

# Contents

## 1

### Getting started

7

Introducing CSS	8
Understanding the cascade	9
Creating style rules	10
Applying internal style rules	12
Linking an external style sheet	14
Importing other style sheets	16
Summary	18

## 2

### Selecting targets

19

Selecting by element type	20
Selecting by class	22
Selecting by identity	23
Selecting descendants	24
Selecting by relationship	26
Selecting by attribute	28
Selecting partial attributes	30
Evaluating importance	32
Summary	34

## 3

### Styling boxes

35

Recognizing content boxes	36
Changing display formats	38
Sizing the content area	40
Controlling borders	42
Adding padding	44
Setting margins	46
Painting colors	48
Repeating backgrounds	50
Positioning backgrounds	52
Summary	54

## 4

## Controlling layout

55

Centering content boxes	56
Positioning boxes absolutely	58
Stacking content boxes	60
Positioning boxes relatively	62
Fixing constant positions	64
Floating content boxes	66
Clipping & handling overflow	68
Constructing columns	70
Summary	72

## 5

## Formatting text

73

Suggesting a font	74
Specifying font size	76
Adjusting font weight	78
Varying font styles	80
Using the font shorthand	82
Aligning text	84
Indenting & spacing text	86
Decorating text	88
Governing space & direction	90
Summary	92

## 6

## Arranging data

93

Setting table columns	94
Spacing table cells	96
Collapsing table borders	98
Assigning table features	100
Choosing list markers	102
Positioning list markers	104
Summary	106

## 7

## Generating effects

107

Inserting text enhancements	108
Numbering document sections	110
Highlighting important content	112
Providing special cursors	114
Outlining page components	116
Indicating current focus	118
Displaying hyperlink status	120
Reacting to user events	122
Interacting with CSS buttons	124
Moving CSS tabs	126

Hiding & revealing elements	128
Summary	130

## 8

### Following guidelines

131

Resetting browser defaults	132
Organizing your code	134
Optimizing style rules	136
Employing multiple classes	138
Validating style sheets	140
Compressing code files	142
Summary	144

## 9

### Customizing pages

145

Recognizing media	146
Adding page structure	148
Specifying default styles	150
Specifying reverse styles	152
Improving readability	154
Reverting to legacy styles	156
Providing handheld styles	158
Setting printer styles	160
Summary	162

## 10

### Looking ahead

163

Rounding corners	164
Reducing opacity	166
Adding shadows	168
Creating resizable containers	170
Summary	172

## +

### Reference section

173

Properties and values	174
Selectors	181
Pseudo-classes	184
Pseudo-elements	186

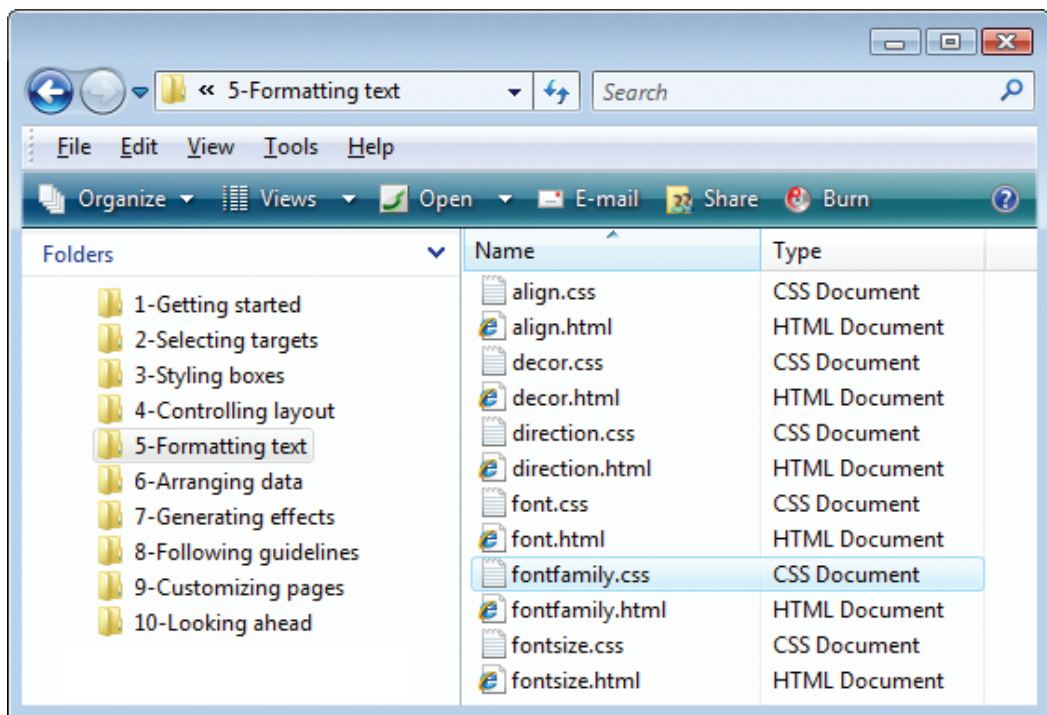
### Index

187

# Foreword

The examples in this book have been carefully prepared to demonstrate Cascading Style Sheets. You are encouraged to try out the examples on your own computer to discover the exciting possibilities offered by CSS. The straightforward descriptions should allow you to easily recreate the examples manually or, if you prefer, you can download an archive containing all the source code by following these simple steps:

- 1 Open your browser and visit our website at <http://www.ineasysteps.com>
- 2 Navigate to the “Resource Center” and choose the “Downloads” section
- 3 Find the “From CSS in easy steps, 2nd edition” item in the “Source Code” list, then click on the hyperlink entitled “All Code Examples” to download the ZIP archive
- 4 Extract the contents of the ZIP archive to any convenient location on your computer – for easy reference these are arranged in sub-folders whose names match each chapter title of this book. The documents are named as described in the book and are located in the appropriate chapter folder of the archive. For example, the **fontfamily.css** style sheet, listed in the fifth chapter, is located in the folder named **5-Formatting text**



# 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.*

- 8** Introducing CSS
- 9** Understanding the cascade
- 10** Creating style rules
- 12** Applying internal style rules
- 14** Linking an external style sheet
- 16** Importing other style sheets
- 18** Summary

# Introducing CSS



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

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.

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. Much interest in the Internet arose when the `<img>` tag was introduced, adding the capability to display images alongside text, so websites began to proliferate. Web content authors increasingly began to demand more ways to control how different web page elements should appear, such as bold and italic text – so `<b>` and `<i>` tags were added to HTML. Further tags were also added controlling text color, size, and background color, until 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 controlled 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:



The W3C is an international consortium whose members work together to develop web standards. The CSS specifications can be found on their website at [www.w3.org/TR/CSS21](http://www.w3.org/TR/CSS21)

- **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

# 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.

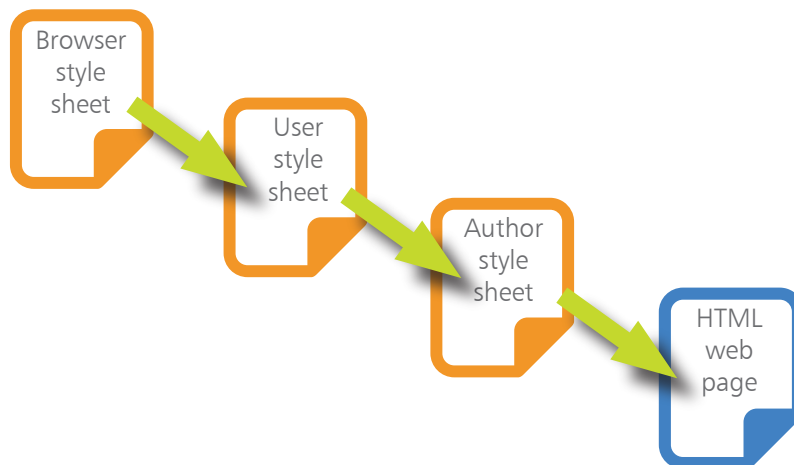
There are three basic types of style sheet that can specify style rules to be applied to HTML elements:

- **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

## 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.



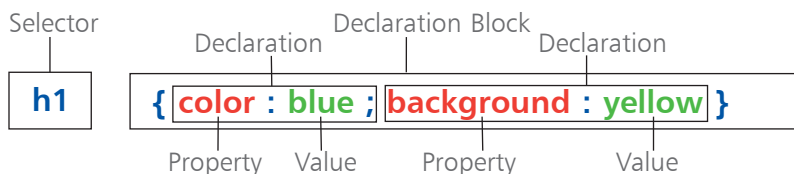
So the cascade means that the browsers will typically apply the style rules in an author style sheet, if present, otherwise it will apply the style rules in a user style sheet, if present, otherwise it will apply the style rules in the browser’s style sheet by default.

# Creating style rules

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

- 1 **Selector** – specifying which element/s of the HTML document are the target of that rule
- 2 **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.



## ...cont'd

- 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 }**

### 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.

#### Creating Style Rules

**Heading styled by CSS**

**Sub-heading styled by CSS**

**Paragraph styled by CSS**

**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

## Applying internal style rules

A style sheet is simply a collection of style rule sets to be applied to a 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 browsers the style rule sets can be enclosed within **<!--** and **-->** HTML comment tags.

- 1 Create a HTML document containing heading, sub-heading, and paragraph elements
 

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">

<html>
<head>
<meta http-equiv="Content-Type"
    content="text/html; charset=ISO-8859-1">

<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

...cont'd



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



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 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



external.css

- 1 Create two HTML documents containing heading and paragraph elements  

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

```
<html>
<head>
<meta http-equiv="Content-Type"
      content="text/html; charset=ISO-8859-1">

<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  

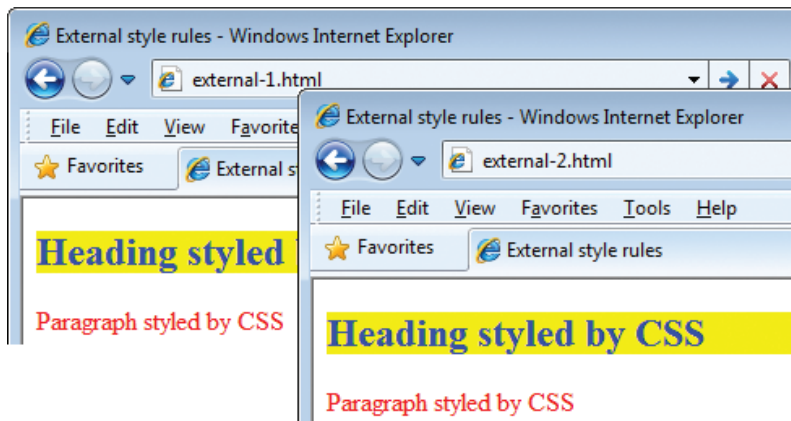
```
<link type="text/css"
      rel="stylesheet" href="external.css" media="screen">
```

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

```
h2 { 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



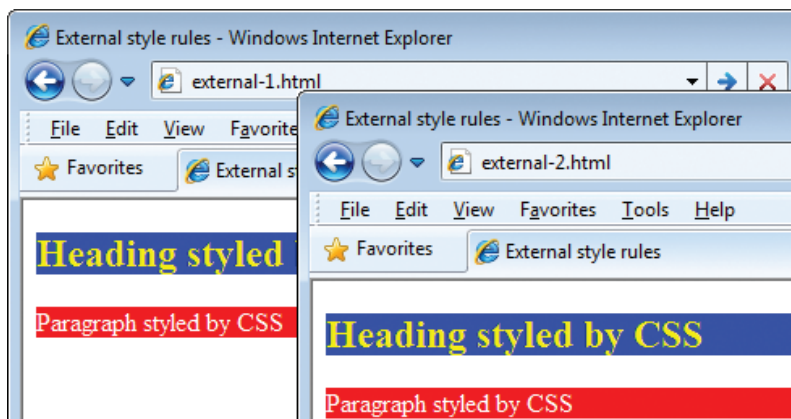
### Hot tip



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

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  
`h2 { 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



### 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 a HTML document containing heading and paragraph elements

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">

<html>
<head>
<meta http-equiv="Content-Type"
    content="text/html; charset=ISO-8859-1">
<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 }
```

...cont'd

- 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 : "lucida handwriting", 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



- 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 : "lucida handwriting", cursive }
h2 { color : blue }
```



headings.css

### Beware



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

### 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 a HTML document
- Style rules can also be applied inline by assigning properties and values to the **style** attribute of a 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