



# CSS & CSS3

**20 Lessons to Successful Web Development**

**Robin Nixon**



**Mc  
Graw  
Hill  
Education**

**Features online video tutorials for each lesson!**



# CSS & CSS3

20 Lessons to Successful Web Development

## About the Author

**Robin Nixon** is a prolific author on programming and web development (as well as psychology and motivation), with his books having been translated into numerous foreign languages—frequently topping the US and international computer book charts. He has worked with computers and technology for all his life, and began writing on the subject about 35 years ago.

He has authored hundreds of articles, and over two dozen books, and is a popular video and online instructor, with thousands of students taking his courses. Robin is also an accomplished programmer, developer, and entrepreneur, with several successful Internet startups to his name, from which he has learned a wealth of programming hints and tips, which he enjoys passing on in his expanding range of web development books, including the following titles:

- *HTML5: 20 Lessons to Successful Web Development* (McGraw-Hill Education, 2015)
- *JavaScript: 20 Lessons to Successful Web Development* (McGraw-Hill Education, 2015)
- *PHP: 20 Lessons to Successful Web Development* (McGraw-Hill Education, 2015)
- *Learning PHP, MySQL, JavaScript, CSS & HTML5* (O'Reilly, 2014)
- *Web Developer's Cookbook* (McGraw-Hill Education, 2012)
- *HTML5 for iOS and Android* (McGraw-Hill Education, 2010)

## About the Technical Editor

**Albert Wiersch** has been writing software since the Commodore VIC-20 and Commodore 64 days in the early 1980s. He holds a Bachelor of Science degree in Computer Science Engineering and an MBA from the University of Texas at Arlington. Albert currently develops and sells software that helps web developers, educators, students, businesses, and government agencies check their HTML & CSS documents and their websites for quality problems, including many SEO (search engine optimization), mobility, and accessibility issues, with discounts made available to students. His website is at [HTMLValidator.com](http://HTMLValidator.com).

# CSS & CSS3

20 Lessons to Successful Web Development

Robin Nixon

**Mc  
Graw  
Hill**  
Education

New York Chicago San Francisco  
Athens London Madrid Mexico City  
Milan New Delhi Singapore Sydney Toronto

Copyright © 2015 by McGraw-Hill Education. All rights reserved. Except as permitted under the United States Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher, with the exception that the program listings may be entered, stored, and executed in a computer system, but they may not be reproduced for publication.

ISBN: 978-0-07-185021-6

MHID: 0-07-185021-X

The material in this eBook also appears in the print version of this title: ISBN: 978-0-07-184996-8, MHID: 0-07-184996-3.

eBook conversion by codeMantra  
Version 1.0

All trademarks are trademarks of their respective owners. Rather than put a trademark symbol after every occurrence of a trademarked name, we use names in an editorial fashion only, and to the benefit of the trademark owner, with no intention of infringement of the trademark. Where such designations appear in this book, they have been printed with initial caps.

McGraw-Hill Education eBooks are available at special quantity discounts to use as premiums and sales promotions or for use in corporate training programs. To contact a representative, please visit the Contact Us page at [www.mhprofessional.com](http://www.mhprofessional.com).

Information has been obtained by McGraw-Hill Education from sources believed to be reliable. However, because of the possibility of human or mechanical error by our sources, McGraw-Hill Education, or others, McGraw-Hill Education does not guarantee the accuracy, adequacy, or completeness of any information and is not responsible for any errors or omissions or the results obtained from the use of such information.

#### TERMS OF USE

This is a copyrighted work and McGraw-Hill Education and its licensors reserve all rights in and to the work. Use of this work is subject to these terms. Except as permitted under the Copyright Act of 1976 and the right to store and retrieve one copy of the work, you may not decompile, disassemble, reverse engineer, reproduce, modify, create derivative works based upon, transmit, distribute, disseminate, sell, publish or sublicense the work or any part of it without McGraw-Hill Education's prior consent. You may use the work for your own noncommercial and personal use; any other use of the work is strictly prohibited. Your right to use the work may be terminated if you fail to comply with these terms.

THE WORK IS PROVIDED "AS IS." MCGRAW-HILL EDUCATION AND ITS LICENSORS MAKE NO GUARANTEES OR WARRANTIES AS TO THE ACCURACY, ADEQUACY OR COMPLETENESS OF OR RESULTS TO BE OBTAINED FROM USING THE WORK, INCLUDING ANY INFORMATION THAT CAN BE ACCESSED THROUGH THE WORK VIA HYPERLINK OR OTHERWISE, AND EXPRESSLY DISCLAIM ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. McGraw-Hill Education and its licensors do not warrant or guarantee that the functions contained in the work will meet your requirements or that its operation will be uninterrupted or error free. Neither McGraw-Hill Education nor its licensors shall be liable to you or anyone else for any inaccuracy, error or omission, regardless of cause, in the work or for any damages resulting therefrom. McGraw-Hill Education has no responsibility for the content of any information accessed through the work. Under no circumstances shall McGraw-Hill Education and/or its licensors be liable for any indirect, incidental, special, punitive, consequential or similar damages that result from the use of or inability to use the work, even if any of them has been advised of the possibility of such damages. This limitation of liability shall apply to any claim or cause whatsoever whether such claim or cause arises in contract, tort or otherwise.

*To Julie*

*This page intentionally left blank*

# Contents at a Glance

<b>PART I</b>	<b>Basic CSS</b>	
1	Introduction to CSS	3
2	Learning the CSS Rules	11
3	Applying Declarations to IDs and Classes	19
4	Accessing Selectors	31
5	Working with the Cascade	43
6	Selecting Fonts and Typography	55
7	Manipulating Color and Position	67
8	Handling Pseudo-Selectors and Using Shorthand Properties	87
9	Understanding the Box Model	107
<b>PART II</b>	<b>CSS3</b>	
10	Introduction to CSS3	121
11	Using Selectors and Attribute Selectors	133
12	Setting Backgrounds	141
13	Attaching Borders	153
14	Controlling Box Shadows, Overflow, and Columns	165
15	Adding Colors and Opacity	177
16	Creating Text Effects and Changing the Box Model	189
17	Linking to Web Fonts	203
18	Making 2D Transformations	211
19	Applying Specific Transformations	225
20	Directing 3D Transformations	237
A	Answers to the Self-Test Questions	257
	Index	273



*This page intentionally left blank*

# Contents

Acknowledgments .....	xvii
Introduction .....	xix

## **PART I Basic CSS**

---

<b>LESSON 1 Introduction to CSS .....</b>	<b>3</b>
How the Document Object Model Works .....	3
Correct HTML Structure and Nesting .....	6
About Cascading Style Sheets .....	6
Importing a Style Sheet .....	7
Importing CSS from Within HTML .....	7
Embedded Style Settings .....	8
Using IDs .....	8
Using Classes .....	8
Summary .....	9
Self-Test Questions .....	9
<b>LESSON 2 Learning the CSS Rules .....</b>	<b>11</b>
Selectors .....	11
IDs .....	12
Classes .....	12
Multiple Assignments .....	12
Comments .....	13
Single-Line Comments .....	13
Style Types .....	14
Default Styles .....	14
User Styles .....	14
External Style Sheets .....	15
Internal Styles .....	15
Inline Styles .....	16
It's All About Separation of Style and Content .....	16
Summary .....	17
Self-Test Questions .....	17

## x Contents

<b>LESSON 3 Applying Declarations to IDs and Classes</b> . . . . .	<b>19</b>
Using IDs . . . . .	19
A Specific Case . . . . .	20
IDs Are Single-Use Only . . . . .	21
Using Classes . . . . .	23
Assigning Multiple Classes . . . . .	25
Accessing IDs and Classes with JavaScript . . . . .	26
The O () Function . . . . .	26
The S () Function . . . . .	27
The C () Function . . . . .	27
Using Hyphenated Property Names . . . . .	27
Why Use JavaScript? . . . . .	28
Summary . . . . .	29
Self-Test Questions . . . . .	29
<b>LESSON 4 Accessing Selectors</b> . . . . .	<b>31</b>
The Type Selector . . . . .	31
The Descendant Selector . . . . .	32
The Child Selector . . . . .	32
The ID Selector . . . . .	33
Narrowing ID Scope . . . . .	33
The Class Selector . . . . .	33
Narrowing Class Scope . . . . .	34
The Attribute Selector . . . . .	34
The Universal Selector . . . . .	34
Selecting by Group . . . . .	35
For JavaScript Programmers . . . . .	36
Points of Interest . . . . .	41
Summary . . . . .	42
Self-Test Questions . . . . .	42
<b>LESSON 5 Working with the Cascade</b> . . . . .	<b>43</b>
Style Sheet Creators . . . . .	43
Style Sheet Methods . . . . .	44
Style Sheet Selectors . . . . .	44
Calculating Specificity . . . . .	45
When Rules Are Particularly Important . . . . .	46
So What Are Divs and Spans Anyway? . . . . .	48
Specifying Measurements . . . . .	50
Pixels . . . . .	50
Points . . . . .	51
Inches . . . . .	51
Centimeters . . . . .	51
Millimeters . . . . .	51

Picas .....	51
Ems .....	52
Exs .....	52
Percent .....	52
Summary .....	53
Self-Test Questions .....	54
<b>LESSON 6 Selecting Fonts and Typography .....</b>	<b>55</b>
Font Family .....	55
Generic Font Families .....	56
Font Style .....	57
Font Size .....	58
Font Weight .....	60
Changing Text Styles .....	61
Text Decoration .....	61
Line, Word, and Letter Spacing .....	63
Horizontal Alignment .....	64
Managing Uppercase and Lowercase .....	65
Text Indenting .....	65
Summary .....	65
Self-Test Questions .....	65
<b>LESSON 7 Manipulating Color and Position .....</b>	<b>67</b>
Colors .....	67
Using <code>rgb()</code> .....	69
Short Color Strings .....	69
Gradients .....	69
Linear Gradients .....	70
Radial Gradients .....	71
Controlling Color Stops .....	71
Repeating Gradients .....	72
An Example Document .....	73
Alpha Transparency .....	75
An Example Document .....	75
CSS3 Browser Incompatibilities .....	76
Older Browsers .....	77
Let George Do It .....	77
The <i>prefixfree.js</i> Utility .....	78
Positioning Elements .....	80
Absolute Positioning .....	80
Relative Positioning .....	81
Fixed Positioning .....	81
An Example Document .....	83
Summary .....	84
Self-Test Questions .....	85

## xii Contents

<b>LESSON 8 Handling Pseudo-Selectors and Using Shorthand Properties . . . . .</b>	<b>87</b>
Introducing the Pseudo-Selectors . . . . .	87
Link-Related Pseudo-Classes . . . . .	90
Accessing Elements Numerically . . . . .	92
Selecting Elements by Numeric Location . . . . .	95
Relational Selectors . . . . .	97
Miscellaneous Pseudo-Selectors . . . . .	99
Pseudo-Elements That Add Content . . . . .	100
Shorthand Properties . . . . .	101
Border and Outline Shorthand . . . . .	102
Background Shorthand . . . . .	103
Font Shorthand . . . . .	104
Margin and Padding Shorthand . . . . .	104
List Shorthand . . . . .	105
Summary . . . . .	105
Self-Test Questions . . . . .	106
<b>LESSON 9 Understanding the Box Model . . . . .</b>	<b>107</b>
Visualizing the Box Model . . . . .	107
Setting Margins . . . . .	108
Using Borders . . . . .	111
Applying Padding . . . . .	112
Non-image Elements . . . . .	112
The Object Contents . . . . .	116
Changing the Box Model . . . . .	117
Summary . . . . .	118
Self-Test Questions . . . . .	118
<b>PART II CSS3</b>	
<b>LESSON 10 Introduction to CSS3 . . . . .</b>	<b>121</b>
Attribute Selectors . . . . .	122
Improved Backgrounds . . . . .	122
Enhanced Borders . . . . .	123
Box Shadows . . . . .	125
Element Overflow . . . . .	125
Multicolumn Layout . . . . .	127
Extended Color Handling and Opacity . . . . .	128
Additional Text Effects . . . . .	129
Box Sizing . . . . .	129
Resizing and Outlines . . . . .	129
Web Fonts . . . . .	130
Transitions and Transformations . . . . .	130
Summary . . . . .	132
Self-Test Questions . . . . .	132

<b>LESSON 11 Using Selectors and Attribute Selectors</b> .....	<b>133</b>
Summary of Selector Types .....	133
The Universal Selector .....	133
Type Selectors .....	134
Class Selectors .....	134
ID Selectors .....	134
Descendant Selectors .....	135
Child Selectors .....	135
Adjacent Sibling Selectors .....	135
General Sibling Selectors .....	135
Attribute Selectors .....	136
Pseudo-Classes .....	136
Pseudo-Elements .....	136
Attribute Selectors .....	137
The ^= Operator .....	138
The \$= Operator .....	138
The *= Operator .....	138
Summary .....	139
Self-Test Questions .....	139
<b>LESSON 12 Setting Backgrounds</b> .....	<b>141</b>
The background-clip Property .....	141
The background-origin Property .....	146
Compatibility Issues .....	146
The background-size Property .....	147
Using the auto Value .....	147
Multiple Backgrounds .....	147
Summary .....	150
Self-Test Questions .....	151
<b>LESSON 13 Attaching Borders</b> .....	<b>153</b>
The border-color Properties .....	153
The border-style Properties .....	154
The border-image Properties .....	156
The border-radius Properties .....	160
Summary .....	163
Self-Test Questions .....	163
<b>LESSON 14 Controlling Box Shadows, Overflow, and Columns</b> .....	<b>165</b>
Adding Box Shadows .....	165
The box-shadow Supported Values .....	168
Managing Element Overflow .....	170
Flowing Text over Multiple Columns .....	172
Summary .....	175
Self-Test Questions .....	175

<b>LESSON 15 Adding Colors and Opacity</b> .....	<b>177</b>
HSL Colors .....	177
HSLA Colors .....	182
RGB Colors .....	182
RGBA Colors .....	186
The opacity Property .....	186
Summary .....	188
Self-Test Questions .....	188
<b>LESSON 16 Creating Text Effects and Changing the Box Model</b> .....	<b>189</b>
The text-shadow Property .....	189
The text-shadow Supported Values .....	191
The text-overflow Property .....	191
The word-wrap Property .....	193
The box-sizing Property .....	194
Browser Compatibility .....	197
The resize Property .....	198
The outline and outline-offset Properties .....	200
Summary .....	201
Self-Test Questions .....	201
<b>LESSON 17 Linking to Web Fonts</b> .....	<b>203</b>
Loading a Web Font .....	203
The Main Types of Web Fonts .....	204
Loading a Web Font .....	204
And the Winner Is... .....	205
Google Web Fonts .....	205
Browser Differences .....	208
Other Font Sources .....	209
Summary .....	210
Self-Test Questions .....	210
<b>LESSON 18 Making 2D Transformations</b> .....	<b>211</b>
2D Transformations .....	211
The none Transformation Type .....	211
The matrix(w, s1, s2, h, x, y) Function .....	213
Argument 1 .....	215
Argument 2 .....	216
Argument 3 .....	217
Argument 4 .....	218
Argument 5 .....	219
Argument 6 .....	220
Internet Explorer Prior to IE9 .....	222
Summary .....	223
Self-Test Questions .....	223

<b>LESSON 19 Applying Specific Transformations</b> .....	<b>225</b>
The <code>translate(x, y)</code> Function .....	225
The <code>scale(x, y)</code> Function .....	227
Older Versions of IE .....	229
The <code>rotate(n)</code> Function .....	229
Older Versions of IE .....	231
The <code>skew(x, y)</code> Function .....	232
Older Versions of IE .....	234
Summary .....	235
Self-Test Questions .....	235
<b>LESSON 20 Directing 3D Transformations</b> .....	<b>237</b>
The <code>perspective</code> Property .....	237
The <code>transform-origin</code> Property .....	239
About the Three Dimensions .....	241
The <code>translate3d(x, y, z)</code> Function .....	242
The <code>scale3d(x, y, z)</code> Function .....	244
The <code>rotate3d(x, y, z, r)</code> Function .....	246
The <code>backface-visibility</code> Property .....	249
The <code>transform-style</code> Property .....	251
Transitions .....	251
The <code>transition-property</code> Property .....	252
The <code>transition-duration</code> Property .....	252
The <code>transition-delay</code> Property .....	252
The <code>transition-timing-function</code> Property .....	253
The <code>cubic-bezier()</code> Function .....	253
Shorthand <code>transition</code> Property Syntax .....	254
Summary .....	256
Self-Test Questions .....	256
<b>APPENDIX Answers to the Self-Test Questions</b> .....	<b>257</b>
Lesson 1 Answers .....	257
Lesson 2 Answers .....	258
Lesson 3 Answers .....	258
Lesson 4 Answers .....	259
Lesson 5 Answers .....	260
Lesson 6 Answers .....	261
Lesson 7 Answers .....	262
Lesson 8 Answers .....	262
Lesson 9 Answers .....	263
Lesson 10 Answers .....	264
Lesson 11 Answers .....	265
Lesson 12 Answers .....	266
Lesson 13 Answers .....	266
Lesson 14 Answers .....	267



**xvi** Contents

Lesson 15 Answers .....	268
Lesson 16 Answers .....	269
Lesson 17 Answers .....	269
Lesson 18 Answers .....	270
Lesson 19 Answers .....	271
Lesson 20 Answers .....	272
<b>Index</b> .....	<b>273</b>

# Acknowledgments

Once again I would like to thank the amazing team at McGraw-Hill Education, with whom it is always a real pleasure to work on new book projects. In particular, I would like to thank my Sponsoring Editor Brandi Shailer, Amanda Russell for overseeing the project's development, Editorial Supervisor Jody McKenzie, Production Supervisor Jean Bodeaux, Copy Editor Megha Saini, and Jeff Weeks for the excellent cover design. Thanks also goes again to Albert Wiersch (whom I have had the pleasure of working with on a number of occasions) for his meticulous eye for detail during technical review.

*This page intentionally left blank*

# Introduction

## Why This Book?

The concept for this book grew out of Robin's extremely popular online courses in which thousands of students are enrolled. From their feedback it became evident that the reason for this popularity was that students love the way the material is broken down into easy-to-digest lessons, each of which can be completed in an hour or less. They also like the no-nonsense, jargon-free, and friendly writing style.

Now, working together, Robin and McGraw-Hill Education have further revised, updated, and developed his CSS & CSS3 course into this book, which not only will teach you everything you need to learn in 20 lessons (of less than an hour each), it also includes an average 15-minute detailed video walk-through for each lesson—comprising 5 hours of footage in total.

Watch the videos after reading the lesson to reinforce key concepts, or use the video as a primer to working through each print lesson. Used together, these course materials make learning CSS & CSS3 easier than it has ever been, and is the ideal way for you to add these essential skills to your web development toolkit.



To view the accompanying video for this lesson, please visit [mhprofessional.com/nixoncss/](http://mhprofessional.com/nixoncss/).

## Who Should Read This Book?

Each chapter is laid out in a straightforward and logical manner as a lesson, with plenty of examples written using simple and clear CSS. Before moving on to each subsequent lesson, you have the opportunity to test your new knowledge with a set of 10 questions about what you have just learned. You can also work along with every lesson by watching its accompanying video tutorial.

Even if you have never used any CSS before, you will still learn everything you need from this book, because the first section provides a thorough grounding in what CSS is, what it can do for you, and how to use it. And if you have used CSS before, it will act as a great revision source to ensure you are fully up-to-date before moving on to learning all the latest additions to CSS3.

Between the lessons, the self-test questions, and the videos, this course will ensure that you become expert at CSS very quickly.

To save you typing them in, all the example files from the book are saved in a freely downloadable zip file available at the companion website: *20lessons.com*.

## What This Book Covers?

This book covers every aspect of CSS, starting with basic syntax and language rules, such as where and how you include CSS in your web documents. Then the differences between styling elements by type, class, and ID are explained, along with how to refine the elements to which CSS will be supplied using selectors.

The term cascade (in cascading style sheets) is fully explained, as is how to create professional results with fonts and typography, and how to lay out compelling pages. All the CSS and CSS3 selectors are detailed, along with the new CSS3 ways to manage colors, backgrounds, borders, opacity, and more. How to transform and animate elements in 2D and 3D is also explained, with simple examples that you can easily incorporate in your own websites.

By the time you finish the book's 20 lessons, you'll have a thorough grounding in CSS, and be able to use it to ensure your web pages look as good as possible.

## How to Use This Book?

This book has been written in a logical order so that each lesson builds on information learned in the previous ones. You should begin at Lesson 1 and then work sequentially through the book, proceeding to the next lesson only when you can correctly answer the self-test questions in the previous one.

If you already use CSS, you may wish to just browse through the first few lessons before tucking into the CSS3 section.

Lessons should take you less than an hour to finish, including viewing the accompanying video walk-through provided with each one. With over 5 hours of video in total, that's an average of 15 minutes dedicated to each lesson.

# How Is This Book Organized?

The first section of this book covers everything you need to know as a newcomer to CSS, including these lessons: Introduction to CSS, Learning the CSS Rules, Applying Declarations to IDs and Classes, Accessing Selectors, Working with the Cascade, Selecting Fonts and Typography, Manipulating Color and Position, Handling Pseudo-Selectors and Using Shorthand Properties, and Understanding the Box Model.

The second section moves on to explaining everything that's new in CSS3, including these lessons: Introduction to CSS3, Using Selectors and Attribute Selectors, Setting Backgrounds, Attaching Borders, Controlling Box Shadows, Overflow, and Columns, Adding Colors and Opacity, Creating Text Effects and Changing the Box Model, Linking to Web Fonts, Making 2D Transformations, Applying Specific Transformations, and Directing 3D Transformations. The appendix lists all the answers to the self-test questions in each chapter.

*This page intentionally left blank*

# PART I

## Basic CSS





*This page intentionally left blank*

# Introduction to CSS



To view the accompanying video for this lesson, please visit [mhprofessional.com/nixoncss/](http://mhprofessional.com/nixoncss/).

When CSS was invented it was based around a Document Object Model (DOM), a means of separating out all the different elements within a web page into discrete objects, each with its own properties and values. This led logically to the introduction of style sheets, enabling the content of a web page to be completely separated from its styling; it also made HTML documents easily modifiable by languages such as JavaScript to provide dynamic user interaction.

Because web pages use a DOM, it is easy for you to style every aspect of it using CSS. For example, each heading will be within pairs of tags such as `<h1>` and `</h1>`, and a single CSS declaration can set the styling of all such occurrences within a document, changing the font used, its size, any font decoration, and so on.

This lets you completely change the design of a page without altering the HTML. Some style settings can even apply dynamic effects to page elements, such as changing their color and other properties when the mouse passes over them, or create transition effects by using proprietary browser extensions.



## Note

The example files from this book are in a file you can download at [20lessons.com](http://20lessons.com). The files for this lesson are saved in it as *embeddedstyles.htm*, *example.htm*, *importedstyles.htm*, *importedstyles2.htm*, *styles.css*, *styletag.htm*, *usingclasses.htm*, and *usingids.htm*.

## How the Document Object Model Works

The DOM separates different parts of an HTML document into a hierarchy of objects, each one having its own properties. The term *property* is used for referring to an attribute of an object, such as the HTML it contains, its width and height, and so on.

The outermost object possible is the window object, which is the current browser window, tab, iframe, or popped-up window. Underneath this is the document object, of which there can be more than one (such as several documents loaded into different iframes within a page). And inside a document there are other objects such as the head and body of a page.

Within the head there can be other objects such as the title and meta objects, while the body object can contain numerous other objects, including headings, anchors, forms, and so forth. For example, Figure 1-1 shows a representation of the DOM of an example document, with the document shown inside the outer window, and having the title Hello, a <meta> tag in the head, and three HTML elements (a link, a form, and an image) in the body section.

Of course, even the simplest of web pages has more structure than is shown here, but it serves to illustrate how the DOM works; starting from the very outside is the window, inside which there's a single document, and within the document are the various elements or objects, which connect to each other.

In the figure, values are shown with a darker background and in italics. For example, the value robots is the value of the property name, which is a property of <meta>, and so on. Although it isn't shown in the figure, the <meta> tag should have another matching property called content, which would contain a string specifying which robots (web crawlers) may access the web page.

Other properties are href, which has a value of *http://google.com* (and is itself a property of <meta>, and so on), and <title>, which has the value Hello. All the other items are objects or object argument names. If the figure were to extend farther down and sideways, other objects and properties attached to the ones shown would come into view. A couple of the places where these would appear are shown by unconnected dotted lines.

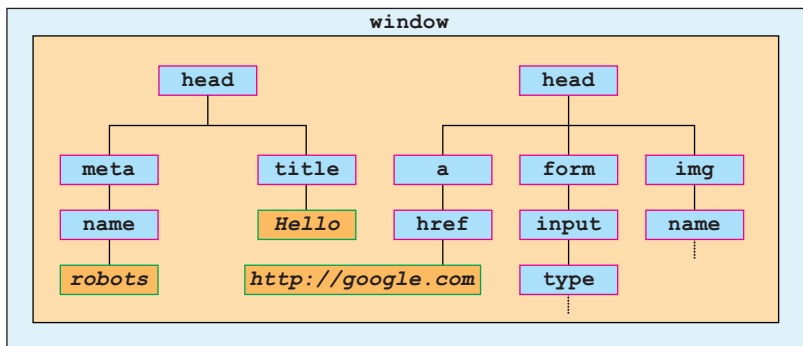


FIGURE 1-1 Example of a DOM showing head and body sections



In HTML tags you can generally use the single or double quotation marks interchangeably. Therefore `<a href="http://google.com">` is equivalent to `<a href='http://google.com'>`. Wherever possible, though, I tend to use single quotes because they don't require pressing the Shift key to type them in. Also there are sometimes occasions when you need two levels of nested quotes, where I would then choose double quotation marks for the outer string, and then apply single quotes within it, like this: `<p style="font-family: 'Times New Roman';">`.

Representing this as HTML, the structure of the head section looks like this:

```
<head>
  <meta name='robots' content='index, follow'>
  <title>Hello</title>
</head>
```

And the body section of HTML might look like this:

```
<body>
  <img src='/images/welcome.jpg'>
  <a href='http://google.com'>Visit Google</a>
  or enter your username and password to continue...
  <form id='login' method='post' action='login.php'>
    <input type='text' name='name'>
    <input type='password' name='password'>
    <input type='submit'>
  </form>
</body>
```

Remembering that these two sections of HTML are part of the same document, we would bring them both together inside an `<html>` tag (preceded by a `!DOCTYPE` declaration), like this:

```
<!DOCTYPE html>
<html>
  <head>
    <meta name='robots' content='index, follow'>
    <title>Hello</title>
  </head>
  <body>
    <img src='/images/welcome.jpg'>
    <a href=http://google.com>Visit Google</a>,
    or enter your username and password to continue...
    <form id='login' method='post' action='login.php'>
      <input type='text' name='name'>
      <input type='password' name='password'>
      <input type='submit'>
    </form>
  </body>
</html>
```

Of course, all web pages are different, but they will usually follow this same form.

## Correct HTML Structure and Nesting

To follow recommended HTML structure and to ensure your documents are readable by the maximum number of browsers and other clients, attribute values within tags should be contained in either single or double quotation marks like this:

`<a href='http://yahoo.com' >`, even though nearly all browsers allow you to omit them, like this: `<a href=http://yahoo.com>`.

You should also close (end) every tag, and do so in the correct order. For example, you shouldn't close a document by issuing `</html >` followed by `</body >` because the proper nesting of tags would be broken by this reversal. The correct way to close a document is with `</body >`, followed by `</html >`.

## About Cascading Style Sheets

Using CSS you can apply styles to your web pages to make them look exactly how you want. This works because CSS is connected to the DOM so that you can quickly and easily restyle any element. For example, if you don't like the default look of the `<h1 >`, `<h2 >`, and other heading tags, you can assign new styles to override the default settings for the font family and size used, or whether bold or italics should be set, and many more properties too.

One way you can add styling to a web page is by inserting the required CSS into the head of a web page between the `<head >` and `</head >` tags. So, to change the style of the `<h1 >` tag you might use the following CSS:

```
<style>
  h1 {
    color      :olive;
    font-size  :18pt;
    font-family:'Times New Roman';
  }
</style>
```



FIGURE 1-2 A simple document with a CSS-styled heading

Within an HTML page, this might look like the following (see Figure 1-2):

```
<!DOCTYPE html>
<html>
  <head>
    <meta name='robots' content='index, follow'>
    <title>The style tag</title>
    <style>
      h1 {
        color      :olive;
        font-size  :36pt;
        font-family:'Times New Roman';
        font-style :italic;
      }
    </style>
  </head>
  <body>
    <h1>I am a level 1 heading</h1>
  </body>
</html>
```

## Importing a Style Sheet

When you wish to style a whole site, rather than a single page, a better way to manage style sheets is to completely remove them from your web pages to separate files, and then import the ones you need. This lets you use different style sheets for different layouts (such as web and print) without changing the HTML.

There are a couple of different ways this can be achieved, the first of which is by using the CSS `@import` directive like this:

```
<style>
  @import url('styles.css');
</style>
```

This statement tells the browser to fetch a style sheet with the name *styles.css*. The `@import` command is quite flexible in that you can create style sheets that themselves pull in other style sheets, and so on. Just make sure that there are no `<style>` or `</style>` tags in any of your external style sheets or they will not work. External style sheets must contain only CSS and never any HTML tags.

## Importing CSS from Within HTML

You can also include a style sheet with the HTML `<link>` tag like this:

```
<link rel='stylesheet' type='text/css' href='styles.css'>
```

This has the exact same effect as the `@import` directive, except that `<link>` is an HTML-only tag and is not a valid style directive, so it cannot be used from within one style sheet to pull in another. Also, it cannot be placed within a pair of `<style>` and `</style>` tags.

Just as you can use multiple `@import` directives within your CSS to include multiple external style sheets, you can also use as many `<link>` tags as you like in your HTML.

## Embedded Style Settings

There's also nothing stopping you from individually setting or overriding certain styles for the current page on a case by case basis by inserting `style` attributes directly within HTML, like this (which results in italic blue text within the tags):

```
<h1 style='font-style:italic; color:blue;'>Hello there</h1>
```

But this should be reserved only for the most exceptional circumstances as it breaks the separation of content and presentation.

## Using IDs

A better solution for setting the style of an element is to assign an ID to it in the HTML, like this:

```
<h1 id='highlight'>Hello there</h1>
```

What this does is state that the contents of the `<h1>` element with the ID `highlight` should have the style applied to it that is defined in the following rule:

```
#highlight {
  text-decoration:underline;
  color           :orange;
}
```



Note the use of the `#` symbol, which specifies that only the element that has an ID of `highlight` should be styled with this statement.

## Using Classes

If you would like to apply the same style to many elements, you don't have to give each one a different ID because you can specify a class to manage them all, like this:

```
<h1 class='highlight'>Hello</h1>
```

What this does is state that the contents of this element (and any other element that uses the same class) should have the style defined for the `highlight` class applied to it. Once a class is assigned you can use the following rule, either in the document's head section or within an external style sheet for setting the styles for the class:

```
.highlight {  
  text-decoration:underline;  
  color           :orange;  
}
```

Instead of using a `#` symbol, which is reserved for IDs, class selectors are prefaced with a `.` (period) symbol.



You may wonder whether the semicolons used in CSS are actually necessary or if are they optional, as with JavaScript. The answer is yes and no. In order to separate multiple CSS statements on the same line, you must place semicolons between them. But if there is only one statement in a set of rules (or in an inline style attribute within an HTML tag), then you can omit the semicolon, as you can for the final statement in a group. However, to avoid hard-to-find CSS errors, you may prefer to always use a semicolon after every CSS declaration so that you can copy and paste them and otherwise modify properties without worrying about removing semicolons where they aren't strictly necessary or having to add them where they are required.

## Summary

Now that you have learned some of the basics of CSS, in Lesson 2, I'll show you how CSS rules work, and how you can apply them in different ways.

## Self-Test Questions

Using these questions, test how much you have learned in this lesson. If you don't know an answer, go back and reread the relevant section until your knowledge is complete. You can find the answers in the appendix.

1. What model is HTML based on?
2. What is the main purpose of style sheets?
3. What is the outermost object in the DOM?
4. What are the two main sections of an HTML document?
5. What types of elements are typically used in the head of an HTML document?