

AN INTRODUCTION TO CSS

In terms of syntax, CSS is an easy language to learn. Its “grammar” consists of only a few concepts. That makes it relatively easy to get started with. What makes CSS a bit more difficult is the way all major browsers render a page. Although virtually every modern desktop browser understands CSS, they all have their quirks when it comes to displaying a page according to the CSS standard. This standard, maintained by the same organization that maintains the HTML standard, the World Wide Web Consortium, or W3C for short, comes in three different versions: 1.0, 2.1, and 3.0. From these three versions, 2.1 is the most applicable today. It contains everything that version 1.0 contains but also adds a lot of possibilities on top of that. It’s also the version that VWD uses and generates by default. Version 3.0 is currently under development and it’s expected to take some time before the major browsers have solid support for it.

Before you look at the actual syntax of CSS, it’s a good idea to see an example first. In the next exercise, you write a simple ASPX page that contains some CSS to format the contents of the page. This helps in understanding the CSS language, which is discussed in full detail in the section that follows.

TRY IT OUT Writing Your First CSS

In this Try It Out you write some CSS that changes the appearance of a header and two paragraphs. You’ll hand code the page for now; the second half of this chapter shows you how to use the CSS tools available in VWD.

1. In the **Demos** folder of the Planet Wrox project, create a new Web Form called `CssDemo.aspx`. For this exercise, it doesn’t matter if you choose inline code or Code Behind.
2. Make sure the page is in Markup View and then locate the closing `</title>` tag in the source. Position your cursor at the end of the line and press Enter to create an empty line between the `title` and `head` tags. On this new line type the word `style` and then press Tab. Visual Web Developer completes the `<style>` element for you. Press Enter twice to create some room between the tags. You end up with the following bolded code:

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

</style>
</head>
```



NOTE This code completion feature uses code snippets that enable you to associate a piece of code (like the `<style>` element) with an identifier (like `style` in this example). Code snippets are very useful to quickly insert pieces of code by typing only the short identifier. Many more code snippets are available, and where appropriate I’ll point them out throughout this book.

Instead of using the style code snippet, you can also type the full code yourself. Note that as soon as you type the opening angle bracket (<), a list pops up that enables you to select the <style> tag. The same applies to the `type` attribute; simply type the letters `ty` and the `type` attribute is preselected in the list. All you need to do to complete the word is press the Tab or Enter key. And, once more, the same help is available for the attribute value `text/css`. Simply select it in the list and press Tab or Enter, and the value is inserted for you automatically, nicely surrounded by the double quotes.

3. Next, between the opening and closing <style> tags, type the following bolded CSS code:

```
<style type="text/css">
  h1
  {
    font-size: 20px;
    color: Green;
  }

  p
  {
    color: Blue;
    font-style: italic;
  }

  .RightAligned
  {
    text-align: right;
  }
</style>
```

Take great care when typing this code, because CSS is rather picky about syntax. The first item in the list is an `h1` tag to style a heading at the first level so it gets a size of 20 pixels and is displayed in a green font. Notice the colon between `font-size` and `20px` and that the line is closed with a semicolon.

The second item in the list simply contains the letter `p` and defines the look and feel for all <p> elements in the page.

The last item is prefixed with a period (.) followed by the text `RightAligned`. This item is used to right-align some text in the page.

4. Scroll down in the page a bit until you see the opening <div> tag. Right after this tag, type the following bolded code:

```
<div>
  <h1>Welcome to this CSS Demo page</h1>
  <p>CSS makes it super easy to style your pages.</p>
  <p class="RightAligned">
    With very little code, you can quickly change the looks of a page.
  </p>
</div>
```

Instead of typing in this code directly, you can also use the Formatting toolbar while in Design View to create elements like `<h1>` and `<p>`. For now, you'll need to switch to Markup View to add `class="RightAligned"`, but in later exercises in this chapter you see how you can have the IDE write this code for you.

5. If you switch to Design View (or Split View), you'll see that the designer shows your text with the formatting defined in the `<style>` element of the page. Figure 3-1 shows the page in Split View so you can see the code and the design at the same time.

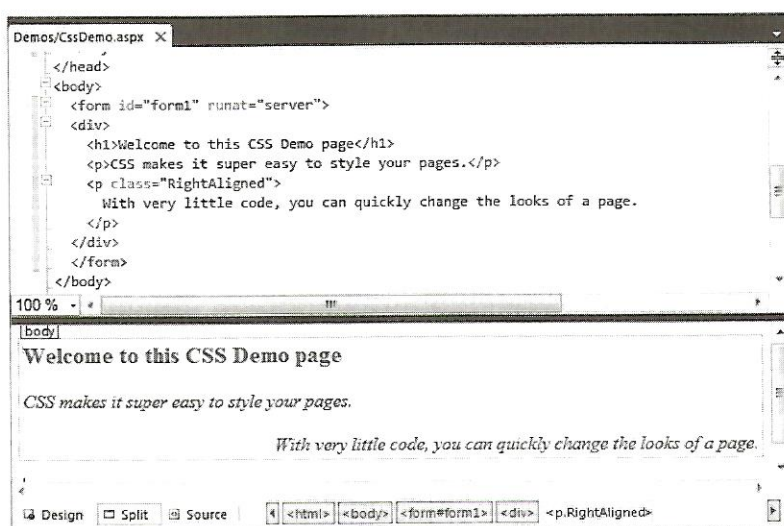


FIGURE 3-1

Although this black and white book makes it difficult to see different font colors, in Figure 3-1 you can clearly see that the `<h1>` has a larger font size. The figure also shows that all paragraphs (both the plain paragraph and the one with `class="RightAligned"`) are now displayed with an italic font. Finally, you can see that the final paragraph is aligned to the right of the window, because the `class` attribute on the tag is set to `RightAligned`.

If you don't see the last paragraph glued to the right border of the Document Window, make sure you typed `RightAligned` exactly the same in the `<style>` tag and in the `class` attribute. Because CSS is case sensitive, there's a big difference between `RightAligned` and `rightaligned`.

6. Press `Ctrl+F5` to view `CssDemo.aspx` in your browser. The page you see in the browser is identical to the preview you got in the Design View of Visual Web Developer.

How It Works

Although the code you typed in this exercise is relatively simple, there's a lot going on under the hood of the browser (and the Design View of VWD 2010) to make this possible. You started by adding some styles to the `<head>` section of the page:

```
<style type="text/css">
  h1
  {
```