

# Lesson 2: Printer Friendly

---

## Create Your First Web Page

[Lessons](#) > [Lesson 2](#) > Printer Friendly

### Chapter 1

---



#### Introduction

Hello, and welcome back! In Lesson 1, we talked about how the Internet and Web works. We discussed how you can start a site by creating a folder on your computer and using it to store the pages that make up your site. That way you can work on the site out of the view of the public. And when you're happy with the website, you can upload it to a Web server, where it will have a URL that allows anyone in the world to view it. You also created a folder for a site in Lesson 1, and you named it *MyWebsite*.

In today's lesson, you're going to create your first Web page and put it in that folder. You should have chosen what you'll use as your text editor to create your pages by now. If you don't have a preference, you can use Notepad in Windows or TextEdit on a Mac, as discussed in the assignment for the last lesson. If you skipped the assignment, you'll want to go back and do that now. Because this lesson assumes you've completed that assignment.

If you're ready to go, let's meet in Chapter 2 and get started!

### Chapter 2

---

#### Create Your First Web Page

As we discussed in Lesson 1, every website in the world consists of one or more Web pages. This lesson that you're reading right now is a Web page from a website. Every Web page contains *source code* and *content*. The content is the stuff you see when you view the page in a browser. For example, as you look at this lesson, you see the content (text and pictures).

The source code (often referred to simply as *code* by professionals) is stuff that you don't see in the browser that determines the look and feel of the content. For example, you use code to define which text are headings, which are paragraphs, where pictures will be placed, and so forth.

#### Tip

The word *code* is always singular, like water or money. We drink water (not waters), we make money (not monies), and we write code (not codes).



There are many computer languages for writing code. When you're writing Web pages and some other kinds of electronic documents, the main language you'll use to write your code is HTML—Hypertext Markup Language.

Hypertext Markup Language consists of *tags* that define what each element in a document *is*. There are about 100 different tags, and they all have one thing in common: Each tag starts with an opening angle bracket (<) and ends with a closing angle bracket (>).

There are certain mandatory tags that every Web page *must* contain. So for our first hands-on exercise, you'll create a Web page that contains those mandatory tags. To get started, you'll need to open whatever program you're going to use to write and edit your Web pages. Again, if you don't have a preference, you can use Notepad in Windows or TextEdit on a Mac. In that editor, start by typing this tag:

```
<html>
```

When it comes to typing code, you have to be very accurate and type exactly what I show you. It's not like typing text for humans, where you can make all kinds of mistakes and the person reading it can figure out what you meant anyway. The code you style is instructions for computers, not something people read. Computers don't have eyes to see with or brains to figure things out with. You have to type the code correctly and without mistakes. Otherwise, there's a good chance it won't work.

So notice that the first tag I had you type contains no blank spaces and uses all lowercase letters. If you typed yours differently, now would be a good time to correct it. And try to make a mental note not to make whatever mistake you made the first time when typing additional code.

Now, please press ENTER to start a new line. Then type one more tag as shown below.

```
<head>
```

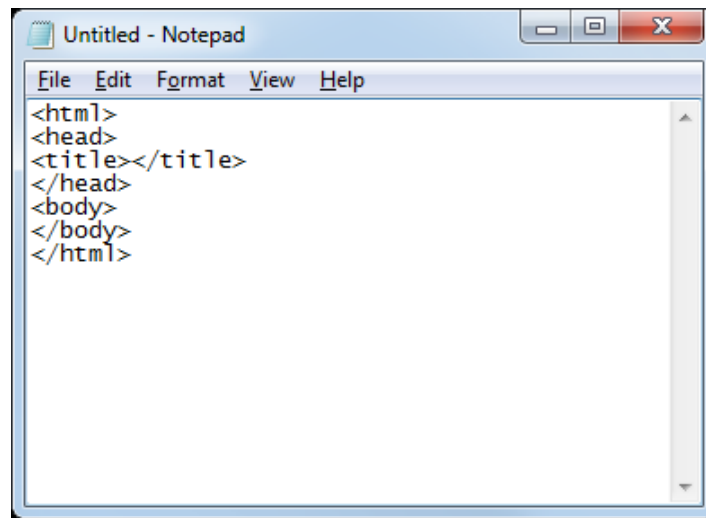
Next, we'll type a couple of tags right next to each other. Notice that here you'll start to see that some tags contain a forward slash (/). That's not the same as a backslash (\), which leans back on the letter that precedes it. Make sure you always use a forward slash (/) in your HTML tags. Here are the two tags I want you to type next to each other now:

```
<title></title>
```

Now, you can press ENTER to end that line. And let's just go ahead and type in the rest of the required tags. I'll discuss what they all mean and why they're there in a moment.

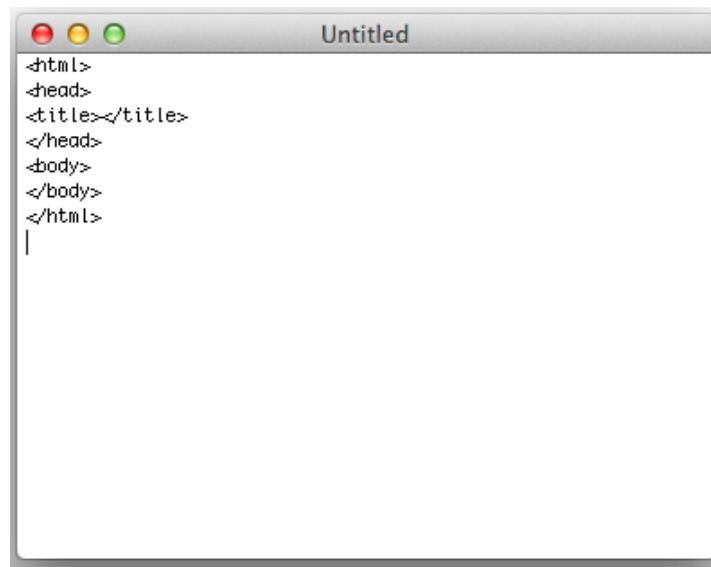
```
</head>  
  
<body>  
  
</body>  
  
</html>
```

Your sample Web page should look like this if you're using Notepad as your editor:



Your first Web page, in Notepad

And here's how it looks in TextEdit:

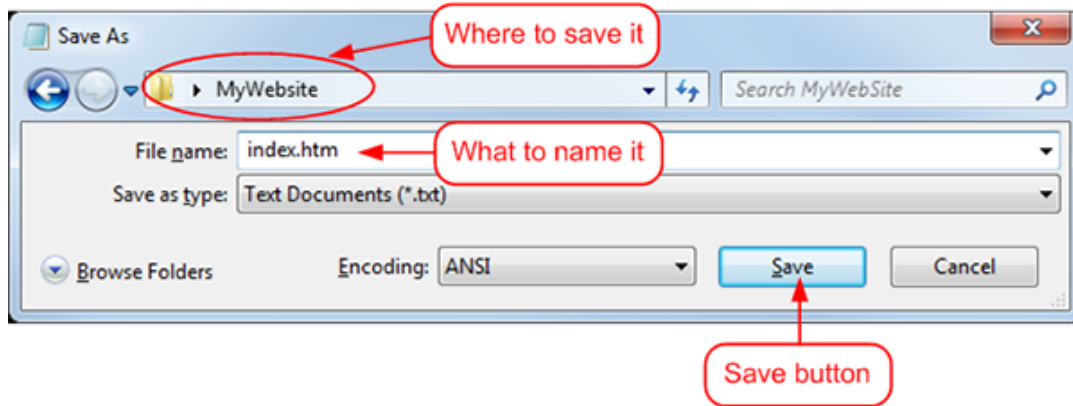


Your first Web page, in TextEdit

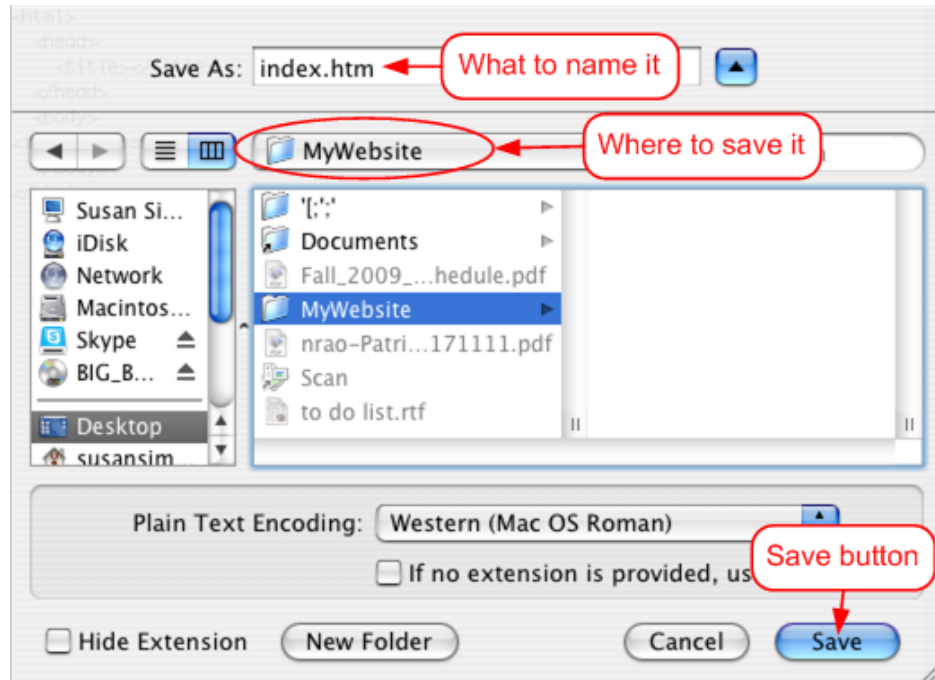
Do you see any errors in your own code? If so, now would be a good time to correct them.

Before we go any further, let's save all this hard work. I'm going to have you save it as *index.htm*, because that's a common name for a site's home page (the first page one sees when visiting your site). Here are the steps:

1. Close your editor. In Windows, click **Close (X)** in the upper right corner, or choose **File > Exit** from the menu bar. In TextEdit, click the red **Close** button.
2. When asked if you want to save your work, click **Save** or **Yes**—or whatever indicates the affirmative in your program.
3. Choose **MyWebsite** as the folder that you want to save the file in.
4. Type *index.htm* as the filename, as shown in the images below.



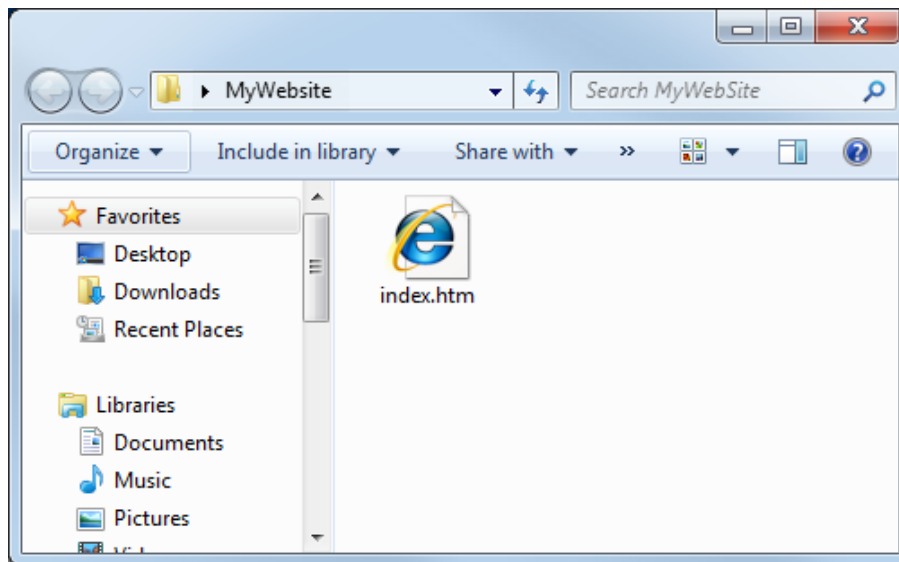
Save page in Notepad



Save page in TextEdit

5. Click the **Save** button.

At this point, the editor and page should close. The page is saved with the filename you specified in the folder you specified. In Windows, that might look something like the image below. The exact size and shape of the icon depends on how you're viewing icons in the folder. And the icon you see will match the icon of your default browser, which may be different from the examples shown. But the filename should be *index.htm*.



Index.htm in MyWebsite folder (Windows 7)

The file will be indicated by an icon in your MyWebsite folder on a Mac too. The exact appearance of that icon depends on how you're viewing icons and your version of Mac OS. But usually it shows as blank document, or a document with a Safari icon on it and the filename index.htm as in the example below.



Index.htm in MyWebsite folder (Mac)

If your file isn't in your MyWebsite folder, you must have goofed in step 3 above where you tell your editor to save the file in that folder. If *index* shows as the name without the .htm filename extension, then perhaps you skipped the assignment for Lesson 1, or perhaps you did something wrong where we made selections to ensure filename extensions are visible. But assuming all is well, let's move on to Chapter 3. There you'll take a look at your first Web page in a web browser, where it will be empty, for reasons you'll discover there.

## Chapter 3

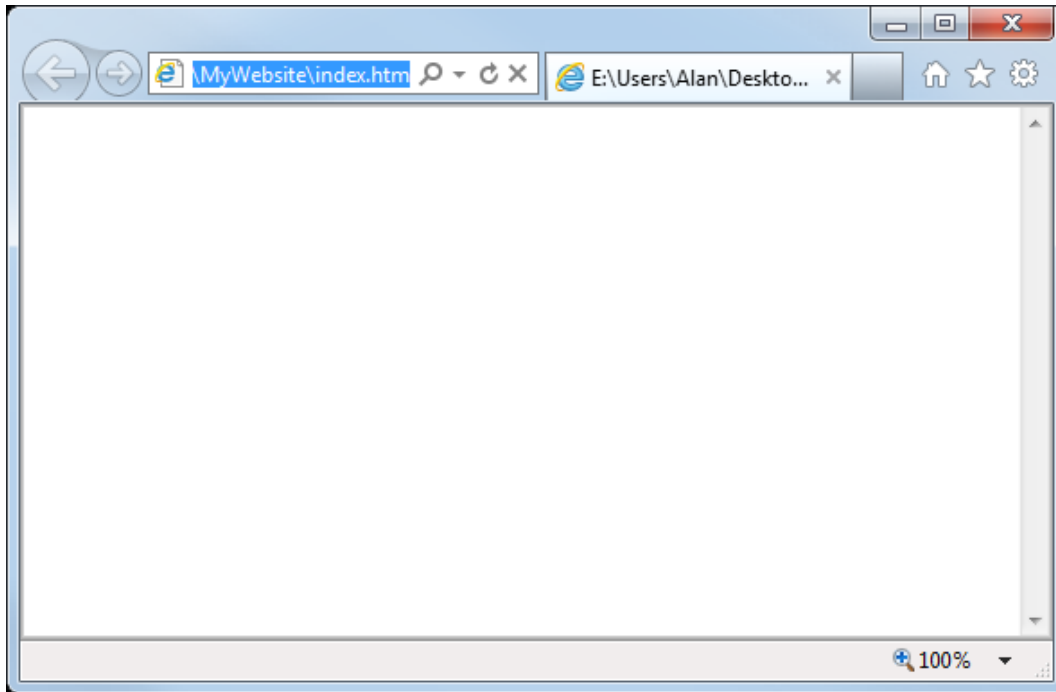
### Browsing and Editing Pages

Let's begin by covering some of the basics about files.. The index.htm page you created in Chapter 2 is a Web page—a plain text file that contains HTML tags, and the filename has a .htm filename extension. All operating systems treat the .htm and .html filename extensions equally as Web pages. There's no advantage or disadvantage to using one extension or the other.

When you look at the page in your text editor, you see the source code—the HTML tags you typed there. To see the page as the rest of the world would see it, you have to view the page in a Web browser. When you open the file in a Web browser, you see the *rendered page* rather than the source code. You'll see a few examples in this lesson.

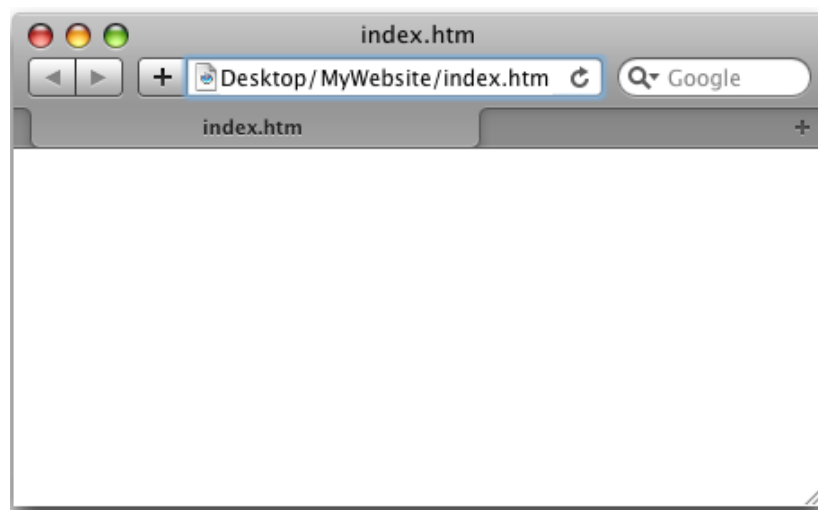
To open index.htm in a Web browser, you just have to double-click the index.htm file's icon in your MyWebsite folder. The page will open in your default Web browser. That'll be whatever you've defined as your default browser in the past. If you've never done anything like that in the past, then most likely your default browser will be Internet Explorer (in Windows) or Safari (on a Mac).

When you open index.htm in a browser, you won't see any source code at all. It's there! You just won't see it. In fact, you won't see anything but an empty page. Why? Because the tags you put into the page define the structure of the page only. You didn't yet put in any content for people to see. Here's how an empty page looks in Internet Explorer (version 9):



Index.htm in Internet Explorer (Version 9)

On a Mac, your empty page will look something like this:



Index.htm in Safari (Version 3)

If double-clicking the file didn't open it in a Web browser on your computer, perhaps you misspelled the filename extension. It should be .htm (though .html would work too).

Why anyone would want to go to the trouble of typing all those tags, only to end up with an empty white page in a Web browser? The answer to that question is actually very important for all Web developers to understand. It's the kind of thing that helps to separate the beginners from the pros. In fact, it's important enough to be typed on its own line:

Web development isn't *just* about how a page looks in your Web browser.

Even though our index.htm page looks empty to a human who is viewing the page in a browser, it's not empty to *user agents*. To a *user agent*, the tags in the source code are still there and provide useful information.

A user agent (sometimes abbreviated *UA*) is any program that might access the page. Your Web browser is a user agent. Screen readers for the blind, which read the page aloud, and other assistive devices for people with sensory impairments are user agents. The *Web-crawling infobots* (also called *spiders*, *crawlers*, and *bots*) that the search engines like Google use to index your page and make it easy for people to find are user agents. The various browsers and browser-like programs on cellphones, e-readers, and tablets are also user agents.

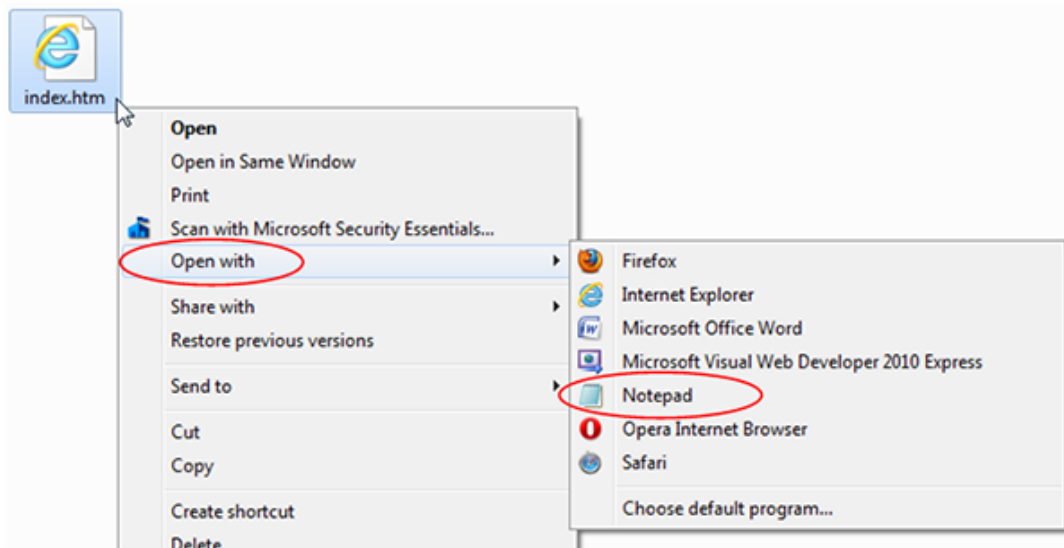
Unlike humans, who only care about content, user agents need information *about* the page to best decide how to present it to the user (the person using the user agent). And that's what the tags you typed provide—information that's for the user agent rather than the human user's eyes. Being a developer will require you to extend your horizons beyond the surface of what appears on the screen for human eyes to see. If that's a little hard to digest right now, just put it on the back burner of your mind. You'll be seeing more examples throughout this course. For now, we have something simpler and more pressing to deal with.

Nobody wants to look at an empty Web page. So needless to say, your index.htm page isn't exactly ready for publishing to the Web. To be useful, you need to add some content to the page. (Recall that content is what people see, unlike code, which is what programs see). If you try to just start typing content into your browser, nothing will happen. Why? Because Web browsers are for *viewing* Web pages, not for creating or editing Web pages. If you want to change or add to a page, you're going to have to open it in your editor again. Which brings us to .

## Opening a Page for Editing

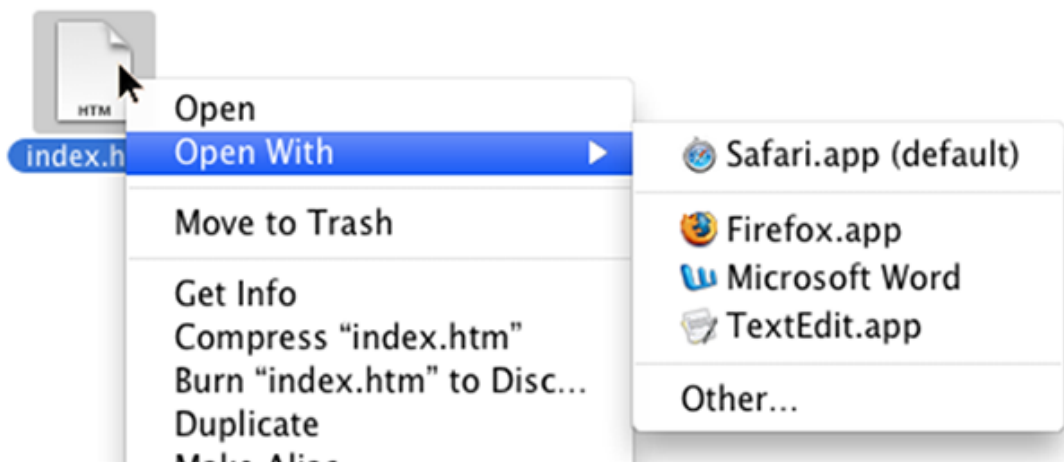
To edit (change) a Web page, you'll need to open it in your editor again. This should be easy to do:

1. Open the folder that you stored the page in (MyWebsite in this course).
  - In Windows, right-click the page's icon (index.htm), point to **Open With**, and then click the name of your editor (Notepad unless you're using something else). See the image below.



Open page in Notepad (Windows)

- On a Mac, CTRL + click the icon, and choose **Open With** and the name of your editor (TextEdit unless you're using some other program). See the image below.



In your editor, you're now back to seeing the source code, which means that you can see all the tags. Let me take a moment to explain what these tags mean. In the code below, I've added some indentations that might make it easier to see how some tags are contained within other tag pairs. The indentations aren't required. Your code works the same with or without them. The indentations are for your eyes only—and not because there's any technical requirement for indentations. But if you want to add them to your own page, you're certainly welcome to, just by adding blank spaces in front of any lines you want to indent.

```
<html>
  <head>
    <title></title>
  </head>
  <body>
  </body>
</html>
```

Source code with some indents

- **<html>**: The very first tag, `<html>`, tells the user agent that's opening the page that it's about to encounter an html document. That very first tag will vary depending on the version of HTML or XHTML you're using. The `<html>` tag is kind of a plain vanilla tag that'll work for right now. We'll talk about other opening tags a little later. But it's nothing to be concerned about right now, because the rest of the tags are core HTML—which means that they're exactly the same regardless of which version of HTML or XHTML you're using to create your page.
- **</html>**: At the very bottom of the page, you see `</html>`. That tag tells the user agent that it has reached the end of the html document.

#### Note



As you can see, most of the tags come in pairs. For example, `<html>` at the top of the page and `</html>` at the bottom. Most tags in HTML come in pairs like that. The first tag in the pair is called the *opening tag* for the pair, and never has a slash (/) in it. The second tag is called the *closing tag* and is always the same as the opening tag except that it has a slash (/) right after the `<` at the start of the tag.

- **<head>**: After the `<html>` tag comes the `<head>` tag. That starts the head element in the page. The head element will contain metadata. Metadata is information about the page for user agents to use. It's not information that's visible on the page body for people to see.
- **<title> and </title>**: Inside the head element are `<title>` and `</title>` tags. These tags will contain the page title. Not a title that appears on the page. But rather, a title that appears in the Web browser program window, as you'll see shortly.
- **<body> and </body>**: Below the head section are `<body>` and `</body>` tags. Those define the body element of the page. All the content of the page goes between those tags. The content is the stuff that people will see when they visit your website using their Web browsers.

The tags above are the only tags that all pages require. From here on out, you can be creative with your tags and content. Let's start by adding some text to the page. Here are the steps:

1. With `index.htm` still open in your editor, type the words *My Website* between the `<title>...</title>` tags, as below. (Feel free to make it more personal by using your own name if you'd like).

```
<title>My Website</title>
```

2. Between the `<body>` and `</body>` tags, type *Welcome to My Site*.
3. Press ENTER to start a new line
4. Type *This is text on my home page*.

In the image below, I've indicated in red the new text you added to your page. That text won't be (and can't be) red in your editor. I just made it red here to make it easier for you to see its placement. In your page, you just want to make sure the new title text is between the `<title>...`



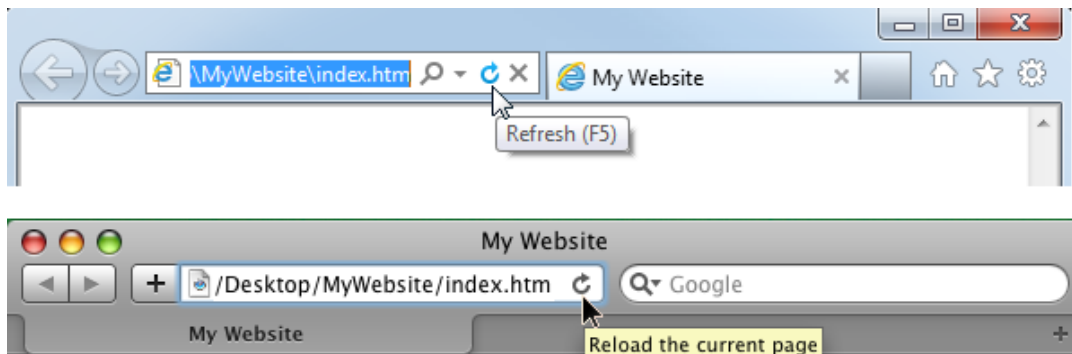
</title> tags and the rest is between the <body>...</body> tags.

```
<html>
  <head>
    <title>My Website</title>
  </head>
  <body>
    Welcome to My Site
    This is text on my home page.
  </body>
</html>
```

Some content added to our page

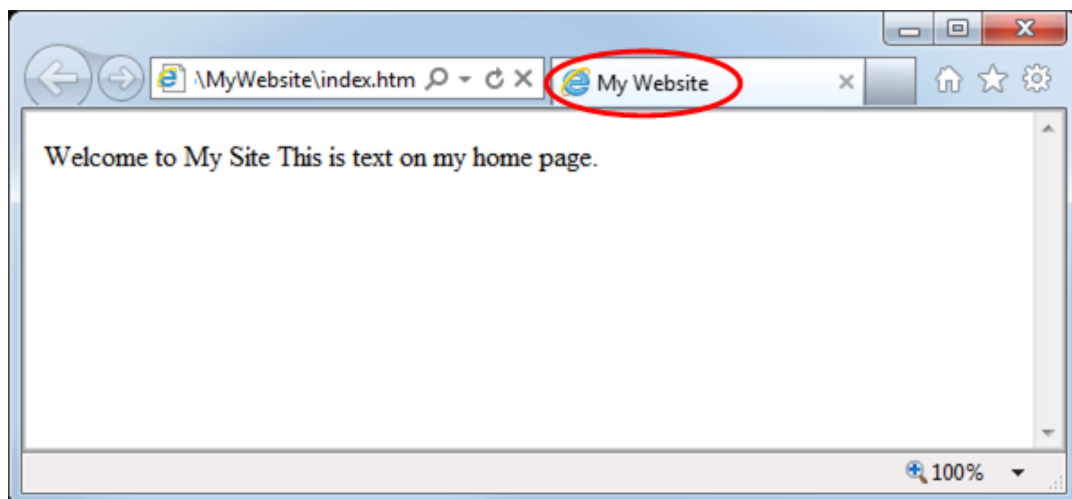
Any changes you make to a web page aren't saved until you save them. That part is easy. In most editors, you can just press CTRL + S (in Windows) or COMMAND + S (on a Mac). Or choose **File > Save** from the program's menu bar. Or just close the editor and choose **Yes** or **Save** when asked if you want to save your changes. You won't be prompted to choose a folder or enter a filename again, because that only happens the very first time you save a new file. After that, any changes you make are saved to the existing file.

Also, the changes won't automatically show up in the Web browser if the page is already open in the browser. For changes to show up in the browser, you have to click the browser's Reload or Refresh button. The location and name of the button varies slightly from one browser to the next. But typically, it's a rounded arrow, and when you point to it, the tooltip at the mouse pointer shows Reload or Refresh, like in the examples below. Clicking that button forces the Web browser to fetch the latest saved copy of the file.



Refresh/Reload buttons in Internet Explorer 9 (top) and Safari (bottom)

If you don't already have the page open in a browser, no problem. Just double-click its icon like you did in the previous lesson. Either way, you should see the current version of the page in your browser. The title that you placed between the <title>...</title> tags will appear somewhere near the top of the browser window, typically in a tab if you're using a Web browser with tabbed pages. The body content will appear in the *viewport*, which is the fancy technical name for the large white area where the page content shows. The images below show the page in Internet Explorer 9 and Safari. I circled the title in red to make it easier to spot.



Sample page in Internet Explorer 9



Sample page in Safari

If things are a little out of whack in your page, perhaps you didn't place the text correctly. The title has to go between the `<title>...</title>` tags. The rest has to go between the `<body>...</body>` tags as shown earlier in this chapter.

If you're following along with no problems at all, great! You may have some questions about why things look the way they do in the browsers. But sit tight. We're not done yet. Come on over to Chapter 4 to learn a little more about how all of this works.

## Chapter 4

### Marking Up Your Text

In Chapter 3, you added some content to your page in the form of a title and some text in the body of the page. You used your editor to make those changes, because only your editor allows you to see, and edit, the source code. The browser doesn't show the code. It just shows the content. You may not have noticed yet. But the browser also doesn't render white space or line breaks from the source. Let me explain!

In the editor, you might (or might not have) indented your code using spaces. Those indents are called *whitespace* in HTML parlance. And you might have broken the text into two lines, like this:

```
<html>
  <head>
    <title>My Website</title>
  </head>
  <body>
    Welcome to My Site
    This is text on my home page.
  </body>
</html>
```

Sample home page

The truth is, the browser doesn't care if you did or didn't indent lines, or if you put text on multiple lines. It's going to show that text as one long, left-aligned line of text either way. And that's because the browser doesn't use the whitespace or line breaks in your code to determine where to break or indent lines of text. The Web browser doesn't care about any of that. The Web browser (and all other user agents that display your page) care only about *markup*. In other words, they use the HTML tags to determine how the page looks.

To illustrate, I'll introduce you to a couple of new HTML tags. The first are the heading 1 tags (h1 tags), which look like this:

```
<h1>...</h1>
```

Any text placed between those tags is considered a *level-1 heading*. Translation to plain English would be "the main title at the top," such as a story title or headline. There are h2 headings for subheadings, h3 for sub-subheadings and so forth. But let's not get ahead of ourselves. For now, the h1 tags will do.

There are tags that mark the beginning and end of each paragraph. A paragraph is any sentence or group of sentences that have some empty space above and below them. In fact, you're reading a paragraph right now. The period at the end of this sentence also marks the end of this paragraph, and when you start reading below, you'll be in a new paragraph.

Let's try these new tags out and see what effect they have on our page. Follow these steps:

1. Open index.htm in your editor. (How you do this will always be the same, as discussed under "Opening a Page for Editing" in Chapter 3. If you've forgotten how, you can review now.)

2. Type an `<h1>` tag to the left of the word *Welcome*, and `</h1>` after *Site*.
3. Type a `<p>` tag in front of the word *This*, and a `</p>` after the period at the end of that sentence. Use the image below as a guide. But again, the red just shows you what to type where; it won't actually be red in your editor.

```
<html>
  <head>
    <title>My Website</title>
  </head>
  <body>
    <h1>Welcome to My Site</h1>
    <p>This is text on my home page.</p>
  </body>
</html>
```

Add tags, shown in red, to your page

Double-check to make sure you typed everything correctly. Save your page (press CTRL + S or COMMAND+S in your editor, or close your editor and choose **Yes** or **Save** when asked about saving your changes). Then, open the page in your browser again, or just Refresh or Reload the page in the browser if you already have it open there. The text that you marked up with the new tags should look something like this in any Web browser:

# Welcome to My Site

This is text on my home page.

Rendered page content

Of course, in the browser, you don't see the tags. You see the rendered page. The h1 text renders as larger text on its own line. The paragraph renders on its own line with some space above. They're on separate lines because headings and paragraphs are block elements, meaning that they always start on a new line. The h1 text is larger, because that's the default style for h1 elements in the browsers. Later in the course, you'll learn to style elements beyond the default. For example, you'll learn how to center the heading and change print style or color—pretty much anything you want. But for now, let's stick with the basics. The more advanced stuff will be a lot easier to understand and remember if you get the basics squared away first.

Are you game to try a couple more tags? Hope so. Follow these steps:

1. Open index.htm for editing in your editor.
2. Click just to the left of the `</p>`, and add a couple spaces after the period and the sentence *I am a sentence that contains some **boldface** text and some *italic* text*. Use the red text in the image below as a guide on where to place the text and what to type.

```
<html>
  <head>
    <title>My Website</title>
  </head>
  <body>
    <h1>Welcome to My Site</h1>
    <p>This is text on my home page. I am a sentence
that contains some <strong>boldface</strong> text and
some <em>italic</em> text.</p>
  </body>
</html>
```

Some more text and tags added to the page

Note

The new text you type may automatically wrap to two or more lines. Or it might just be one long line. It really doesn't matter because the Web browser ignores all that. So don't worry about if or where the lines break. Just worry about typing the tags correctly.



3. When you've finished typing, save the page in your text editor as you've done before.

After adding the new text and saving it, open the page in your browser again. Or, if it's already open in your Web browser, refresh or reload the page. In the browser, the text in the body will now look something like this:

# Welcome to My Site

This is text on my home page. I am a sentence that contains some **boldface** text and some *italic* text.

Some more text and tags added to the page

Your text may all be on one line though, because the text in a browser will always *word wrap* automatically to stay within the width of the browser window (so no text gets cut off). We use the term *word wrap* because it'll always try to break to the next line between two words, not in the middle of a word.

You'll also notice that the word *boldface* is darker. That's because in the source code, that word is enclosed in `<strong>...</strong>` tags. The word *italic* is slanted. That's because that word is enclosed in `<em>...</em>` tags. You may be wondering why the tag is `<strong>` rather than `<bold>` or `<boldface>` or `<b>`. That's actually because `<strong>` plays a couple of roles. In a Web browser, it makes the text look darker (boldface). It'll also cause a screen reader for the blind to say the word with some added strength when reading the page aloud. The `<em>` tag is similar. The *em* is short for *emphasis*, and it makes the text look italic on the screen, and also makes a screen reader for the blind speak the word with some emphasis when reading the page aloud.

Here's something else you might notice: Before, when you put in the `h1` and `p` tags, it separated the heading text from the paragraph with a blank line. That's because those are *block elements*, meaning they always start on a new line (start a new block of text). The `strong` and `em` tags you added didn't do that. In other words, the `<strong>` and `<em>` tags don't cause the text to start on a new line. They allow the text to continue to flow normally. `strong` and `em` are *inline elements* as opposed to block elements. Inline elements stay in line with the rest of the text on the line, rather than start a new line.

So you're off to a good start here. You have a page started. And more importantly, you've learned some key skills and terminology that will serve you well for the rest of this course—and the rest of your career if you decide to pursue that opportunity. If you still have the page open in your editor and Web browser open, feel free to close it now. Then, let's head over to Chapter 5 and wrap up the key concepts.

## Chapter 5

### Conclusion

Whew, we've covered a lot of ground here! This could be the most important lesson in the whole course though. Because many of things you learned here will apply to every single Web page you create in the future—whether just a few, or a few thousand. Let's summarize the key points:

- To create a page, open a text editor (or some program specifically designed for creating Web pages). Then, save the page, and make sure you give it an `.htm` or `.html` filename extension (they're equal, so use whichever you like).
- To view a page in your Web browser, double-click its icon. The browser shows the rendered page, which is how the rest of the world sees your page.
- To change the page, open it in a text editor again so you can see and change the source code. Typically, you right-click (Windows) or CTRL + click (Mac) the icon and choose Open With to open the page in an editor.
- Every page contains code (instructions for the computer) and content (text and pictures that people see in their Web browsers).
- Most of the code consists of HTML tags, each of which starts with `<` and ends with `>`.
- Changing the source code won't affect how the page looks in a browser immediately. You have to save the changes first. Then, if the page is already open in the browser, Refresh or Reload the page.

As you may have guessed, there are a lot more tags in HTML than the ones you learned today. And that's where the real power to design and create comes into play. But you really need to know the things described in this lesson to use *any* tags. So make sure you don't skimp over this lesson lightly.

Please remember to check your knowledge and skill by completing the assignment and taking the quiz for this lesson. Then, I'll see you in Lesson 3, where you'll add some pizzazz to your first Web page and learn some more fun and valuable stuff for creating and designing Web pages. See you there!

## Supplementary Material

---

### A Beginner's Guide to Creating Websites

<http://www.tizag.com/beginnerT/>

Click this link for another tutorial on creating your first Web page. Note that this site mentions using Notepad and copying rather than typing the tags and content. But you can use any text editor (including whatever you used in this lesson), and you can type the tags and text rather than copy them.

### HTML Introduction

[http://www.w3schools.com/html/html\\_intro.asp](http://www.w3schools.com/html/html_intro.asp)

Here's another good resource to reinforce what you learned in this lesson about HTML tags.

### HTML 4.01 / XHTML 1.0 Reference

<http://www.w3schools.com/tags/default.asp>

Here's a list of all of the tags in HTML and XHTML. The ones marked *deprecated* are being phased out of the language, so we won't be talking about those in this course. You won't want to use them in your pages either.

## FAQs

---

**Q:** Why are there two filename extensions, .htm and .html, for Web pages? Is one better than the other?

**A:** In the early days, computers filename extensions were limited to three characters. Even though those days are long gone, there are still plenty of three-character filename extensions around. And there are some longer ones too.

The Internet is *platform-independent*, which means that anybody with any computer can use it. Even ancient computers limited to three-letter filename extensions. Hence, there are two possible extensions for Web pages, .html and .htm (the shorter one is still kept around for compatibility with older computers).

All operating systems, regardless of age, define both .htm and .html as acceptable extensions for Web pages. There's no advantage or disadvantage to using one or the other. The page will look and act exactly the same, regardless of whether you use .htm or .html as your filename extension!

**Q:** Doesn't a text file need to have a .txt extension?

**A:** The .txt extension is common for text files. But a text file can have any extension, because it's not the extension that makes it a text file. It's the contents of the file that make it a text file. A text file is one that's encoded using plain ASCII text (basically, characters you can type at any keyboard on any computer). More than anything else, the program you use to create the file is what determines whether it's a text file. If you create the file with a word processing program, graphics editing program, or something else that's not specifically for creating a text file, then the file you save won't be a text file. But when you use a plain text editor like Notepad, TextEdit, or any other program that's specifically designed for creating text files, the file you save will be a text file, regardless of what filename extension you put on the file you save.

**Q:** How do I know what version of a Web browser I'm using?

**A:** In most programs, you can find the version you're using by choosing Help > About from the program's menu bar. If you don't see a menu bar, click anywhere in the program window to make sure it's active, then tap the ALT key on the keyboard. The menu bar is usually near the top of the program window with Help as the last option.

**Q:** I learned to use `<b>...</b>` tags for boldface and `<i>...</i>` for italics. What's up with the `<strong>` and `<em>` tags?

**A:** The `<b>` and `<i>` tags will work, and they're still part of HTML. But they have no effect on screen readers for the blind. It's always good to extend your horizons a bit and think about everyone who might read (or hear) your page. Also, using `<strong>` and `<em>` is courteous to blind users who can't see or read your text.

## Assignment

---

For this assignment, I'm presenting you with a hands-on challenge. I'd like you to add a paragraph to the end of the index.htm page. The sample paragraph to type is shown below. Before you add it to index.htm, keep the following in mind:

- You need to place all text in the page between the `<body>...</body>` tags.
- Every paragraph starts with a `<p>` tag and ends with a `</p>` tag.

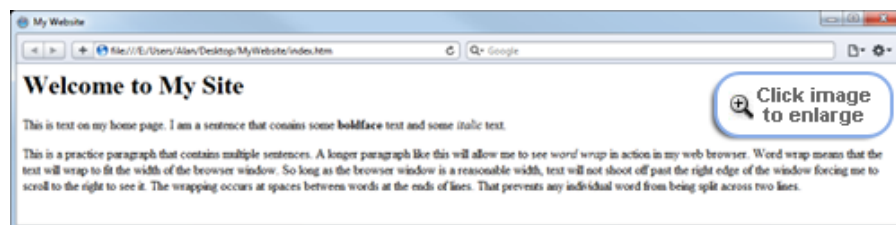
Here's the paragraph I'd like you to add to index.htm:

This is a practice paragraph that contains multiple sentences. A longer paragraph like this will allow me to see `<em>word wrap</em>` in action in my Web browser. Word wrap means that the text will wrap to fit the width of the browser window. So long as the browser window is a reasonable width, text will not shoot off past the right edge of the window forcing me to scroll to the right to see it. The wrapping occurs at spaces between words at the ends of lines. That prevents any individual word from being split across two lines.

If that's enough information for you to take a stab at it on your own, please do so. If you need step-by-step instructions, click on the icon below, which will open a new window or tab in your web browser.

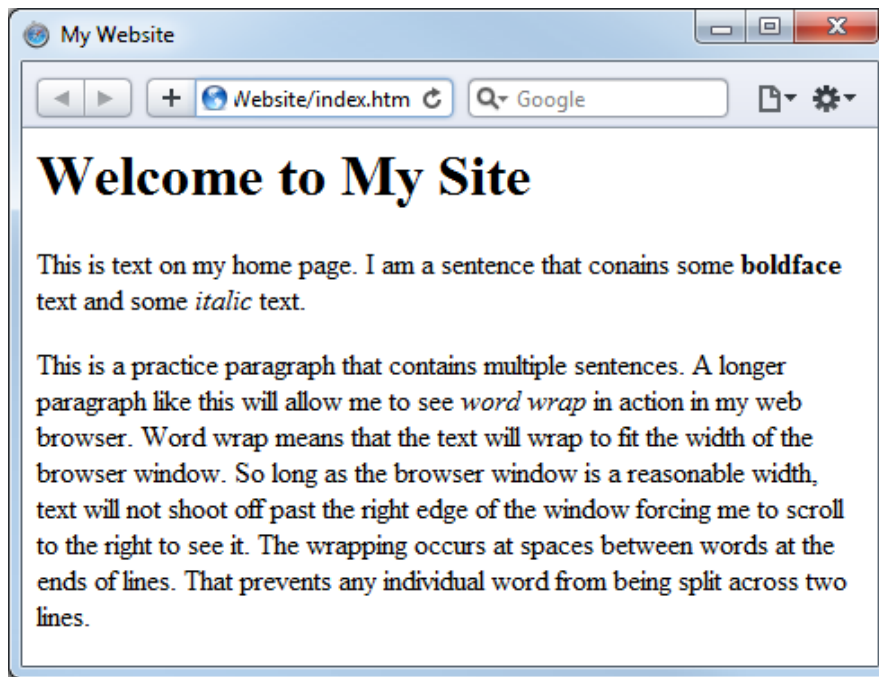


After you've added the new paragraph to the source code and save that change, view the page in your web browser to see the new paragraph. It should be the last paragraph on the page, and it should word wrap within the browser window. For example, if your browser window is wide, then the paragraph may be only two or three lines tall. Here's what that should look like:



Sample page in wide browser window

If your browser window is very narrow, then the paragraph will be narrower with more lines.



Same page in narrow browser window

If you had any difficulties with this assignment, make sure you try peeking at the instructions first. If those instructions were hard to follow, you may want to review the lesson. Of course, you're also welcome to stop by the Discussion Area to ask questions, or to help your fellow students if this assignment was easy for you.

[Back to top](#)

Copyright © 1997 - 2015 All rights reserved. The material on this site cannot be reproduced or redistributed unless you have obtained prior written permission from Cengage Learning.

web-1