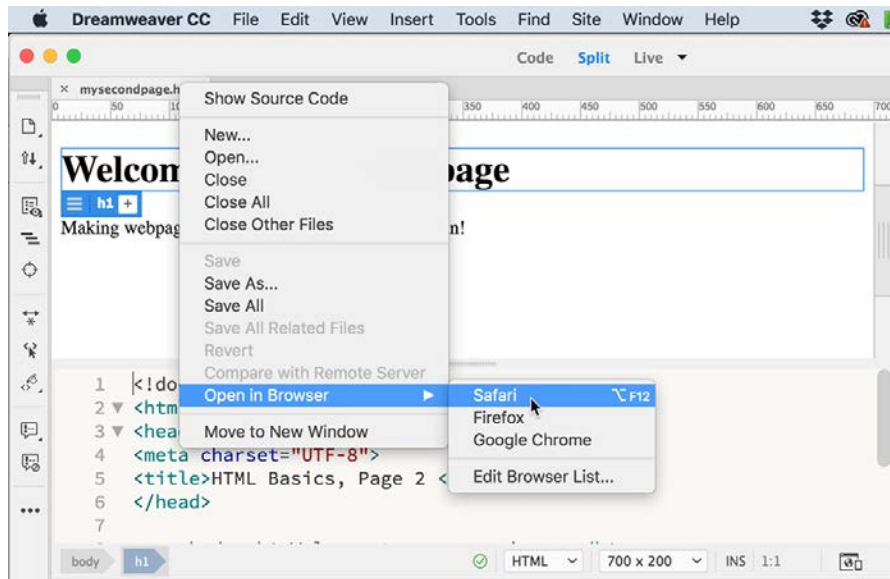


Dreamweaver adds the proper extension (.html) automatically.



● **Note:** Dreamweaver uses the browsers already installed on your computer. You may install additional, alternate browsers and configure their use in the Dreamweaver Preferences dialog.

- 28 Right-click the document tab displaying the name of the file. Select Open In Browser from the context menu, and select your favorite browser.



The completed page appears in the browser window.

Using Dreamweaver you completed the task in a fraction of the time it took you to do it manually in a text editor.

You have just completed two webpages—one by hand and the other using Dreamweaver. In both cases, you can see how HTML played a central role in the whole process. To learn more about this technology, go to the website of the W3 Consortium, [www.w3.org](http://www.w3.org), or check out any of the following books:

- *Introducing HTML5, 2nd Edition* (Peachpit Press, 2012), ISBN: 978-0-321-78442-1
- *HTML and CSS: 8th Edition* (Peachpit Press, 2014), ISBN: 978-0-321-92883-2
- *HTML5 Pocket Reference, 5th Edition* (O'Reilly, 2013), ISBN: 978-1-449-36335-2