

**The Most
Authoritative
Quick Reference
for Programmers**

OSBORNE 

Programmer's Reference

Cascading Style Sheets 2.0

**Essential Programming
Information at Your Fingertips**

**Completely Covers New Properties
and Features of CSS 2.0**

**Includes Sample CSS—
Plus Discussions on When and
How to Use It—and a Browser
Compatibility Chart**

Eric A. Meyer

IT manager and CSS expert

Cascading Style Sheets 2.0

PROGRAMMER'S REFERENCE

Eric A. Meyer

Osborne/**McGraw-Hill**

New York Chicago San Francisco
Lisbon London Madrid Mexico City Milan
New Delhi San Juan Seoul Singapore Sydney Toronto

Osborne/**McGraw-Hill**
2600 Tenth Street
Berkeley, California 94710
U.S.A.

To arrange bulk purchase discounts for sales promotions, premiums, or fund-raisers, please contact Osborne/**McGraw-Hill** at the above address. For information on translations or book distributors outside the U.S.A., please see the International Contact Information page immediately following the index of this book.

Cascading Style Sheets 2.0 Programmer's Reference

Copyright © 2001 by The McGraw-Hill Companies. All rights reserved. Printed in the United States of America. Except as permitted under the 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.

90 DOC DOC 019876

ISBN 0-07-213178-0

Publisher Brandon A. Nordin
Vice President & Associate Publisher Scott Rogers
Acquisitions Editor Jim Schachterle
Project Editor Madhu Prasher
Acquisitions Coordinator Tim Madrid
Copy Editor Mike McGee
Proofreader Paul Tyler
Indexer Claire Splan
Computer Designers Tara Davis and Lucie Ericksen
Illustrator Michael Mueller
Series Design Peter F. Hancik

This book was composed with Corel VENTURA™ Publisher.

Information has been obtained by Osborne/**McGraw-Hill** from sources believed to be reliable. However, because of the possibility of human or mechanical error by our sources, Osborne/**McGraw-Hill**, or others, Osborne/**McGraw-Hill** 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 use of such information.

INTRODUCTION

In the beginning, there was HTML. And it was pretty good, but not great. You couldn't really create nifty visual designs with it, which gave rise to table-based layout and single-pixel GIF tricks. And that was pretty bad. So CSS was born, and it was very good—in theory, anyway. There was a long struggle to make CSS a viable technology, thanks to imperfect interpretations of the specification, but lo! The day arrived when CSS could be used without fear and dread. And the people rejoiced.

Thanks to CSS, designers can cut back on the FONT and table tricks they've been forced to cobble together, and dramatically clean up their markup. With the coming of XHTML and XML, both of which are deeply semantic and must rely on some styling mechanism to become visually appealing, CSS is growing more and more popular. It's a flexible, easy-to-understand language which offers designers a lot of power. Because it reduces markup clutter, it makes pages easier to maintain. And its centralized styling abilities lets designers adjust page layout with quick, easy edits of the styles, not dramatic changes to the markup. In fact, CSS makes it possible to completely reshape the look of a document without changing a single character inside the BODY element.

This book endeavors to efficiently describe the properties and values of CSS2, which was the latest CSS standard when the book was written, and to provide details on property interactions, common authoring mistakes, and other information which designers should find useful.

The text has been arranged to present basic concepts first, with details on important CSS algorithms and behaviors (Chapter 1). This is followed with “core” information which describes the types of values that can be used in CSS2 (Chapter 2), and the various ways in which elements can be selected for styling (Chapter 3). This first part of the book does its best to describe the foundation of CSS, for the rest of it would not function without the values and concepts presented.

The middle of the book (Chapters 4 through 6) is the largest portion, and is probably the area where readers will spend the most time—all of the properties found in CSS2 are defined, described, and annotated with notes. These properties are broken up into separate chapters, with Chapter 4 devoted to visual-media

properties, Chapter 5 to paged-media properties, and Chapter 6 to aural-media properties. Each property is described in terms of its allowed values, its initial (or default) value, and other common aspects. There are also detailed descriptions of the meaning of each allowed value, notes about how the property works, examples of the property in use, and a list of related properties.

The final part of the book (Chapters 7 through 9) contains other useful information about CSS, including a browser support chart, a CSS2 property quick reference, and a list of useful online resources. Between the contents of this book and the resources provided, it should be possible to decipher any CSS conundrums you may encounter. Although CSS can sometimes seem a bit mystifying, it is more than worth the effort of learning its secrets. Enjoy!

CONTENTS @ A GLANCE

I	Reference	
1	Basic CSS Concepts	3
2	Values	31
3	Selectors, Pseudo-Classes, Pseudo-Elements, and At-Rules	43
4	Visual Media Styles	65
5	Paged Media Styles	213
6	Aural Media Styles	231
II	Summaries	
7	Browser Compatibility	261
8	CSS2 Quick Reference	291
9	Useful Resources	313
	Index	317

CONTENTS

Introduction	xv
--------------------	----

I Reference

1	Basic CSS Concepts	3
	Associating Styles with Documents	3
	LINK Element	3
	STYLE Element	4
	STYLE Attribute	5
	@import Rule	6
	Rule Structure	6
	Resolving Style Conflicts	7
	Cascade Rules	8
	Specificity Calculation	9
	Importance	10
	Inheritance	10
	Shorthand Properties	11
	Visual Layout	13
	The Box Model	13
	The Inline Layout Model	15
	Float Rules	18
	Positioning Rules	20
	Relative Positioning	20
	Absolute Positioning	21
	Font Rules	26
	Font Family Matching	26
	Font Face Matching	28
	Font Weight Matching	28
2	Values	31
	Value Representations	31
	Basic Rules	32
	Value Reference	32
	Color Values	32
	Length Values	34
	Angle Values	37
	Time Values	38

Frequency Values	39
Strings	39
Percentages	40
URI Values	41

3 Selectors, Pseudo-Classes, Pseudo-Elements, and At-Rules 43

Selectors	43
Type Selector	43
Descendant Selector	44
Universal Selector	44
Child Selector	45
Adjacent-Sibling Selector	45
Attribute Selectors	46
Class Selectors	48
ID selectors	49
Pseudo-Classes	49
:first-child	50
:link	51
:visited	51
:hover	52
:active	53
:focus	53
:lang(n)	54
:left, :right, and :first	55
Pseudo-Elements	55
:first-letter	55
:first-line	56
:before	57
:after	58
At-Rules	59
@import	59
@media	61
@charset	61
@font-face	62
@page	63

4 Visual Media Styles 65

Reference	65
background	65
background-attachment	67
background-color	69
background-image	70

background-position	72
background-repeat	75
border	76
border-bottom	78
border-bottom-color	79
border-bottom-style	80
border-bottom-width	81
border-collapse	82
border-color	84
border-left	85
border-left-color	86
border-left-style	87
border-left-width	88
border-right	89
border-right-color	90
border-right-style	91
border-right-width	92
border-spacing	93
border-style	94
border-top	97
border-top-color	98
border-top-style	99
border-top-width	100
border-width	101
bottom	103
caption-side	105
clear	106
clip	108
color	109
content	111
counter-increment	113
counter-reset	115
cursor	116
direction	119
display	120
empty-cells	124
float	125
font	127
font-family	130
font-size	132
font-size-adjust	135
font-stretch	137
font-style	139
font-variant	141
font-weight	142

height	145
left	146
letter-spacing	148
line-height	149
list-style	151
list-style-image	153
list-style-position	154
list-style-type	155
margin	158
margin-bottom	160
margin-left	161
margin-right	162
margin-top	164
marker-offset	165
max-height	166
max-width	168
min-height	169
min-width	170
outline	171
outline-color	172
outline-style	173
outline-width	175
overflow	176
padding	178
padding-bottom	179
padding-left	180
padding-right	182
padding-top	183
position	184
quotes	186
right	187
table-layout	189
text-align	191
text-decoration	193
text-indent	195
text-shadow	196
text-transform	198
top	199
unicode-bidi	201
vertical-align	202
visibility	205
white-space	206
width	207
word-spacing	208
z-index	210

5 Paged Media Styles 213

General Paged-Media Rules	214
Page-Breaking Rules	214
Content-Clipping Rules	216
Reference	216
@page	216
:first	218
:left	219
:right	219
marks	220
orphans	221
page	222
page-break-after	224
page-break-before	225
page-break-inside	226
size	227
widows	229

6 Aural Media Styles 231

Reference	231
azimuth	231
cue	234
cue-after	235
cue-before	236
elevation	237
pause	238
pause-after	239
pause-before	240
pitch	241
pitch-range	243
play-during	244
richness	246
speak	247
speak-header	248
speak-numeral	249
speak-punctuation	250
speech-rate	251
stress	253
voice-family	254
volume	255

II Summaries

7 Browser Compatibility	261
Notes	279
1.1 Containment in HTML	279
1.1 Containment in HTML	279
1.3 Inheritance	279
1.4 Class Selector	279
1.5 ID Selector	279
1.6 Contextual Selectors	280
3.2 Cascading Order	280
5.2.2 font-family	280
5.2.4 font-variant	280
5.2.6 font-size	280
5.3.2 background-color	281
5.3.2 background-color	281
5.3.4 background-repeat	281
5.3.4 background-repeat	281
5.3.4 background-repeat	282
5.3.7 background	282
5.4.3 text-decoration	282
5.4.3 text-decoration	283
5.4.5 text-transform	283
5.4.6 text-align	283
5.4.8 line-height	283
5.4.8 line-height	283
5.5.01 margin-top	283
5.5.02 margin-right	284
5.5.03 margin-bottom	284
5.5.04 margin-left	284
5.5.05 margin	284
5.5.06 padding-top	284
5.5.07 padding-right	285
5.5.08 padding-bottom	285
5.5.09 padding-left	285
5.5.10 padding	285
5.5.11 border-top-width	285
5.5.12 border-right-width	285
5.5.13 border-bottom-width	285
5.5.14 border-left-width	286
5.5.15 border-width	286
5.5.16 border-color	286

5.5.17 border-style	286
5.5.18 border-top	286
5.5.19 border-right	286
5.5.20 border-bottom	287
5.5.21 border-left	287
5.5.22 border	287
5.5.23 width	287
5.5.25 float	287
5.5.26 clear	288
5.6.1 display	288
5.6.3 list-style-type	288
5.6.5 list-style-position	288
6.1 Length Units	289
6.3 Color Units	289
6.4 URLs	289
8 CSS2 Quick Reference	291
9 Useful Resources	313
Tools	313
CSScheck (Web Design Group)	313
CSS Validator (World Wide Web Consortium)	313
Discussion Groups	314
Stylesheets Newsgroup (Usenet)	314
Style Discussion List (W3C)	314
Style Discussion List (HTML Writers Guild)	314
References	315
CSS Activity Page (W3C)	315
Style Sheets Reference Guide (Web Review)	315
The House of Style (Western Civilisation)	315
Agitprop	316
CSS Pointers Group	316
Index	317

Part I

Reference

Chapter 1

Basic CSS Concepts

In order to comprehend how CSS affects the presentation of a document, there are some key concepts that must be grasped. Once these are understood, even in part, it becomes easier to see how the properties and values of CSS work. Do not, however, feel that you must completely understand everything in this chapter before experimenting with CSS. In fact, it is better to review this chapter first, then refer back to it as properties are used.

Associating Styles with Documents

There are four ways to associate styles with a document. These range from associating a separate stylesheet with your document to embedding style information in the document itself.

LINK Element

The `LINK` element is found in HTML and XHTML, and is used to associate an external stylesheet with a document.

Generic Syntax

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

Attributes

rel="..."

This attribute describes the relation of the `LINKed` file to the document itself. For external stylesheets, there are two possible values: `stylesheet` and `alternate stylesheet`. Any `LINK` with a `rel` of `stylesheet` will be used in the styling of the document. The value `alternate stylesheet` is used to refer to stylesheets that are not used in the default rendering of the document, but which can, in theory, be selected by the user and thus change the presentation. The user agent must provide a mechanism to do so in

4 Associating Styles with Documents

order for this to work, and unfortunately most user agents do not provide such a mechanism. This attribute is *required*.

href="..."

The value of this attribute is the URL of the external stylesheet. Either relative or absolute URLs may be used. This attribute is *required*.

type="text/css"

This is used to declare the type of data which is being LINKed to the document. When associating a CSS stylesheet, the only allowed value is `text/css`. Other stylesheet languages will call for different values (e.g., `text/xml`). This attribute is *required*.

media="..."

Using this attribute, one can declare a stylesheet to apply only to certain media. The default value is `all`, which means that the styles will be used in all media in which the document is presented. Recognized values under CSS are `all`, `screen`, `print`, `projection`, `aural`, `braille`, `embossed`, `handheld`, `tty`, and `tv`. Any number of these values can be used in a `media` attribute by formatting them as a comma-separated list. This attribute is optional.

Notes

In this approach, the stylesheet is placed in its own file. Such files are usually given an extension of `.css`, such as `main-styles.css`. The `LINK` element must be placed inside the `HEAD` element in HTML and XHTML, but XML-based markup languages may have other requirements.

Examples

```
<link rel="stylesheet" type="text/css"
      href="http://www.my.site/styles/basic.css">
<link rel="stylesheet" type="text/css" href="article.css"
      media="screen,projection">
<link rel="stylesheet" type="text/css" href="printout.css"
      media="print">
```

STYLE Element

The `STYLE` element is found in HTML and XHTML, and is used as a container for an embedded stylesheet.