Creating Web Pages (HTML): Lesson 7

CLOSE

Alan Simpson

Lesson 7: 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

Welcome back! In today's lesson, we'll explore some more cool ways to style your website using modern CSS techniques. As you may recall from Lesson 6, CSS isn't something you use *instead of* HTML. Rather, you use CSS in conjunction with HTML to create websites.

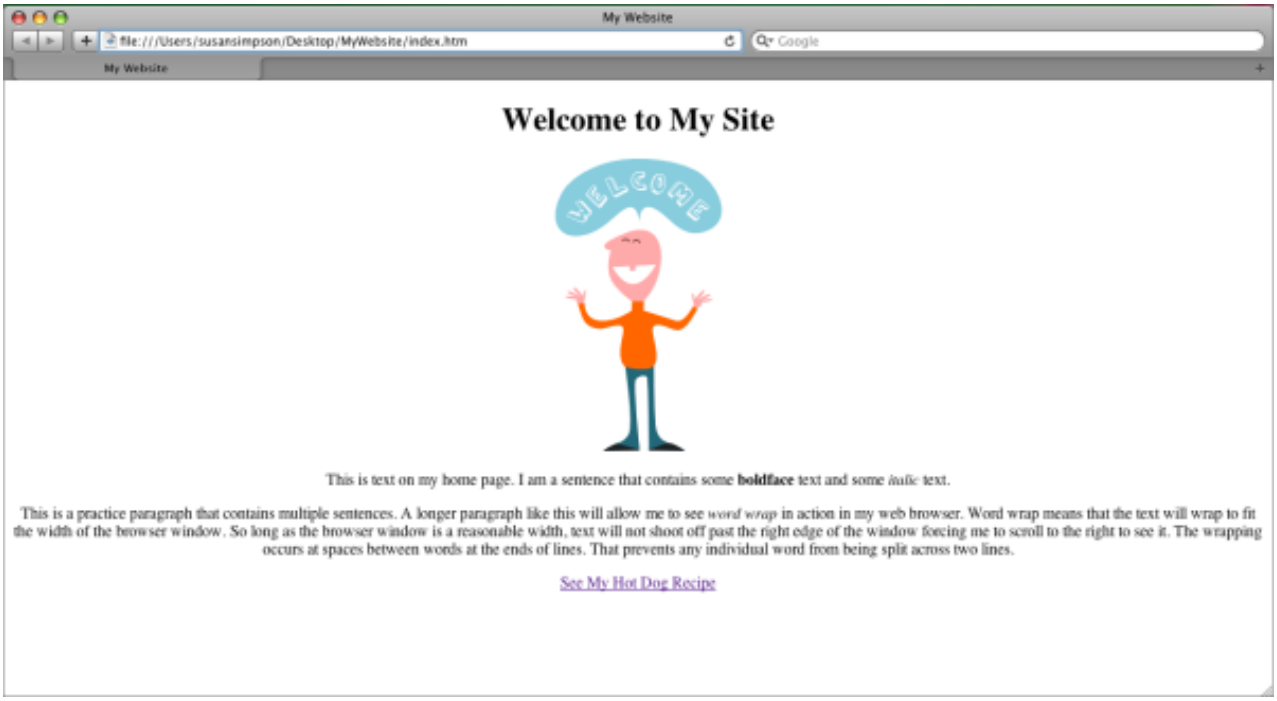
In Lesson 6, you discovered CSS inline styles, where the CSS is right inside an HTML tag, in the quotation marks of a style=" " attribute. In today's lesson, we'll talk about how CSS handles conflicts, and you'll learn a new way to apply CSS through style rules, and how to add color to your pages with some CSS properties.

So let's jump right into it in Chapter 2. I'll see you there!

Chapter 2

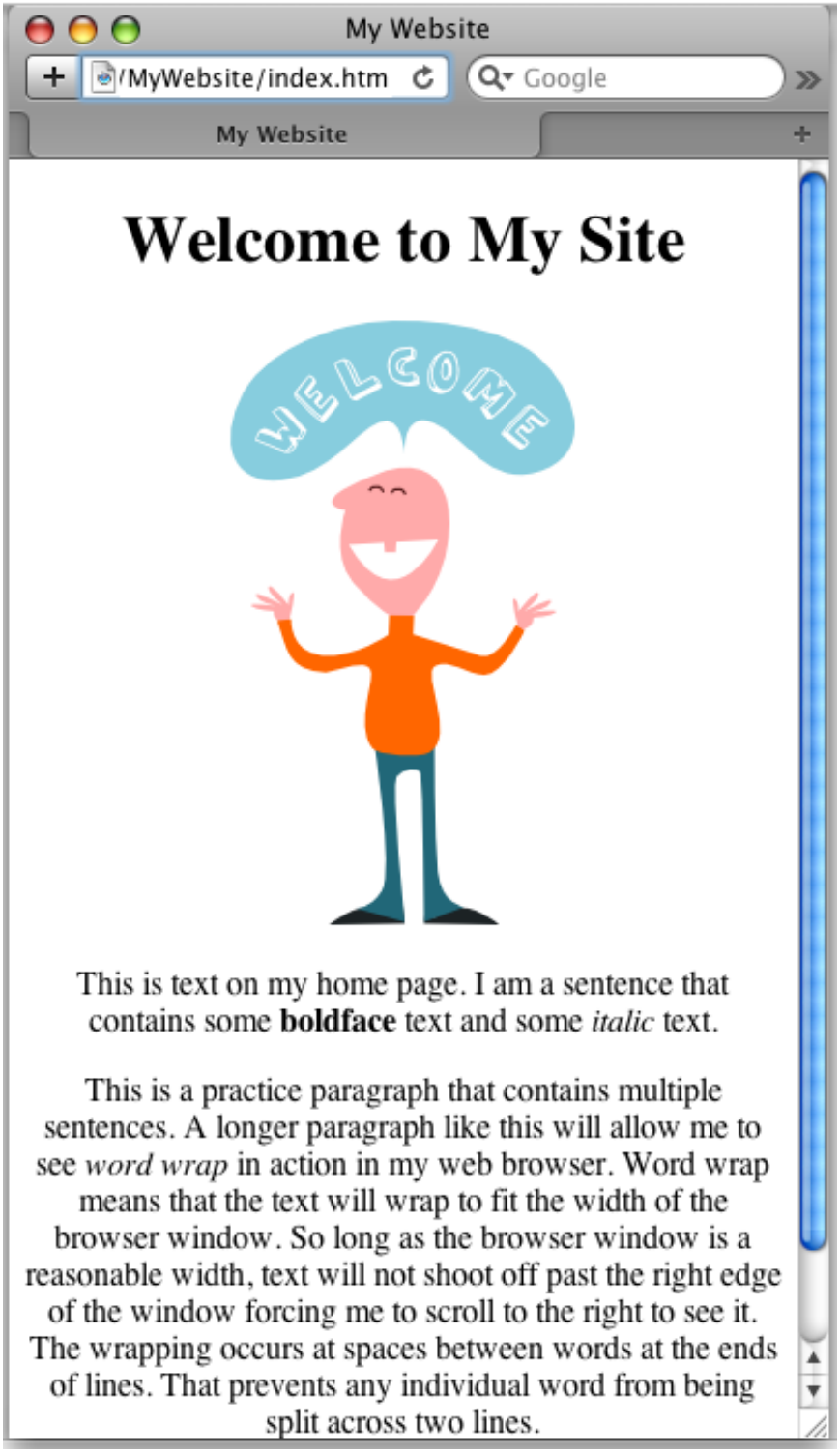
Conflicting Styles in CSS

In the Lesson 6 assignment, we added a CSS inline style to the <body> tag of our index.htm page. The result of that, in the Web browser, was to horizontally center everything on the page. The text is centered no matter how wide or narrow the browser window is. So if your browser window is very wide, the page might look something like this:



Content horizontally centered in wide browser window

If the browser window is narrow, then the page might look something like this:

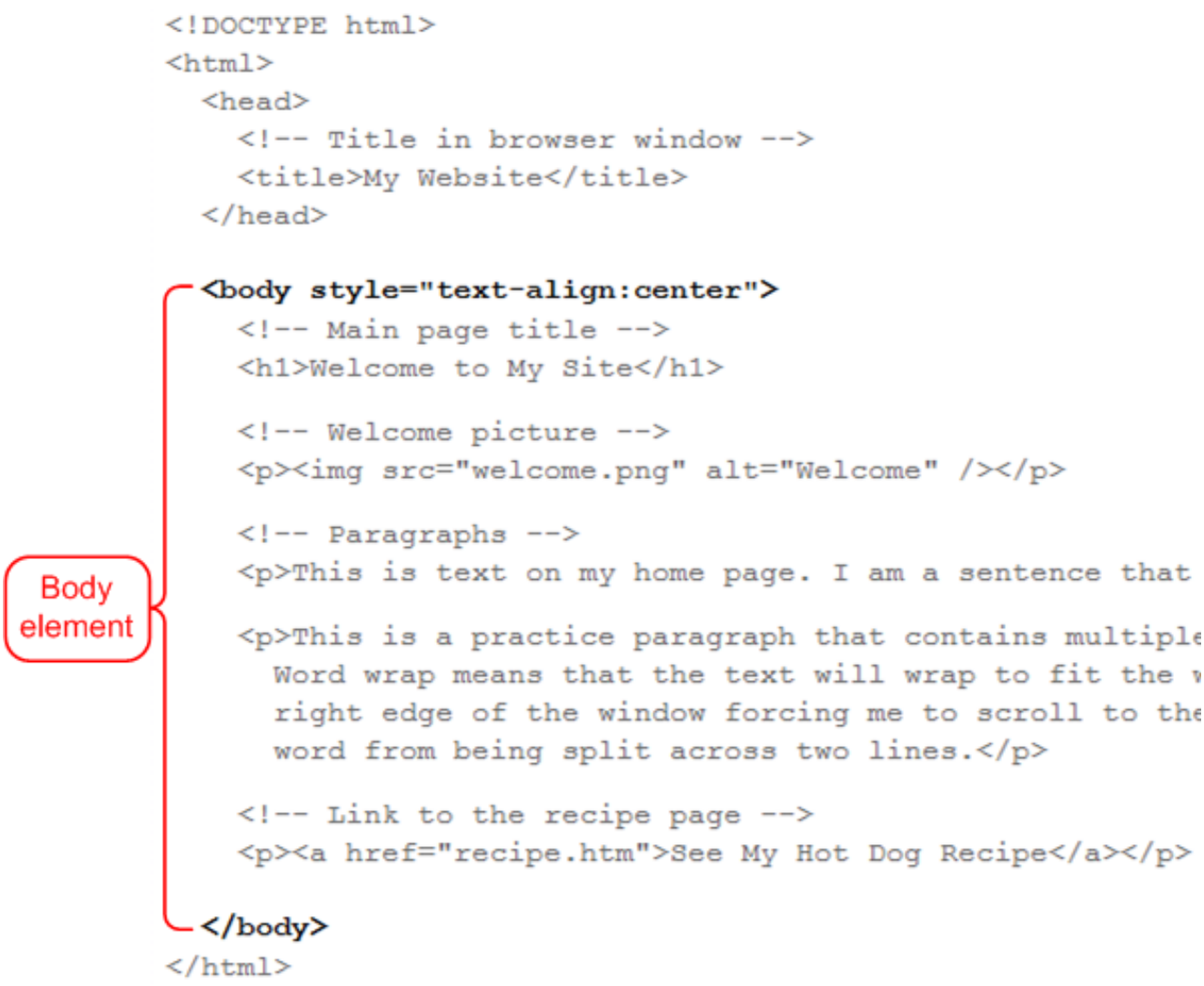


Content horizontally centered in narrow browser window

The width of the browser window doesn't really matter. As a Web developer, you don't get to dictate the size of the user's Web browser window. Everyone is free to size their program windows as they see fit. And that's just fine. All that matters here is that everything inside the browser window is centered horizontally because of the *style="text-align: center"* we put into the <body> tag of that page.

The centering applies to everything on the page, because the CSS text-align property is an *inherited property*—which means that when we apply a style to an element, every element inside that element also inherits the style. In this example, text-align:center is applied to the body element. We made that happen by putting *style="text-align: center"* in the <body> tag.

The body element is quite large. It extends all the way down to the closing </body> tag. In every Web page, the body element contains everything that's visible in the Web browser, as shown below. So all of the text that shows in the Web browser is horizontally centered—because in the code, it's all contained within the body element.



Body element defined by <body>...</body> tags

When Styles Conflict

Suppose that you decide your page might look better if the text in the longer paragraphs (only) were left aligned. Could you use inline styles to left-align the text in those two paragraphs only? Let's try it and see. Follow these steps:

1. Open index.htm for editing.

> **Note**
>
> 
>
> Hopefully, you know the drill by now. But just in case, to open index.htm for editing, first open your MyWebsite folder. Then, in Windows, right-click the **index.htm** icon, and choose **Open With > Notepad**. On a Mac, right-click the **index.htm** icon, and choose **Open With > TextEdit**.

2. Move the cursor between the *p* and > in the paragraph that starts "This is text on my home page."
3. Insert a space and *style="text-align: left"* so the start of that paragraph looks like this:

```
<p style="text-align: left;">This is text on my home page.
```

4. Now move the cursor down between the *p* and > that starts the paragraph that begins "This is a practice paragraph," and again insert a space and *style="text-align: left;"*

   Take a close look at your code, and verify that you did it correctly. The image below shows the new code, darkened against the code that was already in that section of the page. Don't worry about how the text in the paragraphs wordwrap in your editor. That's not important and has no bearing on how the page looks in a Web browser.

```
<p style="text-align:left;">This is text on my home page. I am a ser
and some <em>italic</em> text.</p>

<p style="text-align:left;">This is a practice paragraph that contai
allow me to see <em>word wrap</em> in action in my web browser. Word
the browser window. So long as the browser window is a reasonable wi
window forcing me to scroll to the right to see it. The wrapping occ
prevents any individual word from being split across two lines.</p>
```

Two CSS inline styles added to paragraphs

5. When you're confident that you typed the code correctly, save your changes (choose **File > Save** from the menu).

So now we have a bit of a conflict. Inside the body element, we have a couple of paragraphs that say "text in this paragraph is to be aligned left." But that conflicts with the more general style in the body element that says "everything on this page is centered," as illustrated here:

Left-aligned paragraphs inside body with center alignment

When conflicts like that arise, CSS resolves them with a clear, consistent, and common sense solution that we can state in a single sentence:

When styles conflict, the one that's closest to the element being styled takes precedence.

In other words, when styles conflict, the one that's closest to the tag being styled wins.

The *text-align: left;* in the <p> tag is much closer to the paragraph that's being styled than the *text-align: center* way up in the <body> tag. In fact, the *text-align: left;* is actually inside the <p> tag that's being styled, which is as close as you can possibly get. So the *text-align: left;* wins over the *text-align: center* in each paragraph that contains the *text-align: left;*. Everything else on the page, though, will continue to obey the more general style defined in the body tag, which says all text should be centered.

To verify that, open the page in any browser (double-click the **index.htm** icon). If you typed your code correctly, everything except the two larger left-aligned paragraphs will be horizontally centered within the browser window.

Larger paragraphs left-aligned

So there you have it, proof that when styles conflict, the one that's closest to the tag being styled wins. And you can count on it always working that way, because CSS was designed to work that way.

The Text-Align Property

So far, you've used the *left* and *center* values with the text-align property. There are actually four values you can use with text-align—*left*, *right*, *center*, or *justify*. The image below shows how the different values apply to multiline paragraphs. In real life, you'd probably only use *left* or *justify* for paragraphs. The *center* and *right* options are best used for shorter lines of text like titles, columns of dates, or numbers in a table.

property:value	Example
`text-align:left`	This is a paragraph with left-aligned text. It's smooth down the left side. The right side is ragged and uneven.
`text-align:center`	This is a paragraph of centered text. Each line is centered, so both the left and right edges are ragged and uneven. Centering works best for short one-line headings and such. Not so good for long paragraphs.
`text-align:right`	This is a paragraph of right-aligned text. The right edge is smooth, the left edge is ragged and uneven. Right-alignment is good for numbers or dates in columns, but not so good for multi-line paragraphs like this one.
`text-align:justify`	This is a paragraph of justified text. Both the left and right sides are smooth, no ragged edges. This works best in wide paragraphs. In narrow paragraphs you may notice the extra spaces inserted between some words to stretch short lines to the full paragraph width.

Examples of text alignment

The text-align property is one of many CSS properties that you can use to format text in your Web pages. You'll learn about others as we go.

But before we explore other CSS properties, I want to introduce you to another way of defining styles in your pages. Meet me in Chapter 3 to learn about *style rules*.

Chapter 3

CSS Style Rules

So far, you've seen an example of using a CSS inline style to center all of the text on a page. An inline style gets its name from the fact that it's in line with (actually, inside of) an HTML tag.

Style rules are another way to apply CSS styling to a page. Unlike inline styles, you don't put style rules inside Web tags. With style rules, there's no style= attribute and no quotation marks. Instead, there's a *selector* and some curly braces. The syntax for constructing a style rule looks like this:

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

As always, you never type the italicized text literally in your code. The italicized text is a placeholder for something else that you type in your code. The placeholders in a style rule are as follows:

- selector: The type of element that the style applies to (matches the letters inside the tag that define the element in the page).
- property: Any CSS property.
- value: Any acceptable value for the property.

The *property*: *value* pairs that you can use in a style rule are exactly the same as the *property*:*value* pairs that you can use in an inline style. So a style rule is just a different way to apply the same properties and values that you can apply using inline styles. You put inline styles in HTML tags. You put style rules in CSS *style sheets*.

Creating an Internal Style Sheet

An internal style sheet is one or more style rules inside a Web page. The style rules are *metadata* — stylistic information *about* the page. As such, an internal style sheet goes between the <head>...</head> tags in your page. But you can't just throw the style rules into the head section of the page willy-nilly. That won't work. You have to put them between the following HTML tags inside the head section:

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

Notice that there are no italic placeholders in those tags. That's because you type them exactly as shown. The opening tag marks the start of the internal style sheet and lets the user agent know that what comes next is text that's written in the CSS language. The closing </style> tag marks the end of the internal style sheet.

To try it all out, we'll add an internal style sheet to index.htm. We'll add some programmer comments too. As you may recall, comments are notes that you can write to yourself in the code, as reminders of the purpose of things. Comments are never required and have no effect on how the code operates. But they're useful as future reminders to yourself, especially when you're first learning. So here are the steps to follow:

1. Open index.htm for editing, using the same method you've been using throughout this course.
2. Put the cursor right after the </title> tag, and press ENTER to start a new line.
3. Type *<!-- Start of internal style sheet -->* and press ENTER to end that line.
4. Type *<style type="text/css">* and the press ENTER to end that line.
5. Press ENTER again to insert a blank line.
6. Type *</style>* and press ENTER.
7. Type *<!-- End of internal style sheet -->* and press ENTER.

The top of your page should look something like the example below. Double-check to make sure you that typed the tags correctly, and that they're inside the <head>...</head> tags. It doesn't matter if they're above or below the <title>...</title> tags.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Title in browser window -->
    <title>My Website</title>
    <!-- Start of internal style sheet -->
    <style type="text/css">

    </style>
   <!-- End of internal style sheet -->
  </head>
  <body style="text-align:center">
    <!-- Main page title -->
    <h1>Welcome to My Site</h1>
```

Comments and tags for internal style sheet added

Note

Indentations and line breaks in code are always optional. I didn't include steps for pressing the SPACEBAR to indent, because they're not required.

Now that we have the tags for defining an internal style sheet, we can add some style rules to that style sheet. We can start by defining a style rule for the page body. When we type a style rule for an element on a page, the selector for the style rule must match the word in the opening tag that defines the element on the page. We use a <body> tag to mark the beginning of the body element on the page. So we must use *body* as the selector in a style rule that's written to style the body element.

Also, when we write a style rule, we place an opening curly brace right after the element. And we use a closing curly brace to mark the end of the style rule. You can type the closing curly brace whenever you want. But it's a good idea to get in the habit of typing it right after the opening curly brace, so you don't have to remember to type it later. Here are the steps:

1. Put the cursor between the *<style type="text/css">* and *</style>* tags.

2. Type *body {*
3. Press ENTER twice to end that line and insert a blank line.
4. Type } to end the style rule.

So now you have an empty style rule in your style sheet. Any CSS *property*: *value*; pairs you put inside of that style rule's curly braces will apply to the body element. Why? Because you typed *body* as the selector.

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Title in browser window -->
    <title>My Website</title>
    <!-- Start of internal style sheet -->
    <style type="text/css">
    body{

    }
    </style>
   <!-- End of internal style sheet -->
  </head>
  <body style="text-align:center">
    <!-- Main page title -->
    <h1>Welcome to My Site</h1>
```

Empty style rule for the body element entered

Outside our new style rule, we currently have an inline style to center text in the <body> tag. We can define that in the style rule instead. There's no need to define it in both places, and there's no advantage to keeping the inline style in the body tag. So I suggest that you remove the blank space and style="*text-align:center;"* from the <body> tag first. And then we'll use a style rule to apply the centering. Here are the steps:

1. Delete the blank space and style="*text-align: center;"* from the <body> tag to remove that inline style.
2. Put the cursor between the curly braces of the new body {} style rule, and type *text-align:center;* there (see image below).

```
<!DOCTYPE html>
<html>
  <head>
    <!-- Title in browser window -->
    <title>My Website</title>
    <!-- Start of internal style sheet -->
    <style type="text/css">
    body{
      text-align:center;
    }
    </style>
   <!-- End of internal style sheet -->
  </head>
  <body>
    <!-- Main page title -->
    <h1>Welcome to My Site</h1>
```

text-align:center removed from body tag and placed in style rule

Stylistically, the page is still as it was before. The body element has *text-align: center* applied to it. So if you save the page and view it in a browser, everything will look exactly like it did before. There's no speed or other technical advantage to using the style rule or inline style, so you needn't concern yourself with that. We moved *text-align: center* to the style rule just to get some practice with other ways of doing things.

But we're not done yet! In the next chapter, you'll learn how to add some color to your pages. And we'll also use the new body {} style rule to apply those colors with CSS. Take a break if you'd like, and meet me in Chapter 4.

Chapter 4

Adding Color With CSS

Color is a great way to add some visual interest to your page. In this chapter, you're going to learn about adding color to your pages, using two CSS properties:

- **Background-color:** Allows you to define the background color of any element.
- **Color:** Allows you to define the foreground (text) color of any element.

Each property accepts one value. That value can be any one of the 16 color names listed below. Or it can be a color *hex code*—and there are a lot more hex codes than the 16 shown here:

Color	Name	Hex Code
	black	#000000
	gray	#808080
	silver	#c0c0c0
	white	#ffffff
	red	#ff0000
	maroon	#800000
	purple	#800080
	fuchsia	#ff00ff
	green	#008000
	lime	#00ff00
	olive	#808000
	yellow	#ffff00
	navy	#000080
	blue	#0000ff
	teal	#008080
	aqua	#00ffff

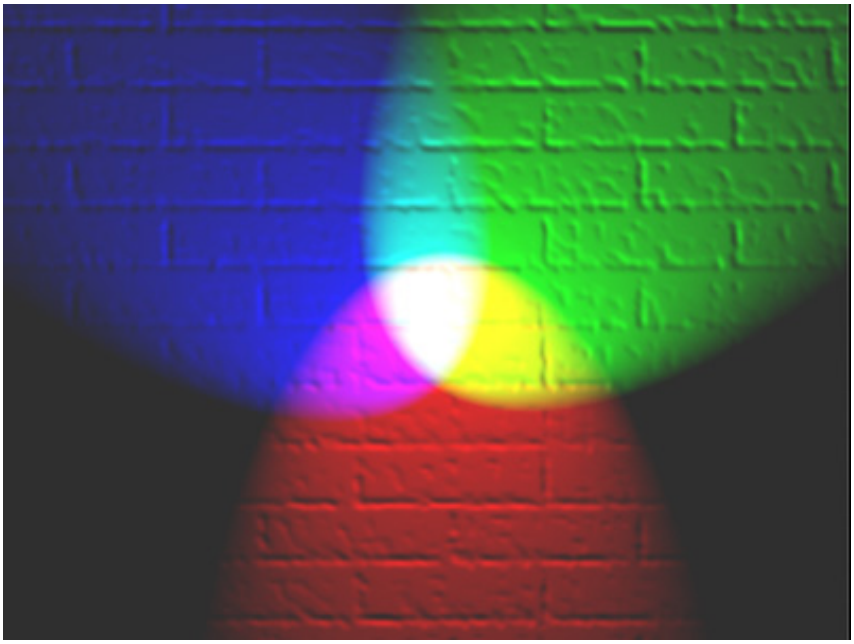
Sixteen named colors for CSS

The color names are nice, because they're familiar, easy to remember, and easy to type. The only disadvantage to the color names is that there are so few of them. If you stick to using only the color names, you're limited to using the 16 colors shown above.

But if you use the hex codes, you have nearly unlimited color options. In fact, using hex codes extends your choices to more than 16 million different colors. The hex codes have no meaning to any normal person, and they're probably a little intimidating at first. But don't be intimidated! We're going to spend some time getting to know them in this chapter.

Hex Codes

The hex codes aren't completely random. They're based on the fact that you can create just about any color (even white) by mixing the colors red, green, and blue.



Hex codes represent mixtures of red, green, and blue

The color hex code is six digits long. The first two digits (*rr* below) represent the amount of red. The second two digits (*gg* below) represent the amount of green. And the third two (*bb* below) represent the amount of blue.

```
rrggbb
```

A pair of zeroes in any position means that color is completely excluded. So if all six digits are zero, as in 000000, that means there's no color at all. And in fact, 000000 is the hex code for black, which is what you get when there are no colors at all.

Full strength of a color is represented by ff, which is the hexadecimal equivalent of the number 255. Hexadecimal is a numbering system that uses *0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, f* as characters, rather than just 0 through 9. It's fairly common in computers, because much of what happens in computers is done in bytes, which is 8 bits, or 2^8, which is 256. But that's not important to know or understand in terms of using colors.

The hex code ffffff means full-intensity red, plus full-intensity green, plus full-intensity blue. When you mix all three colors at full strength, you get white. What's the hex code for white? You guessed it—ffffff.

But you might have noticed in the list of colors and hex codes in the image above, each hex code begins with a # character. So in code, you would actually type *#ffffff* rather than *ffffff*.

Let's look at a few more examples. The hex code #ff0000 means full-intensity red and none of the other two colors. So #ff0000 is the hex code for red. The hex code #00ff00 means no red or blue, but full-intensity green. And #00ff00 is the hex code for green. And #0000ff is the hex code for blue, because 0000 means no red or green, and the *ff* at the end means full-intensity blue. Are you getting the picture?

Note

Color names and hex codes are not case-sensitive. So you don't have to worry about uppercase and lowercase letters. Some people use initial caps on color names, like *Black* rather than *black*. And some people use uppercase letters in hex codes—for example #FF00FF rather than #ff00ff. That's all okay.

Nobody knows the color hex codes by heart. There isn't even a place you can go to see them all, because there are more than 16 million of them, and that would require a huge Web page that would take a long time to download. Fortunately, there's no need to memorize them or see all 16 million at once. You can easily look them up on-the-fly right from your computer whenever you're online.

One easy way to find the hex code for a color is to use an online color wheel like the one at this website: http://www.allprofitallfree.com/color-wheel.html. Just go to that site, put the mouse pointer on the color you want, and the hex code for that color appears in a little box on the page.

There are also color scheme designers like the one here: http://colorschemedesigner.com/. Not only can they help you find hex codes for individual colors, they can help you find color combinations that are pleasing to the eye.

When you look around the Web at CSS and HTML color information, you may occasionally see reference to *Web safe colors* (256 acceptable colors). That's been outdated for years, and you can feel free to ignore it. All 16.7 million colors are perfectly safe nowadays, and they have been for many years.

You may also see names that go beyond the 16 I showed you earlier. Names like *AntiqueWhite* and *BlanchedAlmond*. Those names aren't officially sanctioned by the W3C anymore. While they'll probably work in most browsers, there's no guarantee that they'll work in all user agents. So it's usually safer to use the hex code rather than the fancy name.

CSS Properties to Apply Color

To use a color, you have to apply it to a CSS property. To define the background color, use the background-color property and the following syntax:

```
background-color: value
```

To define the text color, use the color property with this syntax:

```
color: value
```

In both cases, replace *value* with one of the 16 acceptable color names, or any one of the 16.7 million color hex codes. Don't forget that when you use multiple CSS *property: value* pairs, you have to separate them with semicolons. So if you want to set the background color of the body element to aqua, and the text color to navy blue, follow these steps:

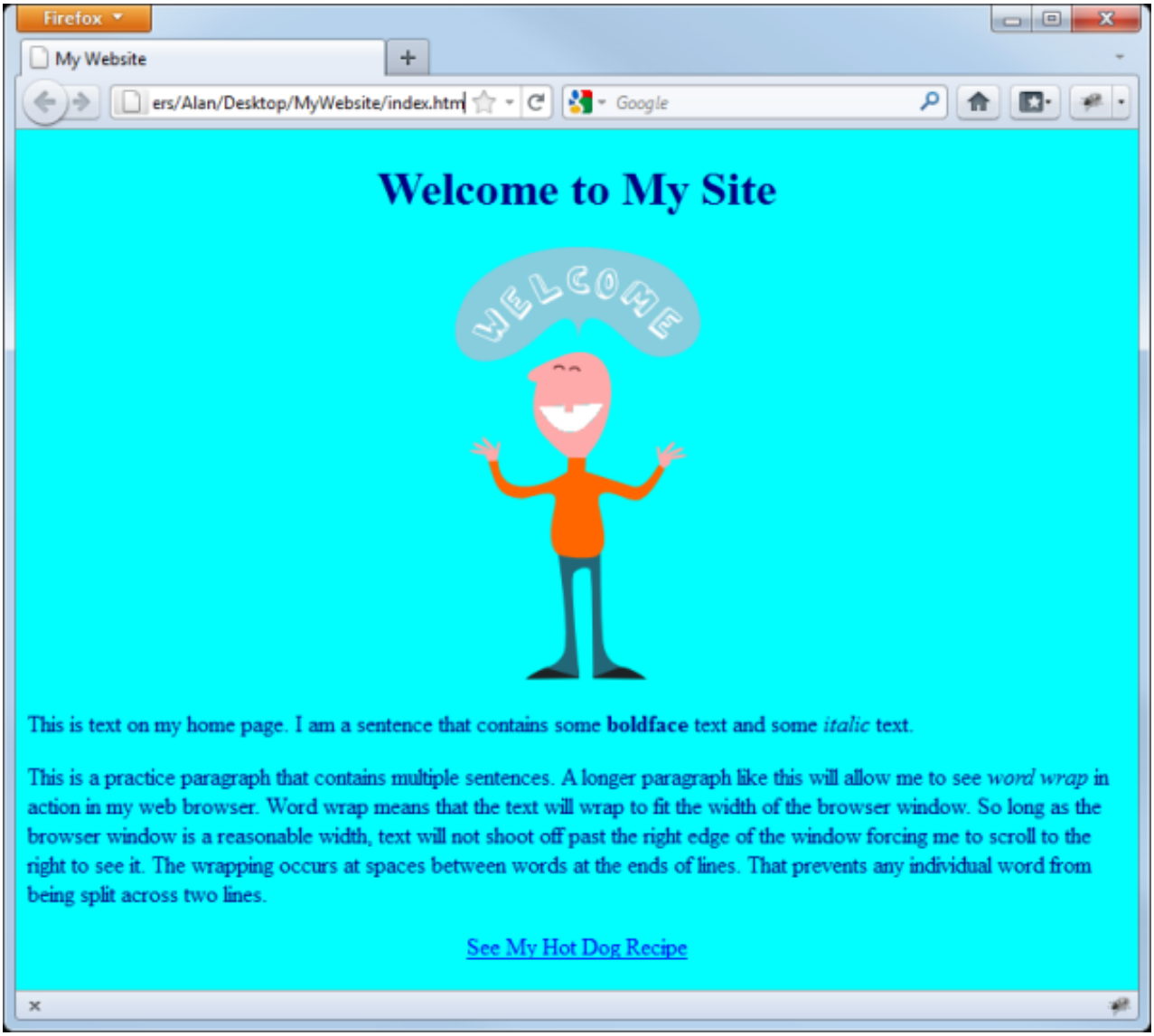
1. In index.htm, put the cursor just after *text-align: center;* and press ENTER to start a new line.
2. Type *background-color: aqua;* and press ENTER.
3. Type *color: navy;* (that last semicolon is optional, but it can't hurt to put it in, so it's already there should you decide to add more property: value; pairs later).
4. Make sure both new lines are inside the body {} style rule and spelled correctly, as below.

```
<!DOCTYPE html>
<html>
<head>
  <!-- Title in browser window -->
  <title>My Website</title>
  <!-- Start of internal style sheet -->
  <style type="text/css">
    body {
      text-align: center;
      background-color: aqua;
      color: navy;
    }
  </style>
  <!-- End of internal style sheet -->
</head>
<body>
  <!-- Main page title -->
  <h1>Welcome to My Site</h1>
```

Two new property: value pairs in the body style rule

5. When you're sure your code is right, save the page and open it in the browser (or refresh the page in the browser if it's already open there).

If you did everything correctly, your page should now have a light blue background and dark blue text. This is what it should look like:



Background and foreground colors

If yours doesn't look right, check your spelling and syntax. Here are a few more tips:

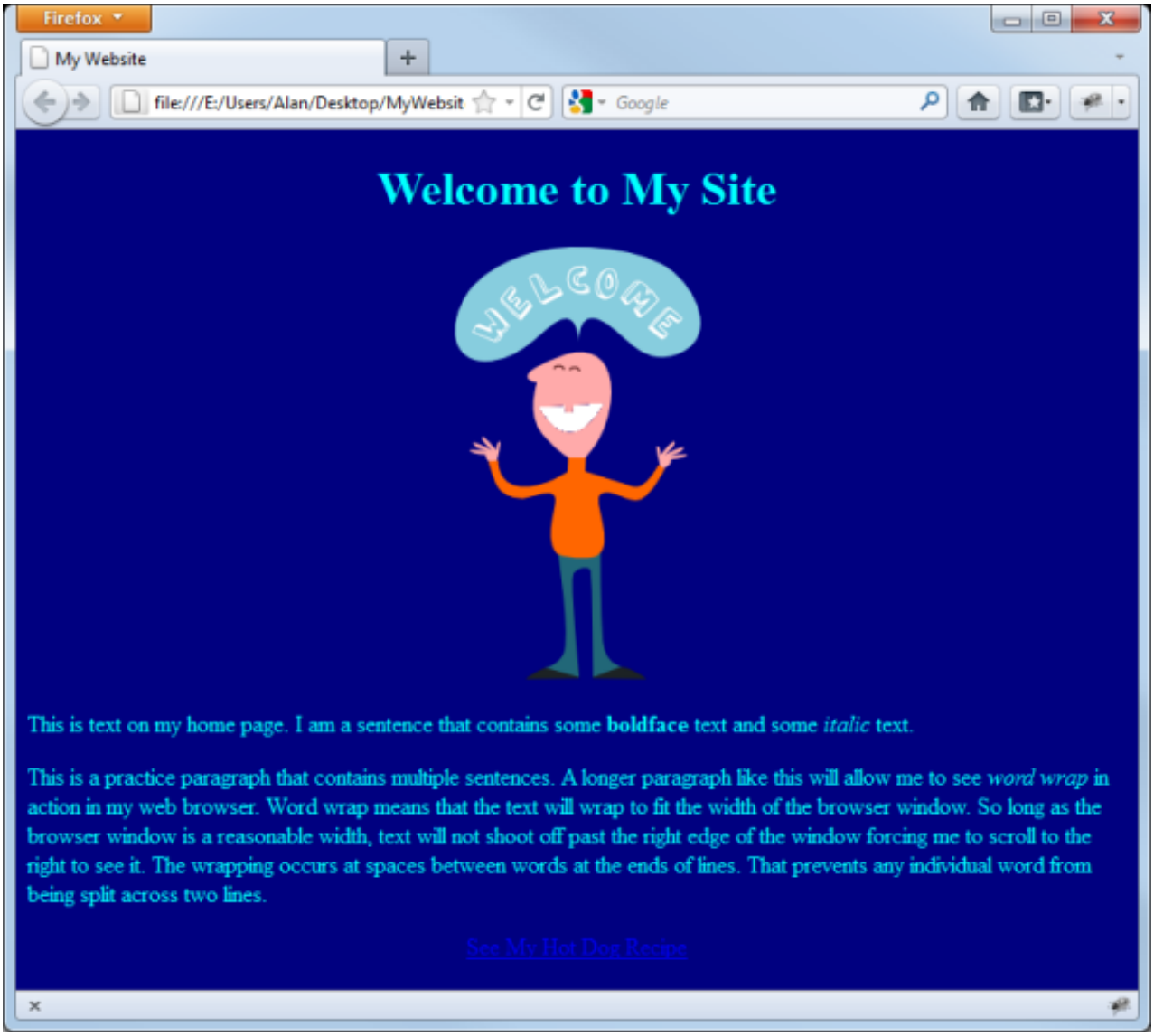
- Make sure you spelled the property names correctly. It's *background-color* (no blank spaces). And it's color for the text. You must use the U.S. spelling *color*, not *colour*).
- Make sure you put a colon (:) between the property and the value.
- Make sure you put a semicolon (;) between each *property: value* pair.
- Make sure all three *property: value* pairs are inside the curly braces of the body {} style rule.

Once your code is working, feel free to be creative. For example, you can try reversing the colors, making the background color navy, and the color aqua. This is what your code would look like:

```
body {

            text-align: center;

            background-color: navy;
```

```
            color: aqua;

            }
```

If you save the page after making that change, then view (or refresh) it in the browser, and the colors will be reversed there as well.



New colors for the index page

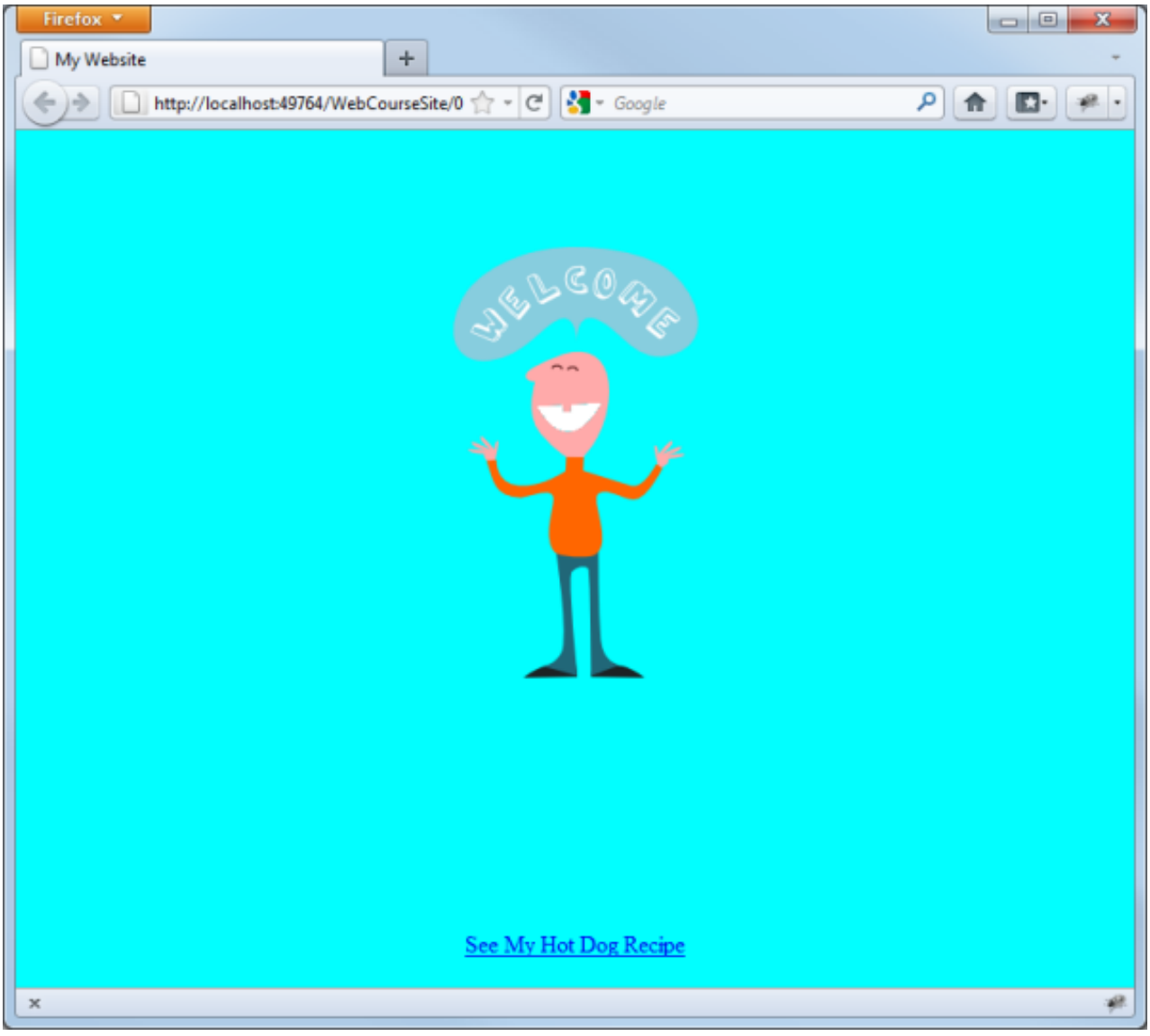
Troubleshooting Disappearing Text

When you choose colors, it's important that you provide good contrast between your background color and text color. Otherwise, the text will be hard to read or invisible. For example, here's a fairly common mistake. Suppose you change your mind about those colors, go back into your page, and change the background-color to aqua, but then you don't do anything to the color. Now, the background-color and color are the same.

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

            body {

            text-align: center;
```

```
background-color: aqua;

color: aqua;

}

</style>
```

You save your page, and then view in a browser. It looks like the page below, and in a panic, you think that most of the text has been deleted or somehow disappeared through some mysterious computer glitch.



Text is same color as background, making it invisible

The text is still there. You just can't see it, because it's the same color as the background.

To fix that problem, you must make sure you choose, and specify, colors that contrast well, like we did in our first two examples. In other words, write your style rule something like this, where the background-color and text color aren't the same.

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

```
body {
text-align: center;
background-color: aqua;
color: navy;
}
</style>
```

I suggest you do that right now. Then, when you save the page and view or refresh it in the browser, the text will all be dark blue and easy to see again.

Tip

Background colors usually don't show on printed pages, because it takes a lot of printer ink to fill the background—and ink isn't cheap! If you want to print background colors, see the FAQs for this lesson. (Click **Resources** near the top or bottom of this page, then choose **Frequently Asked Questions**, and then click **Lesson 7**.)

You probably noticed that the text at the bottom of the page never actually changed color. That's because that text represents a link. The color property doesn't apply to links; it only applies to regular text. To style links, you need to use different style rules, which you'll learn about in the next lesson.

Styling Headings

Okay, so you've styled the background color and the text color, but chances are, you'll also want to style the color of your headings. To style headings, you need to create a style rule with a selector that matches the type of heading you want to style. For example, the first style rule we created used *body* as the selector to style the body element. To create a style rule for level-1 headings (which are defined by <h1>…</h1> tags), the selector will be h1 (the characters inside the first tag).

Like the first style rule you created, this one needs to go in the internal style sheet. You can put as many style rules as you like between those tags—there's no maximum number. Also, like all style rules, this new one must have its own selector and curly braces. So to type it, follow these steps:

1. If you haven't already done so, open index.htm for editing.
2. Click just past the closing curly brace for the body style rule.
3. Press ENTER a couple of times to start a new line, and add a blank line.
4. Type *h1 {* to start the style rule, and then press ENTER.
5. Type *color: red;* which will be the property and value we want to apply to the h1 elements.
6. Press ENTER, and then type *}* to end the style rule.

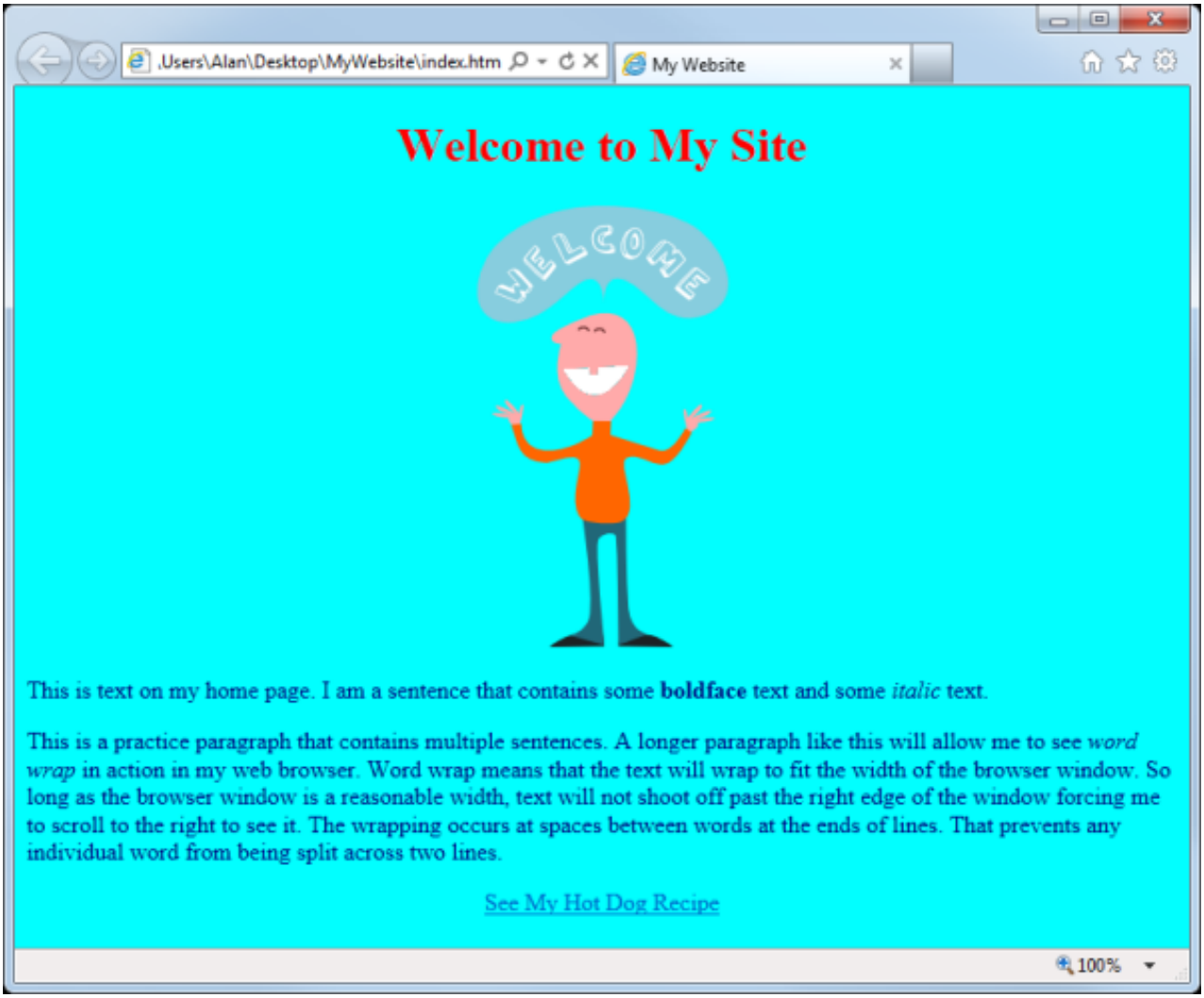
That new style rule should be under the first one, but still above the </style> tag that marks the end of the internal style sheet. The image below shows how it should look. Again, blank lines and indents don't matter, as long as you

don't break a line in the middle of a word.

```
<!DOCTYPE html>
<html>
<head>
  <!-- Title in browser window -->
  <title>My Website</title>

  <!-- Start of internal style sheet -->
  <style type="text/css">
    body {
      text-align: center;
      background-color: aqua;
      color: navy;
    }

    h1{
      color:red;
    }

  </style>
  <!-- End of internal style sheet -->
</head>
<body>
  <!-- Main page title -->
  <h1>Welcome to My Site</h1>
```

Second style rule in internal style sheet

What that new style rule says is "When you come across a level-1 heading (text in <h1>...</h1> tags), color that text red." If you save the page and view or refresh it in a browser, the main title at the top will be red, as in the image below.



Level-1 heading is red

Only the main title, Welcome to My Site, is red. That's because it's the only text on the page that's in <h1>...</h1> tags. And the h1 {} style rule we added to the page applies to h1 elements, like that title. (If it doesn't work for you, there's probably a typographical error in your code.)

You may be wondering if there's really any advantage to using style rules over inline styles in your Web pages. Though it's hard to tell when working with small examples like these, style rules do have a big advantage over inline styles. When you create a style rule for an element, it applies to all instances of that element on the page. For example, if you create a style rule like the one below, the style rules applies to all of the h2 elements on the page.

```
h2 {
            color: green;
            }
```

So if you had a large page with lots of h2 headings on it, that one style rule would color them all green. You wouldn't have to put *color: green;"* in every single h2 tag. And better still, if you had colored them green and then changed your mind and wanted to make them all blue, you'd only have to change that one style rule. You wouldn't have to do each tag individually.

That's a lot quicker and easier than finding and changing every h2 element on the page one at a time. And that's the

main reason why all large, professionally developed sites rely on CSS for virtually all of their styling. So if you ever decide to go into Web design professionally, you'll be expected to know CSS!

Let's head over the Chapter 5 now and summarize what you've learned today.

Chapter 5

Summary

We covered a lot of ground today! Beyond the specifics and details, it's important to understand that when it comes to styling websites and other electronic documents, CSS is king. It has been for several years, and will be for many years to come. So far, we've only scratched the proverbial surface of what you can do with CSS. So don't let the few things we've covered limit your view of what's possible.

In today's lesson, you learned that when style rules conflict, the one that's closest to the element being styled takes precedence. This means that you can set up some general styles that you apply to the body element. And then you can override them, if needed and as needed, for individual elements on the page.

Inline styles are CSS styles that actually go inside an HTML tag using a style= attribute.

A CSS style rule is a way of applying CSS outside the tags—and even outside the body of the page. A style rule consists of a *selector*, which describes what type of element the style rule styles. The selector is followed by a pair of curly braces that contain CSS *property: value* pairs that define how it's styled.

The *background-color* property in CSS lets you define the background color of the page (or any other element). The *color* property allows you to color the text. For either property, you define a color using one of the 16 color names or a color hex code that defines the color as a mixture of red, green, and blue. The Supplementary Material for this lesson provides links to websites that can help you find hex codes for expressing colors.

Don't forget to test your knowledge by taking the quiz for this lesson. And check out the assignment to get some hands-on practice typing color hex codes.

In the next lesson, you'll learn more about style rules, as well as some more *property:value* pairs for styling your Web pages. See you there!

Supplementary Material

Internal CSS

http://www.tizag.com/cssT/internal.php

Click this link for a quick tutorial on creating internal style sheets.

CSS Text

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

This page shows examples of aligning and coloring text with CSS.

Color Palette Generator

http://www.generateit.net/color-palette-generator/

If you're trying to match colors in a favorite digital photo, you can upload the photo here to see a palette of its main colors.

Color Combinations

http://www.webdevelopersnotes.com/design/color_combinations.php3

Click this link for a page that provides some quick-and-easy color combinations that you may want to try out in your own pages.

Color Names

http://www.colorsontheweb.com/colornames.asp

This page lists some common color names like *Brown* and *Silver*, as well as some not-so-common names. Then, it provides sample hex codes that represent that color. Very handy when you have a color name in mind and are looking for a hex code.

Spin the Color Wheel

http://www.colorsontheweb.com/colorwheel.asp

Here's a fun way to look at color combinations. Click the Spin button to spin the color wheel and see a random color scheme. You can just keep clicking Spin until you find a scheme you like.

The Color Wizard

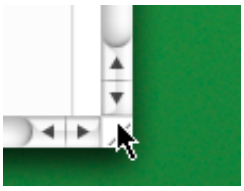
http://www.colorsontheweb.com/colorwizard.asp

On this site, the Color Wizard lets you submit a base color to start with (or you can click Randomize to have it pick one for you), and then it shows you colors that go well with it. You can just keep clicking Randomize until you get a base color that you like.

FAQs

Q: How do I size my browser window in Mac OS?

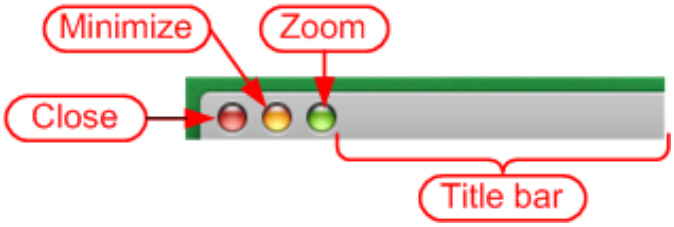
A: To size a window in Mac OS, get the tip of the mouse pointer down to the sizing handle in the lower right corner of the window you want to resize. It's a small area with three diagonal lines. The image below shows the mouse pointer on that handle.



Mouse pointer positioned to move a program window

Once the mouse pointer is in place, hold down the mouse button and move the mouse in the direction you want to size the window. Release the mouse button when the window is the size you want it to be.

To move the window to a different location on the screen, place the mouse pointer on an empty area in the title bar across the top of the window. Then, hold down the mouse button and move the mouse in the direction you want to move the window.



Mac window components

Use the Close button to close the window, or the Minimize button to shrink it to where only its icon is visible near the lower right corner of the screen. Click that icon in the lower right corner of the screen to restore it to its previous size and position.

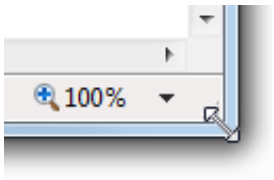
To size the window to its maximum height, click the green Zoom button.

Q: How do I size my browser window in Windows?

A: In Windows, first make sure the window isn't maximized (filling the entire screen). If it is maximized, you'll see a Restore Down button with two small squares in the upper right corner of the window (see image below). Click that Restore Down button before proceeding to the next step.

Windows window buttons

When the window is no longer maximized, put the tip of the mouse pointer anywhere along the border of the window frame, or in a corner of the frame. You'll know the mouse pointer is in a position to size the window when it turns into a two-headed arrow (see example below).



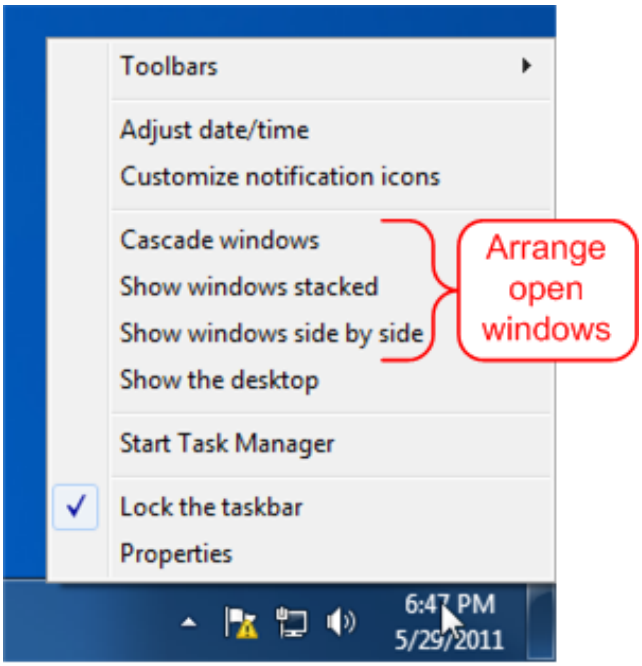
Mouse pointer positioned to size a window

Once you see the two-headed mouse pointer, gently depress and hold down the left mouse button. Keep the left mouse button held down while moving the mouse in the direction you want to move. Release the mouse button when the window is the size you want it to be.

When the window is smaller than the full size of the screen, you can also move it to a different location by dragging its title bar. Put the mouse pointer on the title bar at the top of the window you want to move (see example below). Hold down the left mouse button, and move the mouse in the direction you want to move the window. Then, release the mouse button. When the window isn't maximized to the full screen size, you'll also see the Maximize button (a single large square) where the Restore Down button used to be. You can click that Maximize button to size the window to full screen.

Mouse pointer positioned to move a program window

Most versions of windows also have options for automatically stacking windows or sizing them to fit side by side. Right-click an empty area of the taskbar along the bottom of the screen, or right-click the current date and time in the lower right corner of the screen. Then, choose one of the available options if you'd like to try it out. Note, however, that those options work best when you have at least two open windows on the desktop.



Right-click the date

Q: Why don't background colors print, and how can I make them print?

A: Printer ink isn't cheap, and filling in entire page backgrounds with a color takes quite a bit of ink. So most Web browsers won't print background colors by default. But you can override that setting if you'd like. The exact steps for changing the setting depend on the browser brand and version you're using. Here I'll cover the most commonly used browsers—Safari for the Mac, and Internet Explorer for Windows.

To print background colors on a Mac in Safari, click **File** on the menu, and choose **Print** to start the print process. Click **Copies & Pages,** and choose **Safari**. Then, select (check) **Print Backgrounds**, and print your page.

In Internet Explorer, how you do it may vary depending on which version you're using. In Internet Explorer 9, tap the ALT key, and choose **File** > **Page Setup** from the menu bar. Select (check) the **Print Background Colors and Images** checkbox, and click **OK**. Then, choose **File > Print** to print the page.

In earlier versions of Internet Explorer, tap the ALT key (if the menu bar isn't showing), and choose **Tools** > **Internet Options** from the menu bar. In the Internet Options dialog box that opens, click the **Advanced** tab, scroll down to the Printing section, and select (check) **Print background colors and images**. Click **OK**. Any pages you print from that point forward will print with background colors. To go back to the default, repeat the process, and clear that checkbox.

If you use other browsers and would like to do the same in those, consider searching Google for help. For example, searching Google for something like *firefox print background colors* should help you discover how to print background colors in the Firefox Web browser.

Assignment

Part 1

I used simple color names in the lesson, because they're a lot easier to understand and they're a lot less intimidating than the color hex codes. But you don't need to limit your Web creations to those 16 colors. Any place where you can use one of those names, you can use the hex code instead. For starters, your hands-on challenge will be to use #ffeec6 as the body background color, and #2d3b71 as the body text color and the h1 text color. When you view index.htm in a browser, the colors will look like this:

New color scheme for index.htm

Try to do it on your own without peeking. If need help, click this link for step-by-step instructions.

Part 2

The second part of this assignment is to understand that nobody was ever born knowing all 16.7 million color hex codes. Nor has anybody ever memorized them all. If you want to be good with colors and hex codes, you're going to have to be resourceful about finding them when you need them. To that end, I ask that you spend a little time looking at some of the websites listed in the Supplementary Material for this site. When you find a site you like, add it to your Web browser's Favorites or Bookmarks so that you can find it easily in the future. And after you finish that, if you'd like to extend your horizons even further, consider using a search engine like www.google.com to search for phrases like *html color codes* or *css color schemes*. And if you want to be a little more creative, try searching for things like *nature color scheme* or *zen color scheme*—or whatever suits your fancy.

If you have any trouble, keep in mind that there are three very common mistakes when using color hex codes:

- Forgetting the # character in front of the six-digit code.
- Typing only five of the required six digits (or typing an extra character).
- Typing the letter *l* for the number *1*, or the letter *o* for the number *0* (zero). Remember, after the #, the only acceptable characters are *0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e,* and *f.*

Of course, you must type all of your code correctly. So if your hex code is right but your color isn't, make sure you spelled *background-color:* or *color:* correctly (and didn't spell it *colour* in either case), and that you have a colon between the property and value!

Back to top

CLOSE