**Using Style IDs**

When you create custom style classes, you can use those classes as many times as you would like—

they are not unique. However, there will be some instances when you want to have precise control

over unique elements for layout or formatting purposes (or both). In such instances, look to ID*s*

instead of classes.

A *style ID* is a custom set of formatting specifications that can be applied only to one element in a

web page. You can use IDs across a set of pages but only once per time within each page.

For example, suppose you have a title within the body of all your pages. Each page has only one title,

but all the pages themselves include one instance of that title. Following is an example of a selector

with an ID indicated, plus a property and a value:

p#title {font: 24pt Verdana, Geneva, Arial, sans-serif}

Notice that this selector includes a hash mark, or pound sign (#), after p, followed by a descriptive ID

name. When referencing a style ID in HTML code, simply specify the ID name in the id attribute of an

element, like so:

<p id="title">Some Title Goes Here</p>

Everything between the opening and closing <p> tags will appear in 24-point Verdana text—but only

once on any given page. You will often see style IDs used to define specific parts of a page for layout

purposes, such as a header area, footer area, main body area, and so on. These types of areas in a

page will appear only once per page, so using an ID rather than a class is the appropriate choice.

**Internal Style Sheets and Inline Styles**

In some situations, you might want to specify styles that will be used in only one web page, in which

case you can enclose a style sheet between <style> and </style> tags and include it directly in an HTML

document. Style sheets used in this manner must appear in the <head> of an HTML document. No <link />

tag is needed, and you cannot refer to that style sheet from any other page (unless you copy it into the

beginning of that document, too). This kind of style sheet is known as an internal style sheet, as you

learned earlier in the chapter.

Example 3.3 shows an example of how you might specify an internal style sheet.

Example 3.3 A Web Page with an Internal Style Sheet

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"

"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en">

<head>

<title>Some Page</title>

<style type="text/css">

div.footer {

font-size: 9pt;

line-height: 12pt;

text-align: center;

}

</style>

</head>

<body>

...

<div class="footer">

Copyright 2009 Acme Products, Inc.

</div>

</body>

</html>

**Note**

<span> and </span> are *dummy* tags that do nothing in and of themselves except specify a range of content

to apply any style attributes that you add. The only difference between <div> and <span> is that <div> is a

block element and therefore forces a line break, whereas <span> doesn’t. Therefore, you should use <span>

to modify the style of any portion of text that is to appear in the middle of a sentence or paragraph

without any line break.

In the example code, the div.footer style class is specified in an internal style sheet that appears in the head

of the page. The style class is now available for use within the body of this page. And, in fact, it is

used in the body of the page to style the copyright notice.

Internal style sheets are handy if you want to create a style rule that is used multiple times within a

single page. However, in some instances you might need to apply a unique style to one particular

element. This calls for an inline style rule, which allows you to specify a style for only a small part of

a page, such as an individual element. For example, you can create and apply a style rule within a <p>,

<div>, or <span> tag via the style attribute. This type of style is known as an *inline style* because it is

specified right there in the middle of the HTML code.

Here’s how a sample style attribute might look:

<p style="color:green">

This text is green, but <span style="color:red">this text is

red.</span>

Back to green again, but...

</p>

<p>

...now the green is over, and we're back to the default color

for this page.

</p>

**Validate Your Style Sheets**

Just as it is important to validate your HTML or XHTML markup, it is important to validate your style sheet. A specific

validation tool for CSS can be found at http://jigsaw.w3.org/css-validator/. Just like the validation tool discussed in Chapter 2,

“Understanding HTML and XHTML Connections,” you can point the tool to a web address, upload a file, or paste content into

the form field provided. The ultimate goal is a result such as that shown in Figure 3.3: valid!

**Figure 3.3** The W3C CSS Validator shows there are no errors in the style sheet contents of Example 3.1.

This code makes use of the <span> tag to show how to apply the color style property in an inline style rule.

In fact, both the <p> tag and the <span> tag in this example use the color property as an inline style. What’s

important to understand is that the color:red style property overrides the color:green style property for the text

appearing between the <span> and </span> tags. Then in the second paragraph, neither of the color styles

applies because it is a completely new paragraph that adheres to the default color of the entire page.

**Summary**

In this chapter, you learned that a style sheet can control the appearance of many HTML pages at

once. It can also give you extremely precise control over the typography, spacing, and positioning of

HTML elements. You also discovered that by adding a style attribute to almost any HTML tag, you can

control the style of any part of an HTML page without referring to a separate style sheet document.

You learned about three main approaches to including style sheets in your website: a separate style

sheet file with the extension .css that is linked to in the <head> of your documents, a collection of style

rules placed in the head of the document within the <style> tag, and as rules placed directly in an HTML

tag via the style attribute.

Table 3.1 summarizes the tags discussed in this chapter. Refer to the CSS 2 style sheet standards at

http://www.w3c.org for details on what options can be included after the <style> tag or the style attribute.

Table 3.1 HTML Tags and Attributes Covered in Chapter 3

**Q&A**

**Q. Say I link a style sheet to my page that says all text should be blue, but there’s a <span**

**style="font-color:red"> tag in the page somewhere. Will that text display as blue or will it display as**

**red?**

**A.** Red. Local inline styles always take precedence over external style sheets. Any style

specifications you put between <style> and </style> tags at the top of a page will also take precedence

over external style sheets (but not over inline styles later in the same page). This is the cascading

effect of style sheets that I mentioned earlier in the chapter. You can think of cascading style

effects as starting with an external style sheet, which is overridden by an internal style sheet,

which is overridden by inline styles.

**Q. Can I link more than one style sheet to a single page?**

**A.** Sure. For example, you might have a sheet for formatting (text, fonts, colors, and so on) and

another one for layout (margins, padding, alignment, and so on)—just include a <link /> for both.

Technically speaking, the CSS standard requires web browsers to give the user the option to

choose between style sheets when multiple sheets are presented via multiple <link /> tags. However,

in practice, all major web browsers simply include every style sheet. The preferred technique for

linking in multiple style sheets involves using the special @import command. Following is an example

of importing multiple style sheets with @import:

@import url(styles1.css);

@import url(styles2.css);

Similar to the <link /> tag, the @import command must be placed in the head of a web page. You learn

more about this handy little command in Chapter 25, “Creating Print-Friendly Web Pages,” when

you learn how to create a style sheet specifically for printing web pages.

**Workshop**

The workshop contains quiz questions and exercises to help you solidify your understanding of the

material covered. Try to answer all questions before looking at the “Answers” section that follows.

**Quiz**

**1.** What code would you use to create a style sheet to specify 30-point blue Arial headings and all

other text in double-spaced, 10-point blue Times Roman (or the default browser font)?

**2.** If you saved the style sheet you made for Question 1 as corporate.css, how would you apply it to

a web page named intro.html?

**3.** How many different ways are there to ensure style rules can be applied to your content?

**Answers**

**1.** Your style sheet would include the following:

h1 { font: 30pt blue Arial; }

body { font: 10pt blue; }

**2.** Put the following tag between the <head> and </head> tags of the intro.html document:

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

**3.** Three: externally, internally, and inline.

**Exercises**

• Using the style sheet you created earlier in this chapter, add some style classes to your style sheet.

To see the fruits of your labor, apply those classes to the HTML page you created as well. Use

classes with your <h1> and <p> tags to get the feel for things.

• Develop a standard style sheet for your website and link it into all your pages. (Use internal style

sheets and/or inline styles for pages that need to deviate from it.) If you work for a corporation,

chances are it has already developed font and style specifications for printed materials. Get a copy

of those specifications and follow them for company web pages, too.

• Be sure to explore the official style sheet specs at http://wwww3.org/Style/CSS/ and try some of

the more esoteric style properties not covered in this chapter.