

THE EXPERT'S VOICE® IN WEB DEVELOPMENT

# CSS

## Quick Syntax Reference

Mikael Olsson

Apress®

*For your convenience Apress has placed some of the front matter material after the index. Please use the Bookmarks and Contents at a Glance links to access them.*



**Apress®**

# Contents at a Glance

<b>About the Author .....</b>	<b>xv</b>
<b>About the Technical Reviewer .....</b>	<b>xvii</b>
<b>Introduction .....</b>	<b>xix</b>
■ <b>Chapter 1: Using CSS.....</b>	<b>1</b>
■ <b>Chapter 2: Grouping.....</b>	<b>5</b>
■ <b>Chapter 3: Class and id selectors.....</b>	<b>7</b>
■ <b>Chapter 4: Attribute selectors .....</b>	<b>11</b>
■ <b>Chapter 5: Pseudo selectors.....</b>	<b>15</b>
■ <b>Chapter 6: Relationship selectors .....</b>	<b>27</b>
■ <b>Chapter 7: Specificity .....</b>	<b>31</b>
■ <b>Chapter 8: Colors.....</b>	<b>35</b>
■ <b>Chapter 9: Units.....</b>	<b>39</b>
■ <b>Chapter 10: CSS Properties .....</b>	<b>43</b>
■ <b>Chapter 11: Text.....</b>	<b>47</b>
■ <b>Chapter 12: Spacing .....</b>	<b>51</b>
■ <b>Chapter 13: Font.....</b>	<b>53</b>
■ <b>Chapter 14: Background.....</b>	<b>57</b>
■ <b>Chapter 15: Gradients.....</b>	<b>63</b>
■ <b>Chapter 16: Box model .....</b>	<b>71</b>

■ <b>Chapter 17: Border</b> .....	<b>73</b>
■ <b>Chapter 18: Outline</b> .....	<b>77</b>
■ <b>Chapter 19: Margin and padding</b> .....	<b>79</b>
■ <b>Chapter 20: Dimension</b> .....	<b>83</b>
■ <b>Chapter 21: Positioning</b> .....	<b>87</b>
■ <b>Chapter 22: Classification</b> .....	<b>95</b>
■ <b>Chapter 23: List</b> .....	<b>101</b>
■ <b>Chapter 24: Table</b> .....	<b>105</b>
■ <b>Chapter 25: Media</b> .....	<b>109</b>
■ <b>Chapter 26: Best practices</b> .....	<b>113</b>
<b>Index</b> .....	<b>121</b>

# Introduction

CSS, which stands for Cascading Style Sheets, is a stylistic language that defines how web pages are presented. It complements HTML, which is the language used for describing the structure of web site content. Because HTML predates CSS, it includes a number of limited stylistic elements and attributes, but they have largely been deprecated in favor of the much greater design control offered by CSS.

One of the main features of CSS is that it enables the complete separation of a web site's presentation from its content. This separation reduces the complexity and repetition associated with including stylistic information in the structural content. Furthermore, this separation makes it easy to enforce site-wide consistency in the presentation of a web site because the entire look and feel of a site can be controlled from a single style sheet document.

As one of the core languages of the Web, CSS is used today by almost all web sites to enhance the web experience. It has been a revolution in the World Wide Web and is a must-learn language for anyone working with web design. Like HTML, the CSS language is easy to learn and use, as is shown in this book.

## CSS versions

The CSS language is under ongoing development by the World Wide Web Consortium (W3C), which is the international standards organization for the Internet. The W3C writes the specifications that web browsers implement to display web pages consistently in compliance with those specifications. Each new specification extends the language with new features while maintaining backward compatibility.

The first specification, CSS level 1 (or CSS 1), became a W3C recommendation in 1996. In 1998, CSS 2 was finalized, extending CSS 1 with additional features. Because all widely used web browsers currently implement the features of both these specifications, it is seldom necessary to make a distinction between them, and this book does so only when relevant.

Since 1998, the W3C has been working on CSS 3. Unlike the two earlier levels of CSS, this level became considerably larger and was therefore split into several separate specifications called modules. This split allowed the modules to mature independently at their own pace. As a result of the ongoing development, support for CSS 3 varies. Some features are stable and have widespread browser support; other features are supported only by the latest browser versions or are not supported at all. This book focuses mainly on the CSS 3 features that are supported in the major browsers at the time of writing.

## Rule structure

CSS is commonly used within a style sheet document, which consists of a list of rules. For example, a rule to color all paragraph elements red is shown here:

```
p { color: red; }
```

This rule has two parts: a selector and a declaration block. The selector is the link between the HTML document and the style sheet that specifies the element to which the rule is applied. In this case, it is the type selector `p` that applies to all paragraph elements (`<p>`). Any HTML element can be used as a type selector in this way.

The declaration block, which is enclosed within curly braces, defines the styling applied to the selected elements. The block can contain one or more declarations, each of which is made up of a style property followed by a colon and a valid value for that property. Each declaration is terminated with a semicolon, although this is optional for the last one.

```
p { color: red; background: black }
```

Although the last semicolon is not necessary, it is a good practice to include it because it is easy to forget the missing semicolon when you add more styles to the rule. Another general practice is to use lowercase letters when writing CSS, even though selectors and properties are case-insensitive.

To summarize, a style rule consists of a selector and one or more declarations, each comprising one or more property-value pairs. This structure is illustrated here:

	<b>Property</b>	<b>Value</b>	
<b>p</b>	{	<b>color : red;</b>	}
<b>Selector</b>		<b>Declaration</b>	

# CHAPTER 1



# Using CSS

There are three ways to insert CSS into an HTML document: by using an internal style sheet, inline styles, or an external style sheet. An internal style sheet applies to a single page, an inline style to a single element, and an external style sheet to potentially an entire web site.

## Internal style sheet

An internal style sheet is useful when a single document needs to have its own unique styling. The style sheet is then embedded within the `<head>` section of the web document using the `<style>` element. This element is a container for style sheet rules and should have its `type` attribute set to `"text/css"`.

```
<style type="text/css">
  p { color: red; }
</style>
```

## Inline style

Styling can be assigned to an individual element by using the `style` attribute to set an inline style. It is a generic attribute that can be included in any HTML start tag, and its value is the CSS declarations that will be applied to the element, separated by semicolons. There is no need to specify a selector because the declarations implicitly belong to the current element.

```
<p style="color: green">Green text</p>
```

This approach should be used sparingly because it mixes style with content and therefore makes future changes more cumbersome. It can be useful as a quick way to test styles before they are moved out to an external style sheet.

## External style sheet

The most common way to include CSS is through an external style sheet. The style sheet rules are placed in a separate text file with a `.css` file extension. This style sheet is then referenced using the `<link>` element in the web page header. The `rel` (relationship) attribute must be set to `"stylesheet"` and the meta type attribute can optionally be set to `"text/css"`. The location of the style sheet is specified with the `href` attribute.

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

Another less common way to include an external style sheet is to use the `CSS@import` function from inside of the `<style>` element. For this function to work, it must be placed before any other rules.

```
<style type="text/css">
  @import url("MyStyle.css");
</style>
```

Using an external style sheet is often preferred because it completely separates CSS from the HTML document. It is then possible to quickly create a consistent look for an entire web site and to change its appearance just by editing a single CSS document. It also has performance benefits because external style sheets are cached and therefore need to be downloaded only once—for the first page a visitor views at your site.

## Testing environment

To experiment with CSS, you can use a simple text editor such as Notepad in Windows (found under Start ► Programs ► Accessories ► Notepad) or TextEdit on a Mac (found under Finder ► Applications ► TextEdit). Type the following single style rule into a new document in the editor. The rule will color the background of a web document red.

```
body { background: red; }
```

Save the file as `"MyStyle.css"` and then open another empty document. This new document will become the HTML file that uses the external style sheet you just created. Write the following HTML markup into the document, which includes a reference to the style sheet along with the minimal markup for a HTML 5 web document:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Example</title>
    <link rel="stylesheet" href="MyStyle.css">
  </head>
  <body>
```

```
<p>This page is red</p>
</body>
</html>
```

Go ahead and save this text file as “MyPage.html” in the same folder as the CSS file. You have now created a simple environment in which you can test CSS. To view the page, open MyPage.html with your web browser. You will see that the background is indeed colored red because of the rule in the style sheet.

## View source

While you have the browser opened, you can view the HTML markup that makes up the page by pressing Ctrl+U on a PC or Cmd+U on a Mac. This shortcut works in all major browsers, including Chrome, Firefox, and Internet Explorer (IE). You can also find the view source window by right-clicking on the page and selecting “View Source”. In Firefox and Chrome, the style sheet is clickable, allowing you to view the external style sheet rules that apply to the web page. (Note that in Chrome, you have to right-click the style sheet and select to open it because this file is stored on your local machine.)

Viewing the source code of web pages like this provides a great way to learn from other web developers. Whenever you find an interesting element on a web page—whether it is created with HTML, CSS or JavaScript—the page source will reveal how it was created.

## Comments

Comments in CSS are created by using the C-style notation (`/* */`). Everything placed between `/*` and `*/` will be ignored by browsers, even if the delimiters span multiple lines.

```
/* Multi-line
   Comment */
```

The main use of comments is to clarify the code to developers, including you in the future. They can also be used to improve readability by delimiting sections of the style sheet or providing meta data about the file, such as the author’s name.

```
/*
 * Meta data
 */

/** Section heading **/

p { margin-top: -1px; } /* clarification */
```

Comments are also useful for temporarily deactivating declarations or entire style rules for testing purposes.

```
p { /* color: white; */ }
```

## Whitespace

Whitespace refers to spaces, tabs, and new lines. You are free to format your style sheets however you like with whitespace to make them easier to read. One common formatting convention is to split declarations across multiple lines.

```
.fruit {  
  color: red;  
  margin: 1px;  
}
```

Another popular convention is to keep a rule's declarations in a single line and split the declarations into multiple lines only when they become too numerous.

```
.fruit      { color: red; margin: 1px; }  
.fruit.apple { color: green; margin: 2px; }
```

The formatting you use is a matter of preference. Choose the one that makes sense to you and aim to keep it consistent.