Introduction to CSS3 and HTML5: Lesson 2

Alan Simpson

Lesson 2: Full-Screen View

This view has opened in a new window and will stretch to fit any screen size (large or small). It displays all of this lesson's components. To return to the normal classroom, please click the "close" button or manually close this window.

Chapter 1

Introduction

Hi, and welcome back. Today we're mainly going to focus on CSS, which stands for *Cascading Style Sheets* and is a language for styling websites and other electronic documents. The techniques you'll learn in this chapter represent some of CSS's best features and major reasons for its widespread success as a styling language for the Web and other forms of electronic publishing.

We'll start in Chapter 2 with a review of inline styles and internal style sheets. I use the term *review* because you may have already learned about them in the *Creating Web Pages* course, so they should seem familiar to those of you who completed that course. Our primary focus will be on *external* style sheets, which allow you to centralize your styles so you can apply them to multiple pages in the site. There are some huge advantages to that, as you'll discover. But first let's make sure we're all speaking the same language here by reviewing the key CSS concepts and terminology.

Chapter 2

Using CSS

As you know, a website can contain any number of Web pages (usually called *pages* for short). And any given Web page can contain many different *elements*. For example, this page you're reading right now is a Web page. It contains text organized into different kinds of elements such as headings, paragraphs, lists, and pictures (images).

In a Web page, you use HTML tags to organize the text into different element types (headings, paragraphs, lists, tables, and so forth). You use CSS to define the *style* of those element types. *Style* refers to things like colors, fonts, size, and other artistic elements that define the look-and-feel of the page.

One advantage to using CSS over other styling methods is that you can easily control the *scope* of your styling. In other words, you can control how far-reaching a style is by where you put the CSS code. There are basically three different ways to use CSS that determine how far-reaching a style is:

- **Inline style:** Applies to a single element on a single page.
- **Internal style rule:** Can apply to multiple elements within a single page.
- **External style rule:** Can apply to multiple elements on multiple pages within the site.

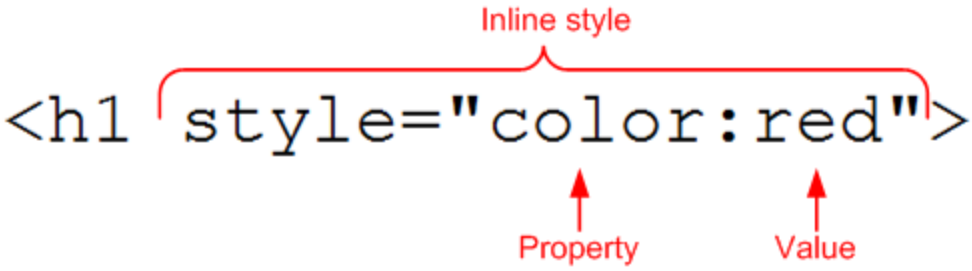
Let's start with the inline style.

Creating Inline Styles

An *inline* style gets its name from the fact that it goes right inside (*inline* with) the tag to which you want to apply the styling. You use a **style**= attribute in the tag to indicate you're about to apply CSS styling to the element. Inside the quotation marks, you list one or more CSS *property*:*value* pairs to define styles. As in HTML, you always type your CSS code in lowercase letters.

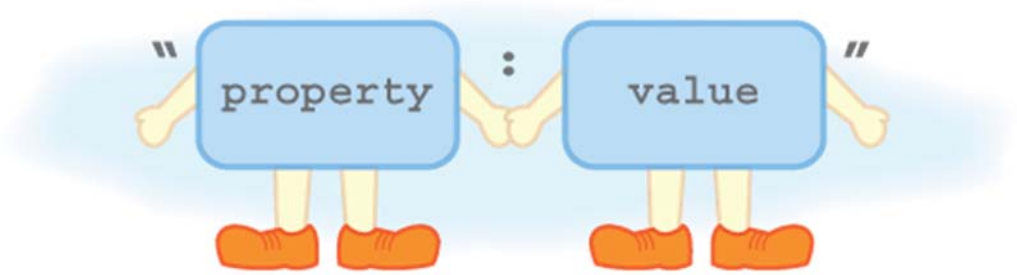
For example, below is an <h1> tag that contains an inline style that sets the color of the text to red.

Show Text Equivalent



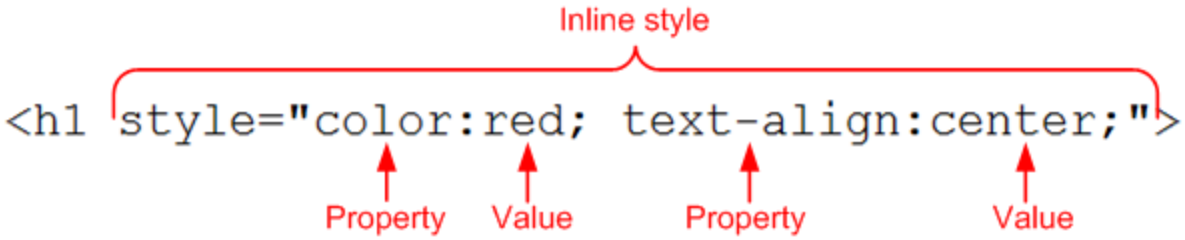
Sample tag with an inline style

This example shows the opening tag for an h1 element. That tag contains an inline style. The inline style contains one property:value pair.



You can put multiple property:value pairs in the quotation marks as long as you separate each pair with a semicolon (;). For example, below you'll see an inline style with two CSS property:value pairs separated by a semicolon (;).

Show Text Equivalent



Inline style with two *property*:*value* pairs

You can put spaces around punctuation marks (=, :, ;) if you like, but they're not required. You can break the line after a punctuation mark, too, if that makes the code seem more readable to you. For example, you could write that last tag like this:

```
<h1 style="
  color:red;
  text-align:center;"
>
```

You might notice there's a semicolon after the last property:value pair above. Technically, the semicolon is only required between property:value pairs. The very last one is optional. But people will often type it in so that, should they decide to add another property:value pair in the future, they won't forget to type that one.

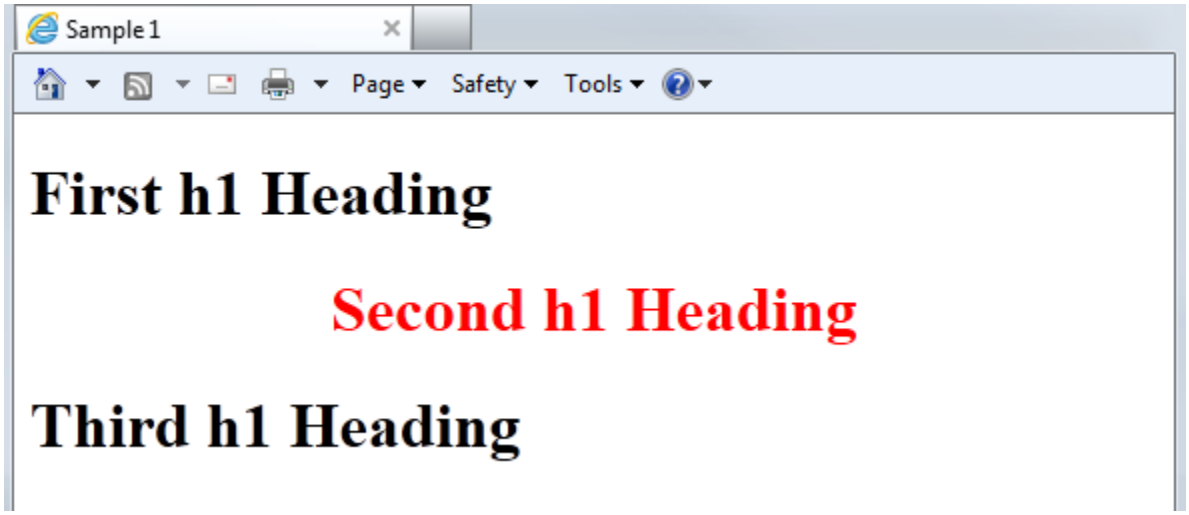
Those examples show an opening tag with an inline style. In real life, an h1 heading will have a closing tag as well, plus some text in between. You don't have to do anything special with the closing tag. You'd apply the inline style to the text between the opening and closing tags.

To illustrate the concept, take a look at a page containing three h1 elements. I've boldfaced those tags to make them stand out. Keep in mind that we're just using this example to illustrate a concept. It's not something you need to type right now, and it's not something you'd probably want to use in a real website. It's just a sample page I'm using to help you understand inline styles.

```
<!DOCTYPE html>
<html>
<head>
    <title>Sample 1</title>
</head>
<body>

<h1>First h1 Heading</h1>

<h1 style="color:red;text-align:center;">
Second h1 Heading
</h1>

<h1>Third h1 Heading</h1>

</body>
</html>
```

The first and third h1 elements use no CSS styling. The middle h1 style has an inline style. Here's a screen shot of how that page would look in a Web browser.

Show Text Equivalent



Sample page in a Web browser

Notice the three h1 headings. The first and last ones are large, black, bold, left-aligned, and have quite a bit of margin space above and below. They look that way because that's the *default style* for h1 headings. In other words, it's the styling you get for h1 headings (text between <h1>...</h1> tags) if you don't do any styling of your own.

The middle h1 heading is similar to the other two except that its text is red and it's centered. That's because only the middle h1 has the inline style that reads style="color:red;text-align:center". So the text is red instead of black, and it's centered horizontally in the browser window, rather than left-aligned. That's because the inline style takes precedence over the default black color and left alignment.

The middle heading is still large, bold text with quite a bit of margin space above and below it. That's because our inline style didn't say anything about changing the size or boldface or margins. So those characteristics from the default styling still come through. Of course, there are lots of properties and lots of values in CSS that you could use to change the font, size, margins, and more. But for now, let's stay focused on inline styles in general.

As you can see, the scope of the inline style is limited to the tag that contains the inline style. The other h1 tags on the page aren't affected by the inline style at all. That's because the scope of an inline style is just the tag to which it's applied. Other elements on the same page and other elements on other pages in the site aren't affected by the inline style. But CSS is a very flexible language, and there are ways you can style multiple elements on a page, or even multiple elements on multiple pages, if you want to.

You don't use inline styles for that kind of styling though. For that broader kind of styling, you use *style rules*. Follow me over to Chapter 3 to learn what those are about.

Chapter 3

Creating Style Rules

An inline style, like you learned about in Chapter 2, is called *inline* because it's typed inside (*inline* with) a tag. There's another way to type CSS—it's known as a *style rule*, and you'll learn about it in this chapter. Unlike an inline style that applies only to the tag in which it's typed, a style rule is like a "rule of styling" that you can apply to multiple elements, not just one element at a time.

The syntax for style rules is a little different from that of inline styles. Every style rule must start with a *selector* that selects (or defines) the elements to which the styling will be applied. The selector is followed by a pair of curly braces {}, and inside those braces, you type one or more property:value pairs separated by semicolons—something like this

```
selector{
    property:value;
}
```

As with inline styles, you're not limited to one property:value pair per style rule. You can use as many as you like. As with an inline style, you must remember to separate each property:value pair with a semicolon, as I've done below.

```
selector{
    property:value;
    property:value;
    property:value;
}
```

As with inline styles, you have some leeway in terms of line breaks and spaces. Spaces around punctuation marks are optional. You can break a line (or not break a line) after the semicolon or braces. The very last semicolon is optional. So the syntax below is an acceptable alternative to the example above.

```
selector{property:value; property:value; property:value;}
```

Note

The *property*:*value* pairs of an inline style or style rule are sometimes referred to as the *descriptor* because, collectively, they describe how the element is to be styled.

Unlike inline styles, style rules have a broad scope that can apply to multiple elements. Exactly how far-reaching that scope is depends on where you put the style rule. There are two places where you can type style rules: internal style sheets and external style sheets. Here's the difference:

- **Internal style sheet:** Style rules apply only to the page that contains the style rule.
- **External style sheet:** Style rules can apply to multiple pages in the website.

Let's start with an internal style sheet. You already had some experience with those in *Creating Web Pages*. So I'll just review the basics here.

Creating an Internal Style Sheet

An internal style sheet is inside (internal to) a Web page. Some people call it an *embedded style sheet*. It doesn't matter what you call it, but how and where you type it matters a lot.

You must place the internal style sheet in the page header. In other words, the internal style sheet must go between the <head> and </head> tags of the page. To mark the beginning of an internal style sheet, use this tag:

```
<style type="text/css">
```

To mark the end of the internal style sheet, use this tag:

```
</style>
```

Between those tags, you can place CSS style rules.

The selector that you put on the style rule determines what items on the page the style applies to. There are many types of selectors you can use, as you'll learn throughout this course. As a working example (and this isn't something you need to type right now), I've typed an internal style sheet into the sample page below and boldfaced its code below to make it stand out from the rest.

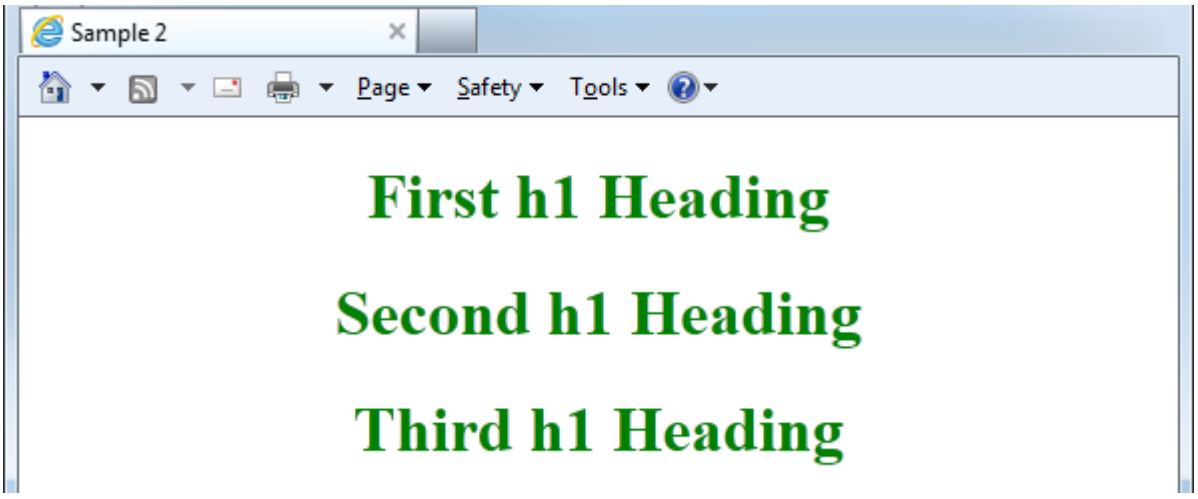
```
<!DOCTYPE html>
<html>
<head>
```

```
   <title>Sample 3</title>
   <style type="text/css">
   h1 {
       color: green;
       text-align: center;
   }
   </style>
</head>
<body>
  <h1>First h1 Heading</h1>
  <h1 >Second h1 Heading</h1>
  <h1>Third h1 Heading</h1>
</body>
</html>
```

That internal style sheet starts with the <**style type="text/css"**> tag and ends with </**style**>. That internal style sheet contains one style rule that looks like this:

```
h1 {
   color:green;
   text-align:center;
}
```

The h1 is the selector, and it says, "Apply this styling to all h1 elements on this page." The styling to be applied is defined by the *property*:*value* pairs inside the curly braces of the style rule. Note that the rest of the tags on that sample page are just the mandatory HTML5 tags that all Web pages must contain, plus three sample h1 headings, none of which includes an inline style. Viewing that sample page in a browser would produce a result like this:



Internal style sheet's style rule applies to all h1 headings on the page

Each heading is large, bold text with quite a bit of margin space between, because that's how h1 headings look in a browser, even without any styling. The green text color and centering come from the internal style rule. I removed the previous inline style that was making the middle heading red, so that inline style is no longer in play in the example above. But it's worth discussing what would happen if we did put in an inline style.

When Styles Conflict

Sometimes, style rules can conflict. For example, suppose I put an inline style back into one of the h1 tags. I'll use an inline style to make the middle style rule blue and left-aligned. You'll notice that change by the boldfaced text in the image below. (Again, you don't need to follow along by typing all of this. For now it's sufficient to focus on the terminology and concepts.)

```
<!DOCTYPE html>
<html>
<head>
   <title>Sample 3</title>
   <style type="text/css">
   h1 {
       color: green;
       text-align: center;
   }
   </style>
</head>
<body>
  <h1>First h1 Heading</h1>
  <h1 style="color: blue; text-align: left;">
       Second h1 Heading
  </h1>
  <h1>Third h1 Heading</h1>

</body>
</html>
```

Do you see the conflict? On the one hand, near the top of the page, there's a style rule that says, "All h1 headings on this page are to be green and centered." But then, in the second h1 heading, there's a style rule that says, "This h1 heading it to be blue and left-aligned." What's a Web browser to do? Whenever these kinds of conflicts arise, the Web browser always handles them in a simple straightforward manner. It's so basic, but it's so fundamental to understanding how CSS works, it deserves its own paragraph:

When styles conflict, the one that's closest to the tag being styled wins.

An inline style is actually *in* the tag of the element being styled. And being *in* the tag is as close to the tag as you can possibly get. Therefore, an inline style always takes precedence over a style rule.

Show Text Equivalent

If you were to view the sample page above in a browser, the middle h1 heading would be blue and left-aligned as you see below. The other h1 headings are unaffected by that styling because an inline's style scope is just the tag that contains the inline style.

Inline style affects only the tag that contains the inline style

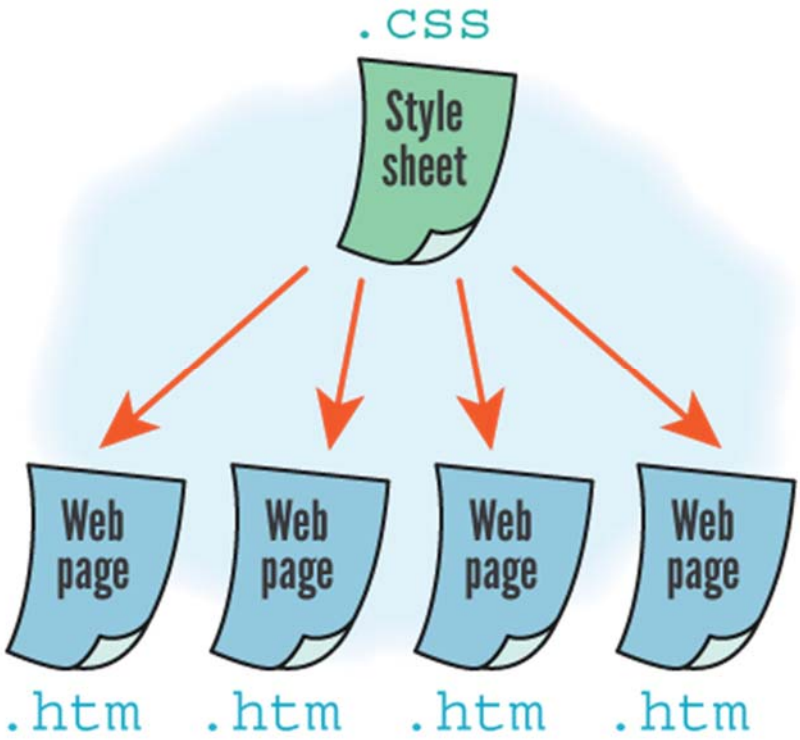
Turns out that this is a good way to handle conflicts, because sometimes in real life, you may define a style rule that says, in general, how you want a certain type of element to look in all your pages. But in a large website, there may be the occasional oddball where you want a little different style. When that happens, you can just use an inline style to change the styling for that one element without messing up the styling for all the other elements.

So that was an example of an internal style rule that applies to all h1 headings on the page that contains the internal style rule. Now, suppose I want to make a style rule that styles all the h1 headings on *all* the pages in the site, not just one page. When you want to define style rules that apply to more than one page in a site, you use an *external* style sheet, which you'll learn about starting in Chapter 4. See you there!

Chapter 4

Using External Style Sheets

As you've seen, inline styles come in handy when you want to style a single element on a single page. Internal style sheets are handy for styling multiple elements on a single page. But if you're looking for stylistic consistency across all the pages in your site, an external style sheet is the way to go. Style rules in an external style sheet can apply to as many pages in your website as you wish. The image below illustrates the concept, where a single external style sheet can define styles for multiple pages. (There's no limit to the number of pages to which you can apply a style sheet.)



One style sheet can define styles for many pages

Using styles consistently across pages in your site gives it a consistent look-and-feel. That's important because it helps visitors know when they're looking at your site and when they've navigated away to another site. This is one of the main reasons why virtually all modern websites use external style sheets.

Putting styles rules in an external file also makes it easier to make site-wide stylistic changes. For example, let's say you used an external style sheet to define your styles when you built your site. After a few months or years, you decide to give the site a stylistic facelift. With an external style sheet, all you have to do is open that one style sheet, make your changes there, and you're done. Every page in the site that uses the style sheet is automatically and instantly updated for you, so your work is done. You don't have to update each individual page one at a time. (That's a really big job on a large site that contains hundreds of pages!)

So what is an external style sheet, exactly? Well, it's a text file that you create with a text editor—the same editor you use to create your Web pages. The style sheet doesn't contain any HTML tags. Instead it contains only CSS style rules. And when you save it, you don't give the filename a .htm or .html extension. Instead, you give the filename a .css extension to indicate the file is a CSS style sheet rather than a Web page. You want to save your style sheet in the same folder as the rest of the files that make up your site. In this course, that's the *Intro HTML5* folder. Let's go through the steps.

1. Open your text editor (Notepad, TextEdit, or whatever it is you normally use to create Web pages).

2. Type (or copy and paste) in the following style rule.

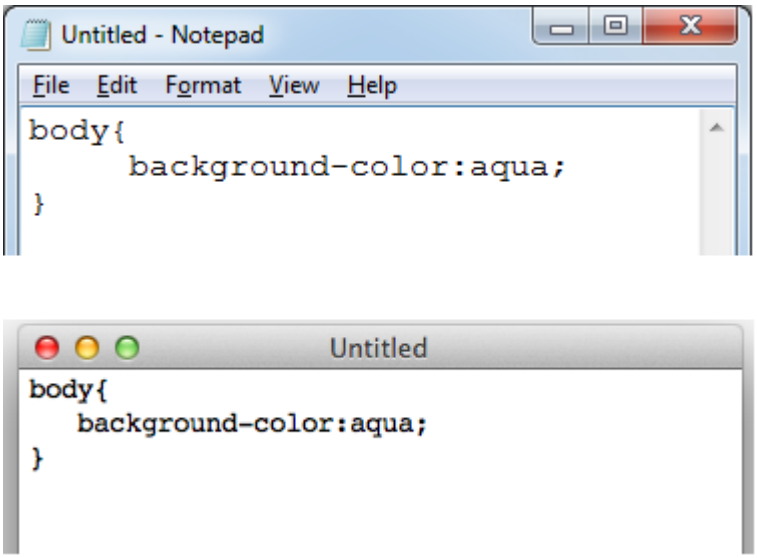
```
3. body{
```

```
4.     background-color:aqua;
}
```

Caution

Notice that *aqua* is spelled with a lowercase *Q*, no letter *G*. As with all computer code, it won't work right if you spell it wrong. You have to spell it as *aqua* with the letter *q* and no *g*.

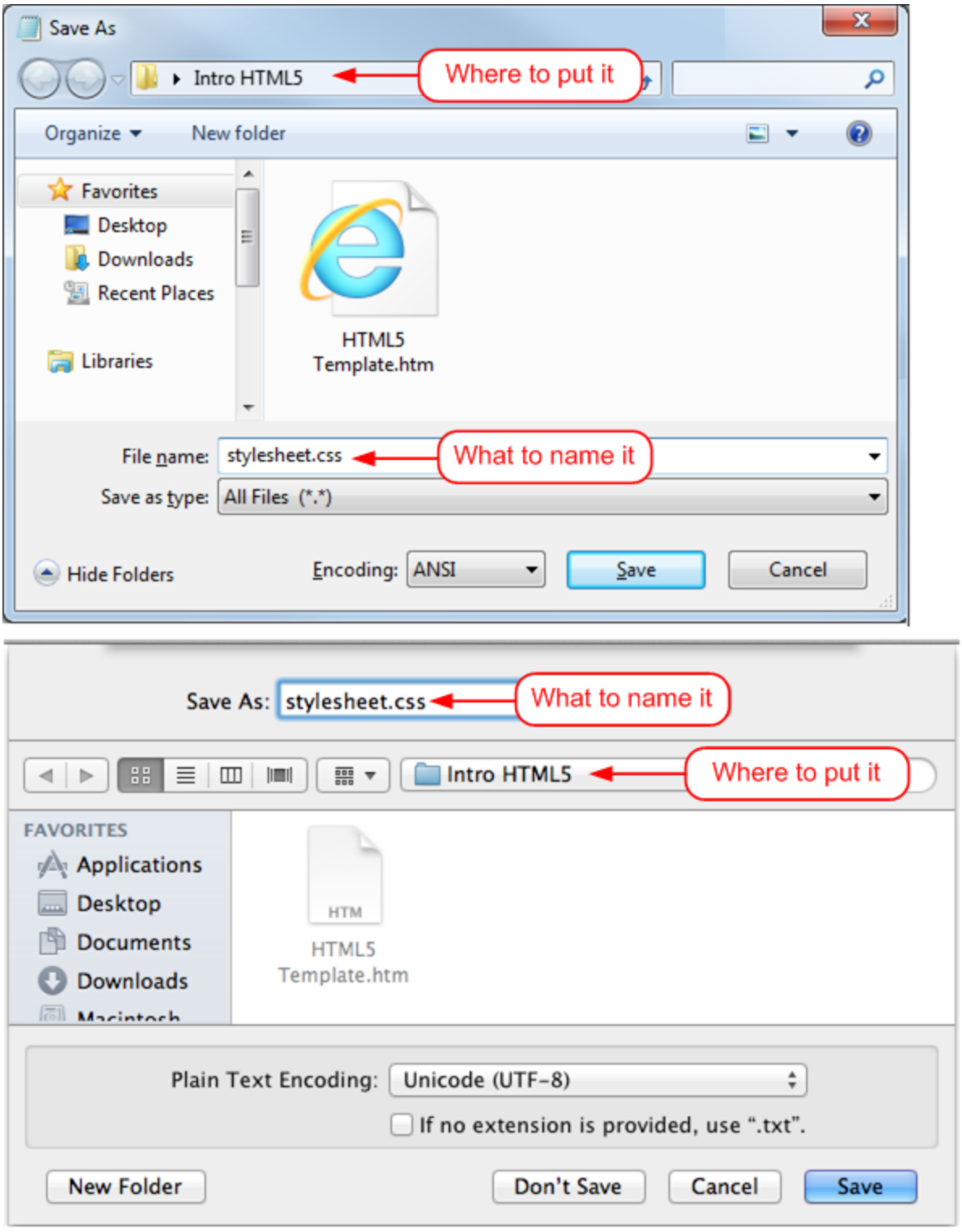
It's not necessary to indent the line between the curly braces. But it does make the code a little more readable if you do it that way, so you can just press the SPACEBAR a few times at the start of that line to indent. Here's how the style rule looks in Notepad and TextEdit:



Style sheet in Notepad (top) and TextEdit (bottom)

5. Verify that you typed your code correctly, and then close your editor (click the X in the upper-right corner of Notepad's window or the red circle in the upper-left corner of TextEdit's window). If you're using Notepad, choose **Save** when asked if you want to save your pages.

6. Make sure you choose **Intro HTML5** as the folder in which to save the file, and type *stylesheet.css* as the filename as you see below (the top image is for Notepad in Windows, the bottom one for TextEdit in Mac OSX.)

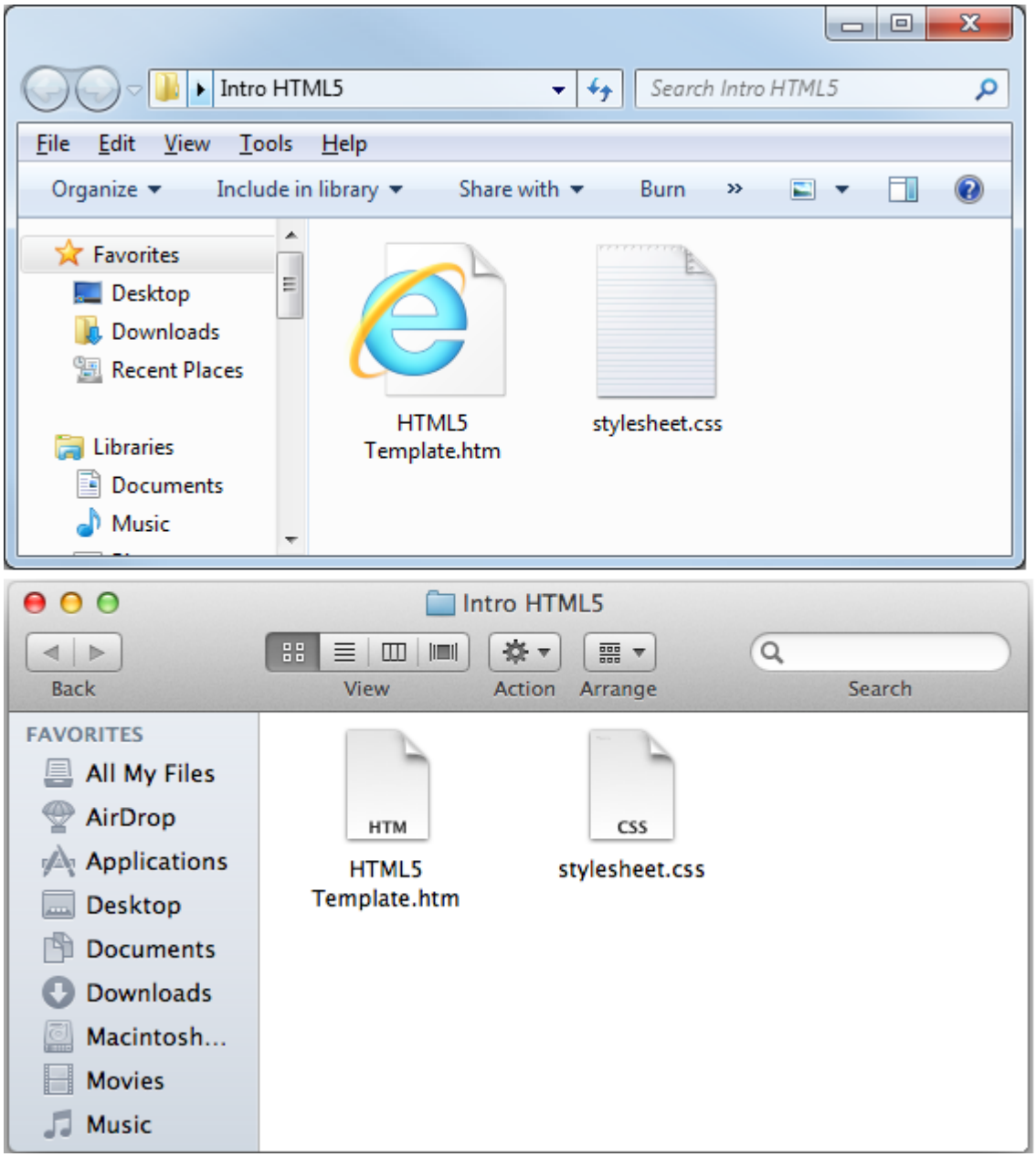


Save stylesheet.css in your Intro HTML5 folder

7. Click **Save**.

The style sheet is saved in the folder you specified with the filename you specified, as usual. Let's verify that the file is in the right folder with the right filename. Because in Web development, it's always

important to know the *exact* names and locations of all the files that make up the site. Guessing, assuming, and hoping rarely work. So open your *Intro HTML5* folder, and verify that it contains the stylesheet.css file as well as the HTML5 Template.htm page you created in Lesson 1. You can use the images below as a guide in Windows (top) and a Mac (bottom). As always, the exact appearance of your icons may vary depending on your operating system, default Web browser, and current icon view. The important thing for now is that the style sheet is named exactly *stylesheet.css* with no spaces and no other letters after the *css* (no extra *.txt* or anything tacked onto the end).



Intro HTML5 folder (Windows 7 on the top, Mac OS on the bottom)

Speaking of guessing and assuming, before you start assuming that you'll see an aqua-colored background on any Web pages, let me explain why that's not going to happen yet. It's because the style

rules in an external style sheet don't automatically apply to any pages in your site. Instead, style rules in an external style sheet apply only to pages that *link* to the style sheet. So let's talk about that.

Linking to an External Style Sheet

To link a page to an external style sheet, you need to put a link tag between the <head> and </head> tags in the page. The syntax of that tag is as follows:

```
<link href="filename" rel="stylesheet" type="text/css" />
```

Sample code like that is often referred to as a *syntax chart* by programmers, and any *italic* text indicates a placeholder for information that you must provide yourself when typing your own code. Notice that only the word *filename* is italicized in the above tag syntax. That's because when you're typing your own code, you must replace *filename* with the complete filename of your style sheet. You'll always type the rest of the tag exactly as you see it here. The **rel=** attribute describes how the linked file is related to the style sheet, and that's always the word *stylesheet* in quotation marks (no matter what the filename of the style sheet). The **type=** attribute should always have *text/css* as its value because a style sheet is always a text file that contains CSS code.

Right now, the only Web page we have in our Intro HTML5 folder is the HTML5 Template.htm page, which contains all of the mandatory HTML5 tags that every HTML5 page must contain. But that's a good place to put the link tag to the style sheet. Because typically, you want all the pages in your website to link to the same style sheet, so they can share the same style rules. So let's get started. Here are the steps:

1. If you haven't already done so, open your Intro HTLM5 folder.

2. Open HTML5 Template.htm in your editor (in Windows, right-click its icon and choose **Open With** > **Notepad**. In Mac OS, CTRL + Click its icon and choose **Open With** > **TextEdit**).

3. Put the cursor just past the closing </title> tag, and press ENTER.

4. Type or copy and paste the link tag below into the page, inside the <head>...</head> tags.

   ```
   <link href="stylesheet.css" rel="stylesheet" type="text/css" />
   ```

   Here's how the page should look with the new tag in place:

   ```
   <!DOCTYPE html>
   <html>


   <head>
       <title></title>
       <link href="stylesheet.css" rel="stylesheet" type="text/css" />
   </head>


   <body>
   ```

```
</body>


</html>
```

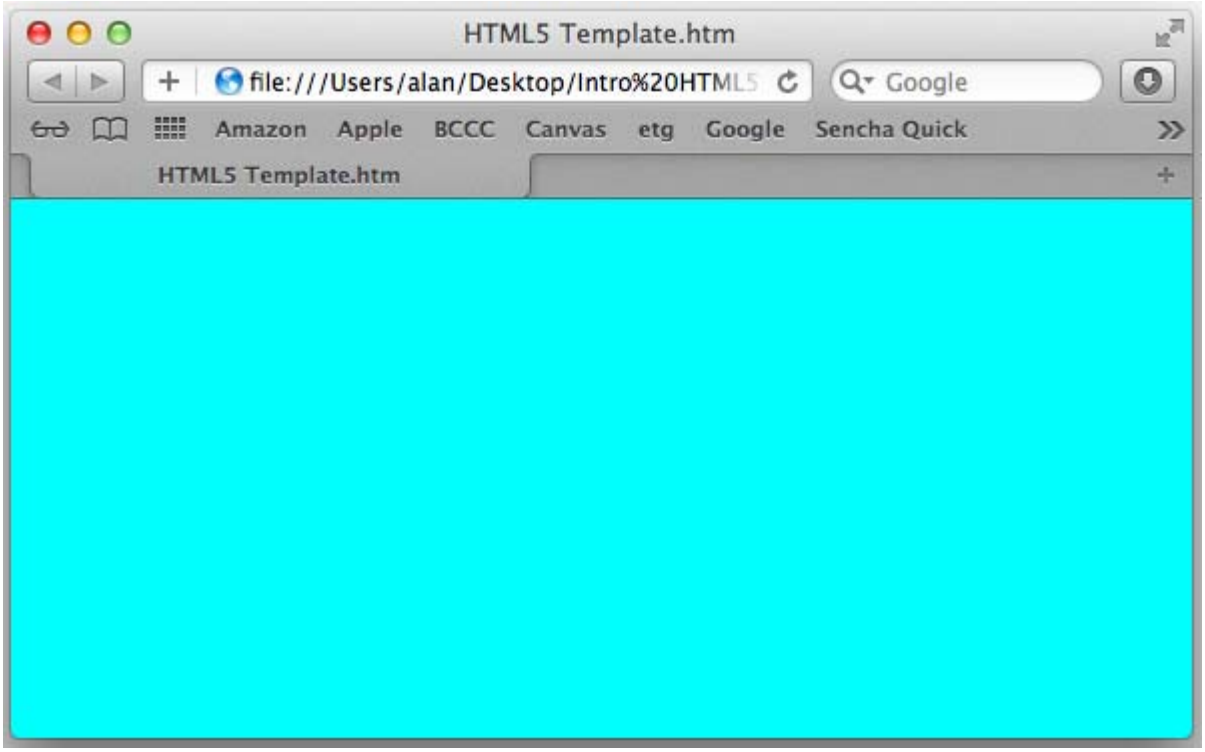
5. Close the HTML5 Template.htm page (click the Close (X) button in the upper-left or right corner of your editor. Choose **Yes** or **Save** if asked about saving your changes.

The HTML5 Template.htm page doesn't have any content yet (no text or pictures that show in the Web browser). But it does have a body element, as defined by its <body>...</body> tags. And in our external style sheet, we created a style rule that sets the background color of the body element to aqua. So let's open HTML5 Template.htm in a Web browser now and see if the style rule works.

1. If necessary, open your Intro HTML5 folder.

2. Double-click the HTML5 Template.htm file's icon to open it in your default Web browser.

If you did everything correctly, you should see an empty page with an aqua-colored background as in the image below.



HTML5 Template.htm file with new background color

If yours doesn't work, it may be a simple typographical error somewhere in your code or a filename. I will provide some style sheet troubleshooting steps in the Lesson 2 FAQs, so feel free to take a look at those when convenient. Come on over to Chapter 5 now, and we'll wrap up what you've learned in this lesson.

Chapter 5

Conclusion

In this lesson, you learned the role the CSS plays in developing a website and different ways you can use it to style your site. The key thing to remember is that, when it comes to defining stylistics things for your site like colors, fonts, backgrounds, and so forth, CSS is the language you'll use. (It's not called Cascading *Style* Sheets for nothin'!)

You can use CSS inline styles to style one element in a page at a time. An inline style is one you actually type in the tag you want to style, enclosing all the CSS code in a **style=** attribute within the tag. But styling one element at a time is a slow, tedious way to develop a large site. It makes it difficult to maintain a consistent look-and-feel across multiple pages and also makes it difficult to make site-wide stylistic changes in the future. So CSS offers style rules as another way to style your site. With style rules, you can define styles that apply to multiple elements on a page.

You never type CSS style rules in an HTML tag. Instead, you type CSS style rules in *style sheets*. There are two types of style sheets you can create:

- **Internal style sheet:** Contains style rules that you can apply to one or more elements on one page (the page that contains the style sheet).

- **External style sheet:** Contains style rules that you can apply to multiple pages in the site.

Virtually all modern websites rely on external styles sheets for most of their styling, and so those will be our main focus throughout this course. The two main things you need to remember about an external style sheet are:

- An external style sheet is a file you put in your website root folder, but unlike a Web page, its filename has a *.css* extension, and the file contains only CSS style rules, no HTML tags.

- An external style sheet's style rules apply *only* to Web pages that *link* to the style sheet using an HTML <link> tag in the header of the page.

If you can remember those two key things about external style sheets, you're well on your way to the creative freedom and power that allow the professionals to create all of the really great websites you've probably seen in your day-to-day Web surfing.

In the next lesson, you'll learn some cool techniques for using CSS and HTML to design cool backgrounds for your pages and other elements. See you there!

Supplementary Material

CSS Beginner Tutorial

http://htmldog.com/guides/cssbeginner/

Click this link for a good tutorial that reviews the basics and also offers a good overview of techniques we'll discuss in this course.

CSS Introduction

http://www.w3schools.com/css/css_syntax.asp

Here's another page that provides a good introduction to CSS to help reinforce what you've learned in these first two lessons. It will also give you an advanced peek at things to come.

Learn HTML and CSS: An Absolute Beginner's Guide Article

http://www.sitepoint.com/html-css-beginners-guide/

Here's another good resource for reviewing the basics and beyond.

Why Use CSS?

https://developer.mozilla.org/en/CSS/Getting_Started/Why_use_CSS%3F

As the title suggests, this page from the folks who brought you the Firefox browser explain why we have CSS, why everyone uses, it, and why you want to learn it.

Three Ways to Insert CSS

http://www.w3schools.com/css/css_howto.asp

Use this page to review the three ways you can use CSS, inline style, internal style sheet, and external style sheet.

Dreamweaver: Link to an External CSS Style Sheet

http://help.adobe.com/en_US/dreamweaver/cs/using/WScbb6b82af5544594822510a94ae8d65-7e1ca.html

This isn't a Dreamweaver course, so you can ignore this one if you don't own or know how to use Dreamweaver. I just put it here for the folks who are using Dreamweaver.

FAQs

Q. How can I tell a page from a style sheet?

A. A *page* is a Web page—a file that has a *.htm* or *.html* extension. The file contains HTML tags and usually content (text and pictures for people to read and look at). When you double-click the icon for a page, it should open in your default browser. To open it in some other program, right-click (or CTRL+ Click on a Mac) its icon, choose **Open With** and then the program you want to use to open it.

A style sheet is a file that usually has a .css extension. It contains no HTML tags and no content. It contains only CSS style, which defines how content is to be styled. You can't view a style sheet in a Web browser since it contains no content. When you double-click the icon for a style sheet, it opens in whatever program is currently set as the default for .css files on your system (if any). As with any file, you

can right-click (or CTRL + Click) the style sheet's file icon, choose **Open With** and the name of the program you want to use to edit it.

Q. None of the styles from my style sheet are showing up in my page. How can I troubleshoot?

A. The most common error is a bad or missing link. A successful link to the style sheet requires that you know the exact location and filename of the style sheet to which you're linking, the location of the page from which you're linking, and a <link> tag with the correct syntax. So let's start at the top:

1. Make sure filename extensions are visible, as per Lesson 1, Chapter 2.

2. Open your *Intro HTML5* folder.

3. Make sure your style sheet is in that folder and you know its exact spelling (stylesheet.css for our working example, with no spaces, and make sure there's no extra *.txt* or something after the .css extension).

4. Make sure the page that you're linking to the style sheet is in the same folder as the style sheet.

5. Open for editing the page that links to the style sheet.

6. Verify that the link tag in that page is between the <head> and </head> tags of the page.

7. Verify that you typed your link tag exactly as below. If your style sheet filename isn't *stylesheet.css*, change the href= value to match your exact style sheet filename. Never change the values for the rel= or type= attributes. Make sure the last value is text/css and not text\css or text-css (two common mistakes).

   ```
   <link href="stylesheet.css" rel="stylesheet" type="text/css" />
   ```

8. Close and save the page.

9. If you have to make any changes, try opening your page in a browser again (double-click the page icon).

If you still can't get it to work, there may be a typographical error in your code somewhere, or you may not have used a text editor to create the page. Try opening the page in a text editor (using the standard right-click or CTRL + Click and Open With method). For this lesson, stylesheet.css should contain exactly this code.

```
body{
   background-color:aqua;
}
```

HTML5 Template.htm should contain exactly this code.

```
<!DOCTYPE html>
<html>
```

```
<head>
    <title></title>
    <link href="stylesheet.css" rel="stylesheet" type="text/css" />
</head>

<body>
</body>

</html>
```

Once you have everything right, the styling from the style sheet should show up in the Web page. If it still doesn't, close the browser completely, and then reopen the page in a browser. If you can't find what's wrong, you can copy and paste the entire style sheet and page to a Discussion Area post and we'll all take a look.

Q. How do I a rename a file in Mac OS?

A. Follow these steps:

1. Click the name of the file you want to change.
2. Wait a moment, then click again right in the name of the file that you want to change.
3. Type the new name, and press ENTER.

Q. How do I rename a file in Windows?

A. Follow these steps:

1. Right-click the icon of the file you want to rename, and choose **Rename**.
2. Type the new name, and press ENTER.

As an alternative to using the right-click method, you can click the file's icon once, wait a moment (so you don't double-click), and then click once on the filename. Type the new name, and press ENTER.

Q. If I have an external style sheet, an internal style sheet, and inline styles, and there's a conflict, which style wins.

A. The style rule that's closest to the tag being styled always wins. So if the link tag comes before the internal style sheet in a page, like this:

```
<head>
```

```
    <title></title>
    <link href="stylesheet.css" rel="stylesheet" type="text/css" />

    <style type="text/css">
    ...
    </style>

</head>

<body>
</body>
```

. . . then the styles inside the internal style sheet (where you see ... above) are closer to the tags (which start at the <body> tag); and therefore the internal style sheet's style rules would take precedence. But an inline style would still trump any style rule, because an inline style is *in* the tag, and that's as close to the tag as you can possibly get!

Assignment

I'm not going to ask you to do any extra work for this assignment, because if your style sheet and page are working, you're in a perfect place for moving forward. And by "working" I just mean that when you open HTML5 Template.htm in a Web browser, it has the light blue aqua-colored background. So your assignment today is to make sure your page is working.

If yours isn't, please spend more time on it and ensure it works before moving on to the next lesson. You can use the Lesson 2 FAQs for some troubleshooting tips (click **Resources** near the top or bottom of any page in the course, click **Frequently Asked Questions**, and then click **Lesson 2**).

If you're feeling uncomfortable with any of the terminology we've used so far, I would strongly suggest you study Lessons 1 and 2 some more. I realize much of what we've covered here in Lessons 1 and 2 is review from the *Creating Web Pages* course. But going forward, there won't be much in terms of review. You'll need to really know and understand what we've covered so far in this course. So if you have any doubts at all, now would be a good time to do a bit more studying. You'll get plenty of hands-on practice in upcoming lessons.