Contents

Foreword

The creation of this book has been for me, Mike McGrath, an exciting personal journey in discovering how CSS3 can really be implemented in today's web browsers. It has been fascinating to determine exactly which parts of the various modules of the CSS3 specifications are currently supported by Internet Explorer, Firefox, Google Chrome, Safari, and Opera. The "Handy Reference" section at the end of the book lists the current working CSS3 properties. All the examples I have given in this book demonstrate CSS3 features that are presently supported by leading web browsers and the screenshots illustrate the actual results produced by the listed code. I truly believe that now, more than ever, authors can integrate CSS3 presentation with JavaScript functionality and HTML5 document structure to produce stunning interactive web pages.

Conventions in this book

In order to clarify the code listed in the steps given in each example I have adopted certain colorization conventions. CSS selectors are colored blue, like this:

```
div#box > p
```

CSS properties and punctuation are also colored blue, but specified values are colored red:

```
{ color : red ; background : yellow }
```

CSS comments are colored green:

```
/* Validated by W3C CSS Validation Service */
```

HTML document content and markup tags are simply colored black:

```
<title>HTML5 in easy steps</title>
```

Additionally, in order to identify each source code file described in the steps a colored icon and the file name appears in the margin alongside the steps, such as these:

style.css page.html data.xml image.png

The source code of HTML documents used in the book's examples is not listed in full to avoid unnecessary repetition but the listed HTML code is the entire fragment of the document, including markup tags, to which the listed CSS code is applied.

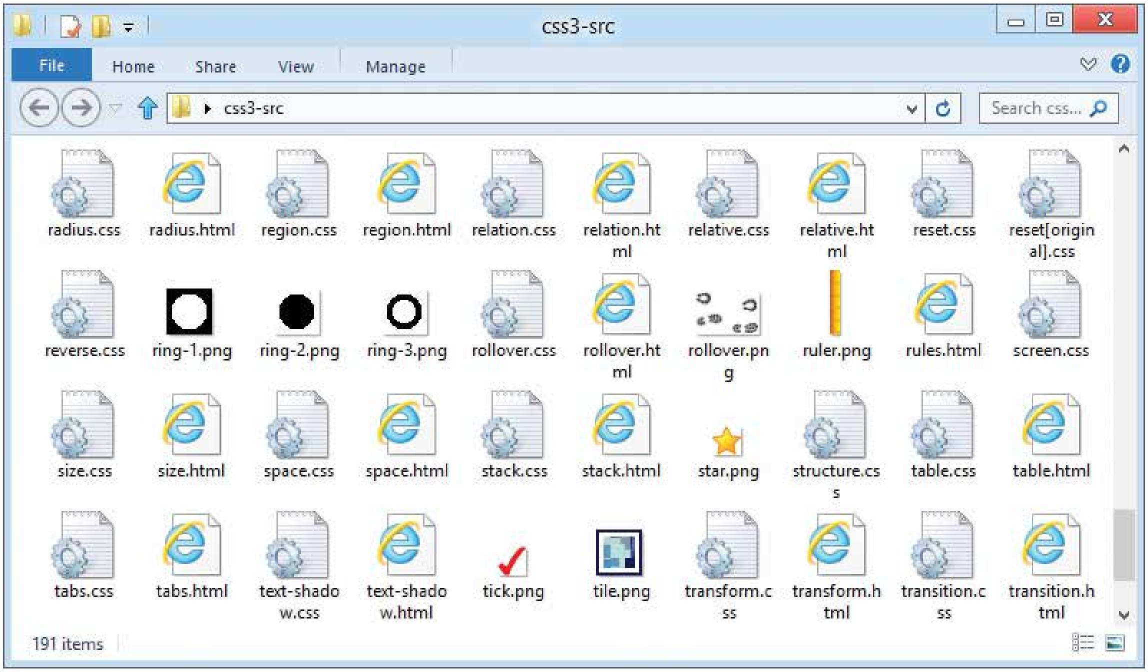
Grabbing the source code

For convenience I have placed all the source code files and associated files featured in this book into a single ZIP archive file. This contains all CSS style sheets, HTML documents, and image files used by the examples. You can obtain the complete archive by following these easy steps:

1. Browse to **http://www.ineasysteps.com**, then navigate to “Free Resources” and choose the “Downloads” section

2. Find “CSS3 in easy steps” in the list, then click on the hyperlink entitled “All Code Examples” to download the archive

3. Now extract the archive contents to any convenient location on your computer



Undoubtedly CSS3 provides significant new creative possibilities in web page authoring – as this book’s examples demonstrate. I sincerely hope you enjoy discovering how CSS3 can be used to create stunning interactive web pages in today’s latest web browsers as much as I did.

Mike McGrath

1 Getting started

Welcome to the exciting world of Cascading Style Sheets (CSS). This chapter demonstrates the various ways in which styles can be applied to HTML elements.

Introducing CSS

Cascading Style Sheets (CSS) is a language used to control the presentation of elements within HyperText Markup Language (HTML) documents. Presentation is specified by “styles” that may be assigned “inline” to HTML element **style** attributes, or by “rules” within **<style> </style>** tags in the HTML document’s head section, or as rules within separate style sheets. Each style rule selects specified elements then applies specified styles to them.

Sir Tim Berners Lee, W3C Director and inventor of the World Wide Web

CSS was created by the World Wide Web Consortium (W3C) to regain control of document markup as HTML grew from the initial few “tags” that merely defined the structural elements of a document – headings, paragraphs, hyperlinks, and lists. As further tags were added controlling images, text color, font size, and background color, it became recognized that the source code of many web pages often contained a great deal of markup for very little actual content.

The W3C offered a solution to regain control of document markup by separating their structural and presentational aspects. HTML tags would continue to control the structure but presentational aspects would now be contolled by “style rules” written in the Cascading Style Sheet (CSS) language. Besides distinguishing between structural and presentational aspects of a document, the CSS solution brings these additional benefits:

- **Easier maintenance** – a single style sheet can control multiple HTML documents, so changing appearance across an entire website is possible by editing just one style sheet
- **Smaller file sizes** – removal of all presentational markup from HTML produces smaller files, which download faster
- **Greater control** – margins, borders, padding, background color, and background images to any HTML element, and the appearance of certain parts of the interface, such as the cursor, can now be specified

The W3C is an international consortium whose members work together to develop web standards. The CSS home page can be found on the W3C website at **www.w3.org/Style/CSS**

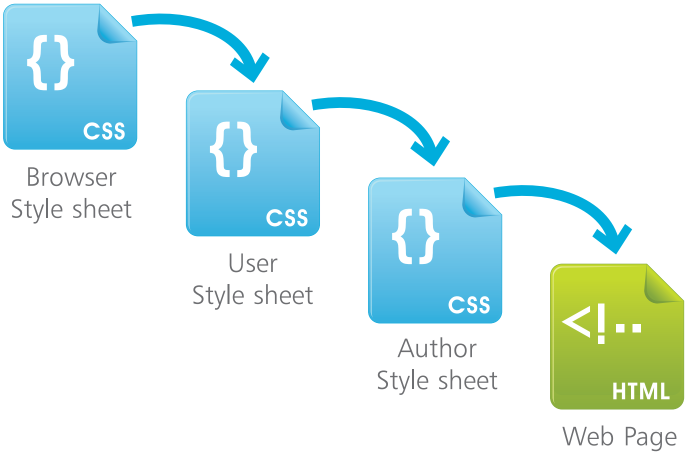
The CSS level 3 specification (CSS3) is divided into modules that allow enhancements such as rounded borders, drop-shadows, gradient color-fills, and animation effects – these and more are demonstrated by example in this book.

Understanding the cascade

The term “Cascading” in CSS describes the manner in which style rules can fall from one style sheet to another. The cascade determines which style rule will have precedence over others and be the one applied to the selected HTML element.

There are three basic types of style sheet that can specify style rules to be applied to HTML elements:

- **Browser (default) style sheet** – browsers employ an intrinsic set of style rules that they apply to all web pages by default. These vary slightly between different browsers but all have common features, such as black text and blue hyperlinks
- **User style sheet** – most browsers allow the user to specify their own appearance preferences, which effectively creates a custom style sheet that overrides the browser’s default style sheet
- **Author style sheet** – where the HTML document specifies a style sheet created by the web page author the browser will apply the style rules it contains, overriding both the user style sheet and the default browser style sheet



So the cascade means that the browser will typically apply the style rules in an author style sheet, if present. Otherwise it will apply the style rules in a user style sheet, if present. Otherwise it will apply the style rules in the browser’s style sheet by default.

Hot tip

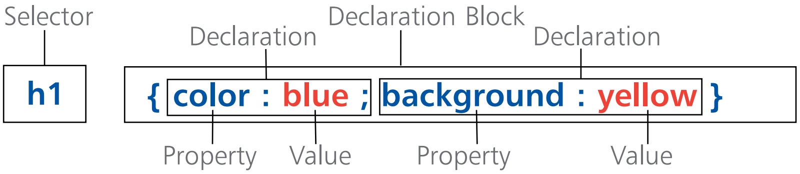
There are other style sheet languages, such as the eXtensible Stylesheet Language (XSL), but CSS is by far the most popular style sheet language.

Creating style rules

In CSS each style rule is comprised of TWO main parts:

- **Selector** – specifying which element/s of the HTML document are the target of that rule
- **Declaration Block** – specifying how properties of the selected target element should be styled

A style rule (or "style rule set") begins with the selector, followed by the declaration block within a pair of curly brackets (braces). The braces contain one or more declarations that each specify a property and a valid value for that property, as in this example:



Typically the selector targets (selects) a particular HTML element for styling – such as all **<h1>** heading elements in the document using the style rule set example above.

The declaration block in the example above contains two declarations to determine the foreground and background colors of the selected target elements. The CSS **color** property is assigned a **blue** value – so each **<h1>** heading element will have blue foreground text. Similarly, the CSS **background** property is assigned a **yellow** value – so each **<h1>** heading element will have a yellow background.

Notice how the CSS declaration uses a : colon character to assign a value to a property. Notice also that it requires the declarations to be separated by a **;** semi-colon character.

The final declaration in the declaration block does not need to be terminated by a semi-colon. However, some web page authors prefer to habitually terminate all CSS declarations so they need not remember to add a separating semi-colon when adding further declarations to an existing style rule set.

Don't forget

In conformance with the CSS specifications, the examples listed throughout this book do not add a semi-colon terminator after the final declaration in a declaration block.

1 When creating a new CSS style rule the author must initially specify a selector to target the HTML element to which the rule will be applied – strictly speaking, the CSS selector is everything that appears before the opening brace of the declaration block

```
h1
```

2 Next the declaration block must be created by adding a pair of braces after the selector

```
h1 {     }
```

3 Now a declaration can be inserted within the declaration block to assign a value to a property

```
h1 { color : blue }
```

4 A second declaration can then be added within the declaration block, separated from the first by a semi-colon

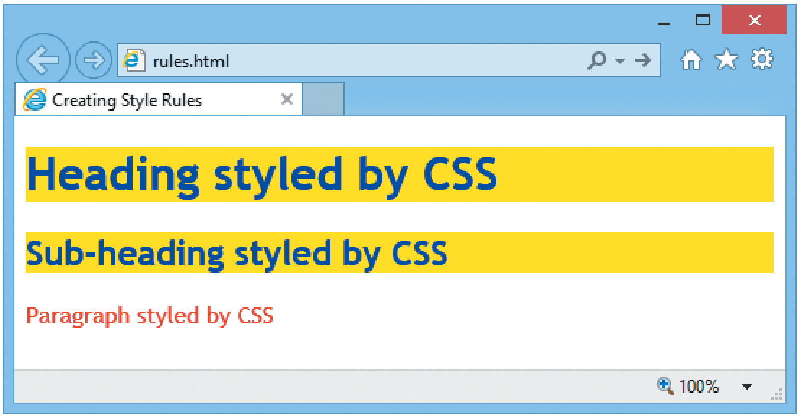
```
h1 { color : blue ; background : yellow }
```

5 The style rule set is now complete but can also be applied to another HTML element by extending the selector to become a comma-separated list

```
h1, h2 { color : blue ; background : yellow }
```

6 Further style rule sets can then be added below the first style rule set to target other elements

```
h1, h2  { color : blue ; background : yellow }
p       { color : red }
```



Heading styled by CSS

Sub-heading styled by CSS

Paragraph styled by CSS

Hot tip

Whitespace (spaces, tabs, line feeds, and carriage returns) is permitted within style rule sets to allow the author to format the style rules to their own preference. Typically style rule sets with fewer than four declarations are written on a single line, otherwise they are written across multiple lines for clarity.

Applying internal style rules

A style sheet is simply a collection of style rule sets to be applied to an HTML document. An internal style sheet can be created by inserting the style rule sets between **<style>** and **</style>** tags in the head section of the HTML document. The opening **<style>** tag should include a **type** attribute assigned a MIME type of "text/css" to describe the style sheet as using the CSS language. Optionally, this tag may also include a **media** attribute assigned a "screen" value to specifically describe the viewing medium as a color computer screen – although this is the default value if the **media** attribute is omitted.

All modern web browsers support CSS but if you wish to hide the internal style sheet from older browers the style rule sets can be enclosed within **<!--** and **-->** HTML comment tags.

Hot tip

MIME (Multipart Internet Mail Extension) types describe file types – **text/html** for HTML files, **text/javascript** for scripts, **text/css** for style sheets.

internal.html

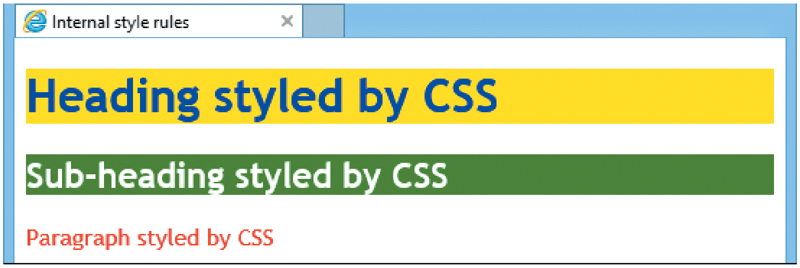
1. Create an HTML5 document containing heading, sub-heading, and paragraph elements

```
<!DOCTYPE HTML>
<html lang=”en”>
<head>
<meta charset=”UTF-8”>
<title>Internal style rules</title>
</head>
<body>
<h1>Heading styled by CSS</h1>
<h2>Sub-heading styled by CSS</h2>
<p>Paragraph styled by CSS</p>
</body>
</html>
```

2. In the head section of the document, immediately after the closing **</title>** tag, insert a style element block containing style rule sets for each content element

```
<style type=”text/css” media=”screen”>
<!--
h1      { color : blue ; background : yellow }
h2      { color : white ; background : green }
p       { color : red }
-->
</style>
```

3. Save the HTML file then open it in a web browser to see the styles applied to the content elements



Internal style rules

Heading styled by CSS

Sub-heading styled by CSS

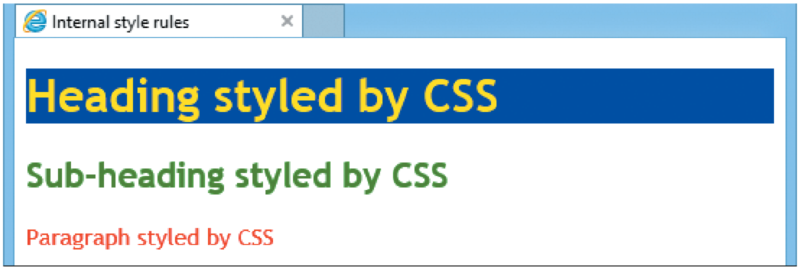
Paragraph styled by CSS

Style rules can also be applied internally to individual HTML elements by assigning "inline" properties and values to the **style** attribute of that element. This will override the rules applicable to the same properties of that element that may have been specified in any other style sheet.

4. Edit the heading elements to reverse the colors specified in the style sheet within the document's head section

```
<h1 style="color : yellow ; background : blue">
                                Heading styled by CSS</h1>
<h2 style="color : green ; background : white">
                             Sub-heading styled by CSS</h2>
```

5. Save the HTML file again then re-open it in a web browser to see the previous styles get overridden



Internal style rules

Heading styled by CSS

Sub-heading styled by CSS

Paragraph styled by CSS

Although described here for completeness there are drawbacks to using internal style rules. Multiple individual inline styles are more difficult to maintain than a single style rule in a style sheet as they would need to be individually altered to affect a style change to a number of elements. An internal style sheet is only applicable to the HTML document in which it is contained, whereas an external style sheet can apply to multiple web pages to consistently style an entire website from a single file.

Don't forget

Internal style sheets should only be used where it is desirable to create a single HTML document for portability – external style sheets, described overleaf, are always preferable.

Beware

The use of inline style rules should be avoided at all costs – always place style rules within a style sheet instead.

Linking an external style sheet

An external style sheet style is simply a collection of style rule sets listed in a plain text file, such as those created by Windows' Notepad application, then saved with a ".css" file extension. The style sheet can then be linked to HTML documents using a **<link>** tag. This must include a **type** attribute assigned a MIME type of "text/css" to denote using the CSS language, a **rel** attribute assigned a "stylesheet" relationship value, and an **href** attribute assigned the path to the style sheet file. Optionally, this tag may also include a **media** attribute assigned a "screen" value to describe the viewing medium as a color computer screen – although this is the default value if the **media** attribute is omitted.

external-1.html

external-2.html

1. Create two HTML documents containing heading and paragraph elements

```
<!DOCTYPE HTML>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>External style rules</title>
</head>
<body>
<h1>Heading styled by CSS</h1>
<p>Paragraph styled by CSS</p>
</body>
</html>
```

2. In the head section of each HTML document, immediately after the closing **</title>** tag, insert a link element specifying a style sheet to be used

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

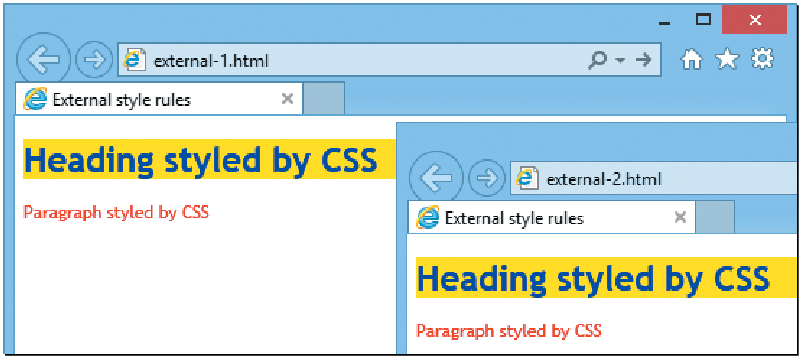
external.css

3. Save the HTML files then open a plain text editor, such as Notepad, and list style rule sets for content elements

```
h1 { color : blue ; background : yellow }
p  { color : red ; background : white }
```

4. Save the CSS file alongside the HTML files, named as "external.css", then open the HTML files in a web browser to see the styles applied to each web page

...cont'd



Heading styled by CSS

Paragraph styled by CSS

Heading styled by CSS

Paragraph styled by CSS

Single line or multi-line comments can be added to CSS style sheets between /* and */ characters. These are ignored by web browsers but are useful to describe aspects of the style sheet.

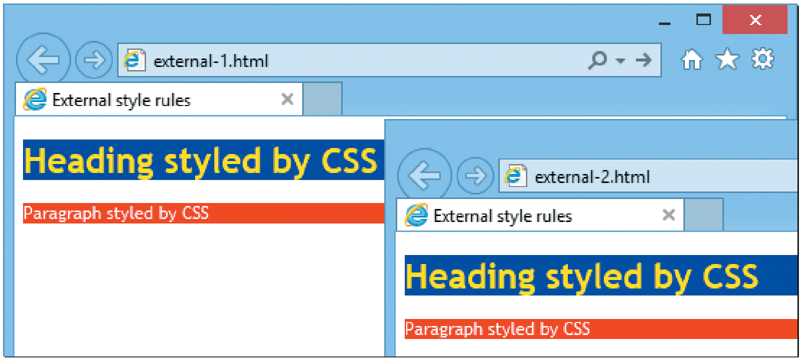
5 Re-open the style sheet file in a plain text editor and insert a commented title at the very start of the file

```
/* Master Style Sheet for external-x.html Pages */
```

6 Now edit the style rule sets to reverse the previous colors

```
h1 { color : yellow ; background : blue }
p  { color : white ; background : red }
```

7 Save the style sheet file once more then re-open (or refresh) each HTML page in a web browser to see modified style rules applied to both web pages



Heading styled by CSS

Paragraph styled by CSS

Heading styled by CSS

Paragraph styled by CSS

Hot tip

CSS comments can appear both outside and within rule sets – so can be used to "comment-out" entire rules or individual declarations.

Don't forget

Always use external style sheets rather than internal style sheets or inline style rules – to cleanly separate the structural and presentational aspects of web pages.

Importing other style sheets

Multiple external style sheets can be used to control different aspects of HTML documents using **@import** directives to specify the path to each CSS file. These must be placed before any other rules in the style sheet and be terminated by a semi-colon.

import.html

1 Create an HTML document containing heading and paragraph elements

```
<!DOCTYPE HTML>
<html lang=”en”>
<head>
<meta charset=”UTF-8”>
<title>Importing style sheets</title>
</head>
<body>
<h1>Heading styled by CSS</h1>
<h2>Sub-heading styled by CSS</h2>
<p>Paragraph styled by CSS</p>
</body>
</html>
```

2 In the head section of the HTML document, immediately after the closing **</title>** tag, insert a link to a “master” style sheet

```
<link type=”text/css”
      rel=”stylesheet” href=”import.css” media=”screen”>
```

import.css

3 Save the HTML file then open a plain text editor and create the “master” style sheet that simply imports two other style sheets to style the headings and paragraphs within the web page

```
@import “headings.css” ;
@import “paragraphs.css” ;
```

paragraphs.css

4 Save the master style sheet as “import.css” alongside the HTML file then create a style sheet to specify the appearance of paragraphs

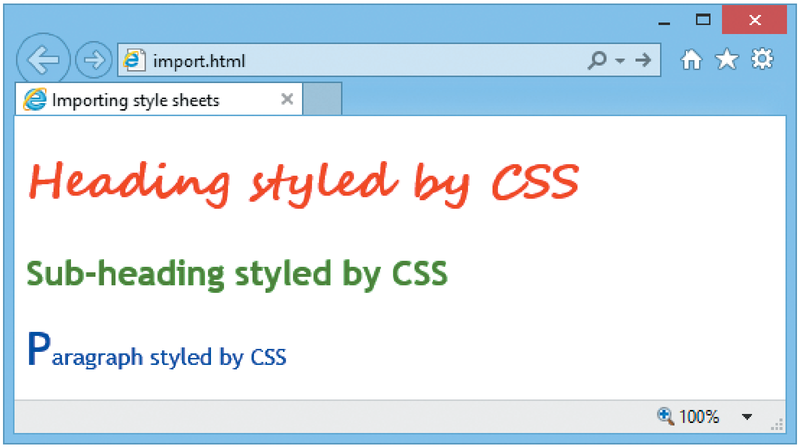
```
p { color : blue }
p:first-letter { font-size : xx-large }
```

headings.css

5 Save the second style sheet as “paragraphs.css” alongside the HTML file then create another style sheet to specify the appearance of headings

```
h1 { color : red ; font-family : “Script”, cursive }
h2 { color : green }
```

6. Save the third style sheet as "headings.css" alongside the HTML file then open the web page in a browser to see the content elements styled by the style sheets imported via the master style sheet



Heading styled by CSS

Sub-heading styled by CSS

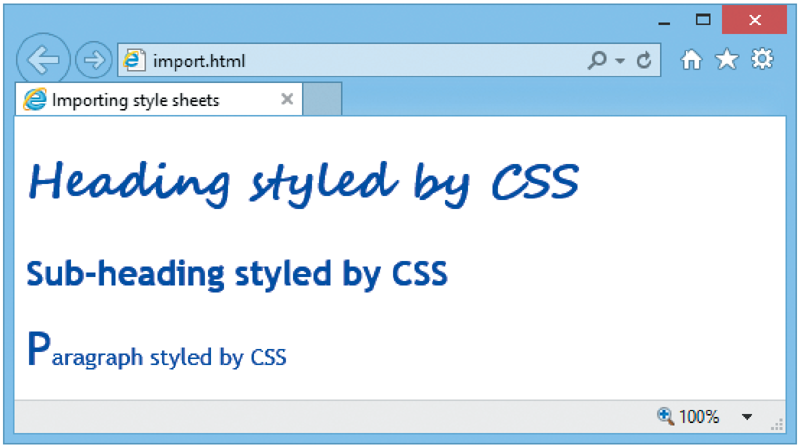
Paragraph styled by CSS

Beware

Omitting the semi-colon after an **@import** directive causes the browser to stop reading the style sheet.

7. Edit the "headings.css" style sheet to match all heading colors to the paragraph color then refresh the browser to see the changes

```
h1 { color : blue ; font-family : "Segoe Script", cursive }
h2 { color : blue }
```



Heading styled by CSS

Sub-heading styled by CSS

Paragraph styled by CSS

Hot tip

The **@import** directive can be used to hide style sheets from older browsers that do not recognize that instruction.

Summary

- CSS is a language provided by the W3C to regain control of markup by separating document structure from presentation
- The cascade allows style rules to fall from one style sheet to another and determines which style rule will be applied
- Each style rule comprises of a selector and a declaration block
- Each declaration specifies a property and a value to be applied to that property, separated by a : colon character
- A style rule set contains multiple declarations, each separated from the next by a ; semi-colon character
- The final declaration in a declaration block need not be terminated with a ; semi-colon character
- An internal style sheet is a collection of style rules contained within **<style>** tags in the head section of an HTML document
- Style rules can also be applied inline by assigning properties and values to the **style** attribute of an HTML element
- Inline style rules are difficult to maintain and should be avoided
- External style sheets are recommended to cleanly separate structure and presentation – internal style sheets should only be created in special circumstances
- An external style sheet is a collection of style rule sets listed in a plain text file saved with a **.css** file extension
- An HTML document can link an external style sheet file by adding a **<link>** element in the document's head section
- Each **<link>** element must contain **rel**, **type**, and **href** attributes
- The MIME type of CSS is **text/css**
- All **@import** directives must appear before other rules in a style sheet and must be terminated with a ; semi-colon character