

WORLDWIDE
BESTSELLER
NOW IN FULL
COLOR!

VISUAL QUICKSTART GUIDE



HTML, XHTML & CSS

SIXTH EDITION

*Learn HTML, XHTML,
and CSS the Quick and
Easy Way!*

ELIZABETH CASTRO

HTML, XHTML, and CSS, Sixth Edition: Visual QuickStart Guide

by Elizabeth Castro

Peachpit Press

1249 Eighth Street
Berkeley, CA 94710
(510) 524-2178
(510) 524-2221 (fax)

Find us on the Web at: www.peachpit.com

Or check out Liz's Web site at www.cookwood.com

To report errors, send a note to errata@peachpit.com

Peachpit Press is a division of Pearson Education

Copyright © 2007 by Elizabeth Castro

All images are copyright © 1997–2007 by Elizabeth Castro

Cover design: Peachpit Press

Notice of rights

All rights reserved. No part of this book may be reproduced or transmitted in any form by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher. For information on getting permission for reprints and excerpts, contact permissions@peachpit.com.

Notice of liability

The information in this book is distributed on an “As is” basis, without warranty. While every precaution has been taken in the preparation of the book, neither the author nor Peachpit Press, shall have any liability to any person or entity with respect to any loss or damage caused or alleged to be caused directly or indirectly by the instructions contained in this book or by the computer software and hardware products described in it.

Trademarks

Visual QuickStart Guide is a registered trademark of Peachpit Press, a division of Pearson Education. Openwave, the Openwave logo, Openwave SDK, Openwave SDK Universal Edition, Openwave SDK, WAP Edition are trademarks of Openwave Systems Inc. All other trademarks are the property of their respective owners. Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and Peachpit Press was aware of a trademark claim, the designations appear as requested by the owner of the trademark. All other product names and services identified throughout this book are used in editorial fashion only and for the benefit of such companies with no intention of infringement of the trademark. No such use, or the use of any trade name, is intended to convey endorsement or other affiliation with this book.

ISBN: 0-321-43084-0

0 9 8 7 6 5 4 3 2 1

Printed in the United States of America

For my parents
(all four of them!)
who didn't always agree,
but who supported me anyway.

Special thanks to:

Nancy Davis, at Peachpit Press, who has given me the most perfect combination of encouragement, editing, and friendship.

Andrei Pasternak and **Mimi Heft**, and all the other folks at Peachpit Press, for all their help getting this book out.

Kate Reber and **Nolan Hester**, both formerly of Peachpit Press, for their help with earlier editions of this book.

The Web is an incredible resource. Not only do people share their knowledge freely on their Web sites, but they are also incredibly generous with their time, helpfully answering my questions, no matter how arcane. In particular, I'd like to thank **Larry Ullman**, whose *PHP for the World Wide Web: Visual QuickStart Guide, Second edition* is an excellent way to begin with PHP, and who kindly helped me polish up my first scripts (<http://www.dmcinsights.com/pbpmysql/>), **Richard Isbida**, whose tutorials at the W3C (<http://www.w3.org/International/>) on creating Multilingual Web sites are invaluable; **Patrick Woolsey**, at Barebones Software (<http://www.barebones.com>), for very helpful and timely answers to all my encoding and BBEdit questions, **Alan Wood**, whose Unicode Resources Web site (<http://www.alanwood.net/unicode/>) was a major source of information for the Symbols and Non-English Characters chapter, **Jeffrey Zeldman**, whose online magazine, *A List Apart* (<http://www.alistapart.com>), is an essential resource for Web designers, and **Paul Boutin**, **Douglas Bowman**, **Dan Cederholm**, **Patrick Griffiths**, **Ian Hickson**, **Molly Holzschlag**, **Steve Krug**, **Drew McLellan**, **Eric Meyer**, **John Oxtton**, **Dave Shea**, **Geoff Stearns**, and **Danny Sullivan**, whose books, blogs, Web sites, and generosity I have found most inspiring.

Andreu, for his feedback, for his great Photoshop tips, and for sharing his life with me; and **my sweet kids** to whom I can finally reply, "Yes, I can play with you now."

Llumi and **Xixo**, for chasing cherry tomatoes and each other around my office and for helping me think up examples of (X)HTML documents; and the new generation of **Sky**, **Night**, and **Sir Edmund**, who we still hope might come back.

And all the readers of earlier versions of this book, who took the time to write me with accolades, questions, and suggestions.

Table of Contents

Introduction	13
<i>The Internet, the Web, and HTML</i>	14
<i>Open but Not Equal</i>	15
<i>The Browser Wars</i>	16
<i>The Push for Standards</i>	17
<i>CSS and Browser Support</i>	20
<i>XHTML vs. HTML: What Should You Use?</i>	21
<i>How This Book Works</i>	23
<i>What's Changed in the Sixth Edition</i>	24
<i>The HTML VQS Web Site</i>	26
Chapter 1: Web Page Building Blocks	27
<i>Markup: Elements, Attributes, and Values</i>	28
<i>A Web Page's Text Content</i>	32
<i>Links, Images, and Other Non-Text Content</i>	33
<i>File Names</i>	34
<i>URLs</i>	35
<i>HTML vs. XHTML</i>	38
<i>Versions, Flavors, and DOCTYPE</i>	40
<i>The Default Display of (X)HTML</i>	42
Chapter 2: Working with Web Page Files	43
<i>Designing Your Site</i>	44
<i>Creating a New Web Page</i>	45
<i>Saving Your Web Page</i>	46
<i>About Microsoft Word and Web Pages</i>	48
<i>Specifying a Default or "Home" Page</i>	49
<i>Editing Web Pages</i>	50
<i>Organizing Files</i>	51
<i>Viewing Your Page in a Browser</i>	52
<i>The Inspiration of Others</i>	53

Chapter 3: Basic (X)HTML Structure	55
<i>Starting Your Web Page</i>	56
<i>Creating the Foundation</i>	58
<i>Declaring the Encoding</i>	59
<i>Creating a Title</i>	60
<i>Creating Section Headers</i>	61
<i>Starting a New Paragraph</i>	62
<i>Naming Elements</i>	63
<i>Breaking up a Page into Divisions</i>	64
<i>Creating Inline Spans</i>	65
<i>Creating a Line Break</i>	66
<i>Adding Comments</i>	67
<i>Labeling Elements in a Web Page</i>	68
Chapter 4: Basic (X)HTML Formatting	69
<i>Making Text Bold or Italic</i>	70
<i>Changing the Size of Text</i>	71
<i>Using a Monospaced Font</i>	72
<i>Using Preformatted Text</i>	73
<i>Quoting Text</i>	74
<i>Creating Superscripts and Subscripts</i>	76
<i>Marking Changed Text</i>	77
<i>Explaining Abbreviations</i>	78
<i>Centering Elements on a Page</i>	79
Chapter 5: Images	81
<i>About Images for the Web</i>	82
<i>Getting Images</i>	86
<i>Choosing an Image Editor</i>	87
<i>The Save for Web Command</i>	88
<i>Inserting Images on a Page</i>	90
<i>Offering Alternate Text</i>	91
<i>Specifying Size for Speedier Viewing</i>	92
<i>Scaling an Image</i>	94
<i>Making Images Smaller</i>	95
<i>Making Images Float</i>	96
<i>Stopping Elements from Wrapping</i>	98
<i>Adding Space around an Image</i>	99
<i>Aligning Images</i>	100
<i>Adding Horizontal Rules</i>	101
<i>Adding an Icon for Your Web Site</i>	102
Chapter 6: Links	103
<i>Creating a Link to Another Web Page</i>	104
<i>Creating Anchors</i>	106
<i>Linking to a Specific Anchor</i>	107
<i>Targeting Links to Specific Windows</i>	108
<i>Setting the Default Target</i>	109
<i>Creating Other Kinds of Links</i>	110
<i>Creating Keyboard Shortcuts for Links</i>	112
<i>Setting the Tab Order for Links</i>	113
<i>Using Images to Label Links</i>	114
<i>Linking Thumbnails to Images</i>	115
<i>Dividing an Image into Clickable Regions</i>	116
<i>Creating a Client-Side Image Map</i>	117

Chapter 7: Style Sheet Building Blocks 119

<i>Constructing a Style Rule</i>	<i>120</i>
<i>Adding Comments to Style Rules</i>	<i>121</i>
<i>The Cascade: When Rules Collide</i>	<i>122</i>
<i>A Property's Value</i>	<i>124</i>

Chapter 8: Working with Style Sheet Files 127

<i>Creating an External Style Sheet</i>	<i>128</i>
<i>Linking External Style Sheets</i>	<i>129</i>
<i>Offering Alternate Style Sheets</i>	<i>130</i>
<i>Creating an Internal Style Sheet</i>	<i>131</i>
<i>Importing External Style Sheets</i>	<i>132</i>
<i>Using Media-Specific Style Sheets</i>	<i>133</i>
<i>Applying Styles Locally</i>	<i>134</i>
<i>The Importance of Location</i>	<i>135</i>
<i>The Inspiration of Others: CSS</i>	<i>136</i>

Chapter 9: Defining Selectors. 137

<i>Constructing Selectors</i>	<i>138</i>
<i>Selecting Elements by Name</i>	<i>139</i>
<i>Selecting Elements by ID or Class</i>	<i>140</i>
<i>Selecting Elements by Context</i>	<i>141</i>
<i>Selecting Part of an Element</i>	<i>144</i>
<i>Selecting Link Elements Based on Their State</i>	<i>146</i>
<i>Selecting Elements Based on Attributes</i>	<i>147</i>
<i>Specifying Groups of Elements</i>	<i>148</i>
<i>Combining Selectors</i>	<i>149</i>

Chapter 10: Formatting with Styles 151

<i>Choosing a Font Family</i>	<i>152</i>
<i>Specifying Alternate Fonts</i>	<i>153</i>
<i>Creating Italics</i>	<i>154</i>
<i>Applying Bold Formatting</i>	<i>155</i>
<i>Setting the Font Size</i>	<i>156</i>
<i>Setting the Line Height</i>	<i>158</i>
<i>Setting All Font Values at Once</i>	<i>159</i>
<i>Setting the Color</i>	<i>160</i>
<i>Changing the Text's Background</i>	<i>161</i>
<i>Controlling Spacing</i>	<i>162</i>
<i>Adding Indents</i>	<i>163</i>
<i>Setting White Space Properties</i>	<i>164</i>
<i>Aligning Text</i>	<i>165</i>
<i>Changing the Text Case</i>	<i>166</i>
<i>Using Small Caps</i>	<i>167</i>
<i>Decorating Text</i>	<i>168</i>

Chapter 11: Layout with Styles	169
<i>Structuring Your Pages</i>	170
<i>The Box Model</i>	171
<i>Changing the Background</i>	172
<i>Setting the Height or Width for an Element</i>	174
<i>Setting the Margins around an Element</i>	176
<i>Adding Padding around an Element</i>	177
<i>Offsetting Elements In the Natural Flow</i>	178
<i>Positioning Elements Absolutely</i>	179
<i>Affixing an Element to the Browser Window</i>	180
<i>Making Elements Float</i>	181
<i>Controlling Where Elements Float</i>	182
<i>Positioning Elements in 3D</i>	183
<i>Setting the Border</i>	184
<i>Changing the Cursor</i>	186
<i>Determining Where Overflow Should Go</i>	187
<i>Aligning Elements Vertically</i>	188
Chapter 12: Dynamic Effects with Styles	189
<i>Displaying and Hiding Elements</i>	190
<i>Creating Rollover Buttons</i>	192
<i>Creating Pop-ups</i>	193
<i>Creating Drop-Down Menus with Lists</i>	194
<i>Replacing Headers with Images</i>	196
Chapter 13: Style Sheets for Handhelds	199
<i>Mobilize vs. Miniaturize</i>	200
<i>Looking at Your Site</i>	201
<i>XHTML and CSS for Handhelds</i>	202
<i>Creating Style Sheets for Handhelds</i>	204
<i>Hiding Extraneous Elements</i>	205
<i>Creating and Using an Image Header</i>	206
<i>Creating Extra Links to the Top</i>	207
<i>Adjusting for the Small Screen</i>	208
Chapter 14: Style Sheets for Printing	209
<i>Creating a Style Sheet for Print</i>	210
<i>How Print Style Sheets Differ</i>	211
<i>Controlling Page Breaks</i>	212
<i>Printing Link URLs</i>	213
<i>Controlling Widows and Orphans</i>	214
Chapter 15: Lists	215
<i>Creating Ordered and Unordered Lists</i>	216
<i>Choosing Your Markers (Bullets)</i>	218
<i>Choosing Where to Start List Numbering</i>	219
<i>Using Custom Markers</i>	220
<i>Controlling Where Markers Hang</i>	221
<i>Setting All List-Style Properties at Once</i>	222
<i>Creating Definition Lists</i>	223
<i>Styling Nested Lists</i>	224

Chapter 16: Tables 227

<i>Mapping Out Your Page</i>	228
<i>Creating a Simple Table</i>	229
<i>Adding a Border</i>	230
<i>Setting the Width</i>	232
<i>Centering a Table</i>	234
<i>Wrapping Text around a Table</i>	235
<i>Combining Tables</i>	236
<i>Aligning a Cell's Contents</i>	238
<i>Changing the Background</i>	240
<i>Controlling the Space</i>	242
<i>Spanning a Cell across Columns</i>	244
<i>Spanning a Cell across Rows</i>	245
<i>Dividing Your Table into Column Groups</i>	246
<i>Dividing the Table into Horizontal Sections</i>	248
<i>Choosing Which Borders to Display</i>	249
<i>Controlling Line Breaks in a Cell</i>	251
<i>Speeding up Table Display</i>	252

Chapter 17: Forms 253

<i>Creating a Form</i>	254
<i>Processing Forms</i>	256
<i>Sending Form Data via E-mail</i>	258
<i>Organizing the Form Elements</i>	260
<i>Creating Text Boxes</i>	262
<i>Creating Password Boxes</i>	263
<i>Formally Labeling Form Parts</i>	264
<i>Creating Radio Buttons</i>	265
<i>Creating Menus</i>	266
<i>Creating Checkboxes</i>	268
<i>Creating Larger Text Areas</i>	269
<i>Allowing Visitors to Upload Files</i>	270
<i>Creating Hidden Fields</i>	271
<i>Creating the Submit Button</i>	272
<i>Resetting the Form</i>	274
<i>Using an Image to Submit Data</i>	276
<i>Setting the Tab Order in a Form</i>	277
<i>Adding Keyboard Shortcuts</i>	278
<i>Disabling Form Elements</i>	279
<i>Keeping Elements from Being Changed</i>	280

Chapter 18: Video, Audio, and other Multimedia281

<i>Of Plugins and Players</i>	<i>282</i>
<i>Getting Players for Your Visitors</i>	<i>283</i>
<i>Getting Multimedia Files</i>	<i>284</i>
<i>Linking to Multimedia Files</i>	<i>285</i>
<i>Embedding QuickTime Movies for Windows</i>	<i>286</i>
<i>Embedding QuickTime Movies for Everyone besides IE</i>	<i>288</i>
<i>Embedding QuickTime Movies for All</i>	<i>290</i>
<i>Using JavaScript to Call a Movie</i>	<i>292</i>
<i>Scaling a QuickTime Movie</i>	<i>294</i>
<i>Changing the Movie's Background Color</i>	<i>295</i>
<i>Looping a QuickTime Movie</i>	<i>296</i>
<i>Offering a Preview Movie</i>	<i>297</i>
<i>Adding Attributes to Secondary Movies</i>	<i>298</i>
<i>Offering a Sequel</i>	<i>299</i>
<i>Embedding MP3 Audio on a Page</i>	<i>300</i>
<i>Embedding Windows Media, Part 1</i>	<i>302</i>
<i>Embedding Windows Media, Part 2</i>	<i>304</i>
<i>Embedding Flash</i>	<i>305</i>
<i>Embedding Google and YouTube Video</i>	<i>306</i>
<i>Inserting Java Applets</i>	<i>308</i>
<i>Embedding Other Multimedia Files</i>	<i>309</i>
<i>Creating an Automatic Slide Show</i>	<i>310</i>

Chapter 19: Scripts311

<i>Adding an "Automatic" Script</i>	<i>312</i>
<i>Calling an External Automatic Script</i>	<i>313</i>
<i>Triggering a Script</i>	<i>314</i>
<i>Creating a Button that Executes a Script</i>	<i>316</i>
<i>Adding Alternate Information</i>	<i>317</i>
<i>Hiding Scripts from Older Browsers</i>	<i>318</i>
<i>Hiding Scripts from XML Parsers</i>	<i>319</i>
<i>Setting the Default Scripting Language</i>	<i>320</i>

Chapter 20: A Taste of JavaScript321

<i>Adding the Current Date and Time</i>	<i>322</i>
<i>Setting a New Window's Size</i>	<i>323</i>
<i>Changing an Image When a Visitor Points</i>	<i>324</i>
<i>Loading Images into Cache</i>	<i>326</i>

Chapter 21: Symbols and Non-English Characters327

<i>About Character Encodings</i>	<i>328</i>
<i>Declaring Your Page's Character Encoding</i>	<i>330</i>
<i>Declaring a Style Sheet's Encoding</i>	<i>332</i>
<i>Saving Your Page with the Proper Encoding</i>	<i>333</i>
<i>Editing a Page with the Proper Encoding</i>	<i>334</i>
<i>Adding Characters from Outside the Encoding</i>	<i>336</i>
<i>Specifying A Page's Language</i>	<i>338</i>

Chapter 22: Testing and Debugging Web Pages	339
<i>Some Debugging Techniques</i>	<i>340</i>
<i>Check the Easy Stuff First!</i>	<i>341</i>
<i>Checking the Easy Stuff: HTML</i>	<i>342</i>
<i>Checking the Easy Stuff: XHTML</i>	<i>343</i>
<i>Checking the Easy Stuff: CSS</i>	<i>344</i>
<i>Validating Your Code</i>	<i>345</i>
<i>Testing Your Page</i>	<i>346</i>
<i>When the Browser Displays the Code</i>	<i>348</i>
<i>When Images Don't Appear</i>	<i>349</i>
<i>Differences from Browser to Browser</i>	<i>350</i>
<i>Still Stuck?</i>	<i>351</i>
Chapter 23: Publishing Your Pages on the Web	353
<i>Finding a Host for Your Site</i>	<i>354</i>
<i>Getting Your Own Domain Name</i>	<i>355</i>
<i>Transferring Files to the Server</i>	<i>356</i>
Chapter 24: Getting People to Visit	361
<i>About Keywords</i>	<i>362</i>
<i>Explicitly Listing Keywords</i>	<i>363</i>
<i>Providing a Description of Your Page</i>	<i>364</i>
<i>Controlling Other Information</i>	<i>365</i>
<i>Keeping Visitors Away</i>	<i>366</i>
<i>Keeping Pages From Being Archived</i>	<i>367</i>
<i>Creating a Site Map Manually</i>	<i>368</i>
<i>Using Google Sitemaps</i>	<i>369</i>
<i>Submitting Your Site to a Search Engine</i>	<i>370</i>
<i>Improving Your Ranking by Getting Linked</i>	<i>371</i>
<i>Writing Pages That Are Easy to Index</i>	<i>372</i>
<i>Other Techniques for Publicizing Your Site</i>	<i>373</i>
Chapter 25: Syndication and Podcasting	375
<i>What a Feed Looks Like</i>	<i>376</i>
<i>Getting Ready for Syndication</i>	<i>378</i>
<i>Starting an RSS Feed</i>	<i>379</i>
<i>Describing Your Site in the Feed</i>	<i>380</i>
<i>Adding Items to a Feed</i>	<i>382</i>
<i>Add an Enclosure</i>	<i>384</i>
<i>Creating Podcasts for iTunes</i>	<i>386</i>
<i>Validating a Feed</i>	<i>392</i>
<i>Submitting a Podcast to iTunes</i>	<i>393</i>
<i>Publishing your RSS Feed on Your Site</i>	<i>394</i>
<i>Subscribing to an RSS Feed</i>	<i>396</i>
<i>Subscribing to a Podcast with iTunes</i>	<i>397</i>

Appendix A: (X)HTML Reference399

Appendix B: CSS Properties and Values.....409

Appendix C: Intrinsic Events417

Appendix D: (X)HTML Symbols and Characters.....419

Appendix E: Hexadecimals431

Appendix F: (X)HTML Tools433

(X)HTML Editors 434

Images and Graphics..... 436

Graphics Tools..... 436

Index437

Colophon..... 456



INTRODUCTION

The World Wide Web is the Gutenberg press of our time. Just about anyone can create their own Web site and then present it to the Internet public. Some Web pages belong to businesses with services to sell, others to individuals with information to share. You get to decide what your page will be like.

All Web pages are written with some form of HTML. HTML lets you format text, add graphics, sound, and video, and save it all in a text file that any computer can read.

HTML is not hard to learn or to master. It is much more an exercise in careful typing and consistency than in mind-blowingly complicated procedures. You can have a simple HTML page up and running in just a few minutes. And while there are many software programs that will create HTML code for you, writing HTML yourself means you won't have to study new software nor be limited by its features.

In this book, you'll find clear, easy-to-follow instructions that will take you through the process of creating Web pages step by step. It is ideal for the beginner, with no knowledge of HTML, who wants to begin to create Web pages.

If you're already familiar with HTML, this book is a perfect reference guide. You can look up topics in the hefty index and consult just those subjects about which you need more information.

The Internet, the Web, and HTML

Sure, you've heard of the Internet, but what is it exactly? Simply put, the Internet is a collection of computers that are all connected to each other. Many people have 24-hour, high-speed broadband connections—through DSL, cable, or satellite—while others use a modem to link their home computers during a certain amount of time each day. Regardless of the type of connection, once you're on, you and your computer become a part of the Internet and are linked to every other computer that's also connected at that moment.

The World Wide Web, for its part, is much more ethereal. It is an ever-changing, kaleidoscopic collection of hundreds of millions of documents, all of which reside someplace on the Internet and are written in some form of HTML.

HTML, or *HyperText Markup Language*, has two essential features—hypertext and universality. Hypertext means you can create a link in a Web page that leads the visitor to any other Web page or to practically anything else on the Internet. It means that the information on the Web can be accessed from many different directions. Tim Berners-Lee, the creator of the Web, wanted it to work more like a person's brain and less like a static source of data, such as a book.

Universality means that because HTML documents are saved as Text Only files, virtually any computer can read a Web page. It doesn't matter if your visitors have Macintosh or Windows machines, or whether they're on a Unix box or even a handheld device like a Palm. The Web is open to all.

Open but Not Equal

However, while HTML is available to all, that doesn't mean that everyone experiences it the same way. It's something like Central Park in New York City. You and I can both go take a walk there. However, if you live in a pent-house apartment on Fifth Avenue and I sleep on a bench, our view of the park will be quite different.

So it is with HTML. While practically any computer can display Web pages, what those pages actually look like depends on the type of computer, the monitor, the speed of the Internet connection, and lastly, the software used to view the page: the *browser*. The most popular browsers today are Internet Explorer, Firefox, Opera, and Safari with handhelds and PDAs gaining momentum every day. Unfortunately, none of these displays a Web page exactly like the next. So it turns out it's not enough to design a beautiful park, you've also got to worry about your visitor's accommodations.

But as you worry, remember that your control is limited. While the New York City Tourist Board would like to ensure that everyone has a good time in their town, they're not handing out free vouchers for rooms at the Park Plaza Hotel, and some people wouldn't accept them even if they did, preferring instead a bed and breakfast or their sister's house. You get the idea. The moral is this: People will be viewing your pages with vastly different setups. Create your pages accordingly—so that the largest number of visitors can view your page as close to the way you want them to as is possible. This book will show you how.

The Browser Wars

Now imagine what would happen if each hotel and apartment building on Fifth Avenue staked out a bit of Central Park and put a fence around it, limiting access to its own residents. It's bad enough that those of us on park benches can only glimpse in to "exclusive" areas. But, there's also the problem that folks from one hotel can't get to the piece of park that belongs to the other hotel. Instead of a rich, public resource, teeming with rollerbladers, hot dog carts, and strolling elders, the park is divided into small, sterile, isolated lots.

In 1994, Netscape Communications put up the first fences on the Web in the so-called *browser wars*. In order to attract users, they threw universality to the wind and created a set of extensions to HTML that only Netscape could handle. For example, Web surfers using Netscape could view pages with colored text, photographs, and other improvements. Surfers with any other browser would get errors and funny-looking results. Or nothing at all.

But people liked those extensions so much that they flocked to Netscape's "hotel". By 1996, it had become the most popular computer program in the world. Microsoft started fencing in its own chunk of the Web. Again, to attract users they added non-standard extensions that only Internet Explorer, Microsoft's browser, could recognize.

According to The Web Standards Project (www.webstandards.org), founded by a coalition of top-flight designers disgusted with the increasing fragmentation of the Web, at the height of the browser wars, Web designers were wasting an incredible 25% of their time devising workarounds for proprietary tags, writing multiple versions of pages to satisfy each browser, and simply educating their clients about the impossibility of creating certain effects for all browsers. It was a mess.

The Push for Standards

The Web's United Nations is an organization called the World Wide Web Consortium (www.w3.org), often abbreviated as W3C, and directed by the Web's inventor, Tim Berners-Lee. Its aim is to convince the Web community of the importance of universality while attempting to satisfy its thirst for beautiful looking pages. Their work is to remove existing fences and guard against new ones.

Attacks on the Ivory Tower

Lately (mid 2006), there has been a crescendo of rising voices complaining about the W3C's slow pace, overemphasis of the abstract, and lack of concrete results. Many Web designers, including those who led the charge for standardization, feel ignored by the W3C and its corporate backers.

Almost seven years after HTML 4.01 and XHTML became *Official Recommendations*, there is no consensus from the W3C on where we go from here. Almost eight years after CSS2 became an Official Recommendation, there is not a single browser that fully supports it, despite the fact that every major browser was developed by a member of the W3C. CSS3 is still in *Working Draft* stage, and perhaps years from completion, let alone implementation.

Some designers have taken matters into their own hands, creating extensible standards-based solutions (<http://microformats.org/>). For more, see Jeffrey Zeldman's article "An Angry Fix" (<http://www.zeldman.com/2006/07/17/an-angry-fix/>) and John Oxtan's "No I am not bloody sorry" (<http://josbuaink.com/blog/753/no-i-am-not-bloody-sorry>).

What should you do meanwhile? For the time being, I recommend what I've always recommended: moderation. Follow the standards but don't be a slave to them. Even Ivory Soap is only 99.4% pure.

The W3C's membership list (<http://www.w3.org/Consortium/Member/List>) reads like a *Who's Who* of movers and shakers on the Web and includes such longtime players as Apple (of iTunes and iPod fame, among others), Adobe (maker of important Web design tools like Photoshop), America Online (which absorbed Netscape Communications as it imploded in 1998), Opera (makers of the Opera browsers for desktop computers and handhelds), and Microsoft (whose Internet Explorer browser took over the #1 spot from Netscape and hasn't looked back), and more modern companies like Google (the ultra-popular search engine and more), and Mozilla Corporation (makers of the popular open source Firefox browser that is the first competition Explorer has had in years). The idea is that these companies come together and agree on the standards and then try to differentiate their products with speed, ease of use, price, or other features that don't turn the Web back into the tower of Babel.

HTML 3.2: Standardization begins

The W3C's first answer to the Web's balkanization was to standardize the proprietary extensions, including some in the official specifications and removing others altogether. At the same time, they encouraged browser manufacturers to support the official HTML specifications as closely as possible, so that a Web page written to standards would behave the same way across browsers.

HTML 4 and CSS

The W3C's next move was much more bold. The old version of HTML joined content, structure, and formatting instructions in a single document, which was simple but not very powerful. The W3C envisioned a new system in which formatting instructions could be saved separately from the content and structure and thus could be applied not just to a single paragraph or Web page but to an entire site, if so desired. So, in the new HTML version 4, the W3C marked most of the formatting elements for future removal from the specifications. These elements would henceforth be *deprecated*, and their use discouraged. At the same time, they created the new system for formatting instructions—called *Cascading Style Sheets*, or *CSS*—to fill the gap.

The original specifications for Cascading Style Sheets mostly limited themselves to recreating HTML effects. CSS Level 2, published in 1998 and lightly updated to Level 2.1 in 2006, however, brought new capabilities, in particular the ability to position elements on a Web page with great precision. CSS could now not only recreate HTML's formatting, it could make professional looking layouts.

However, between proprietary extensions and just plain sloppy code, HTML pages themselves were still a mess. Most browsers bent over backward to accommodate them, always in slightly different ways, which just made the whole situation worse. And there was still no standard system for adding new features. HTML was simply not a sturdy enough platform upon which to build. The W3C decided that we all needed a bit of structure. Their answer was XML, or *Extensible Markup Language*.

XML and XHTML

From the outside, XML looks a lot like HTML, complete with tags, attributes, and values. But rather than serving as a language just for creating Web pages, XML is a language for *creating other languages*. You can use XML to design your own custom markup language which you can then use to format your documents. Your custom markup language will contain tags that actually describe the data that they contain.

And herein lies XML's power: If a tag identifies data, that data becomes available for other tasks. A software program can be designed to extract just the information that it needs, perhaps join it with data from another source, and finally output the resulting combination in another form for another purpose. Instead of being lost on an HTML-based Web page, labeled information can be reused as often as necessary.

But, as always, power comes with a price. XML is not nearly as lenient as HTML. To make it easy for XML *parsers*—software that reads and interprets XML data—XML demands careful attention to upper- and lowercase letters, quotation marks, closing tags, and other minutiae. In addition, there are billions of Web pages already written in HTML and millions of servers and browsers that already know how to read them.

The solution was quite clever. The W3C rewrote HTML *in* XML. This new language had all of the features of HTML and thus could be understood by every browser on the planet. And since its entire lexicon came from HTML, people who already knew HTML only had to learn a few basic syntax rules before they were off and running. And at the same time, since it used XML's syntax, it gained all of XML's power and flexibility and was a perfect foundation for CSS. It was to be the best of both worlds. Its name? XHTML.

CSS and Browser Support

While XHTML and CSS are a powerful combination, there is one small wrench that has continued to plague Web designers: browser support. While it didn't seem to be much of a problem to add extensions willy-nilly, when it comes down to serious, full support of the specifications, no browser has yet been up to the task. However, it's important to note that they've come a long way.

Netscape 6, completely reformed from its extension-madness days, now boasts good CSS support. Too bad its user base is down to less than 1 percent. Firefox 1.5, the Open Source dynamo which rose from the ashes of Netscape's demise (and was even called Phoenix and Firebird early in its history), has excellent CSS support as does Opera 9, whose user base is expanding by leaps and bounds particularly in the handheld and mobile telephone markets. And Internet Explorer, currently the most used browser, has steadily improved its CSS support, although it still has a number of glaring bugs and what sometimes seem like arrogant and obstinate omissions.

All in all, most users use browsers that support CSS either well, or very well. While the number of users on legacy browsers a few years ago might have given folks pause before contemplating a switch to CSS, that number has dwindled below 5% (some say below 2%) and continues to fall. And even many of these are on Internet Explorer 5.5 whose support, though not stellar, was really not that bad.

In short, there's never been a better time to move confidently over to CSS.

XHTML vs. HTML: What Should You Use?

And now an admission. I liked HTML. I thought it was great that you didn't have to obsess over punctuation. Maybe I'm just lazy, but I honestly believe that the Web's popularity is due in part to the fact that browsers cut us all some slack. It made it easy to write Web pages, and so all of us did. Now, a couple of billion pages later, perhaps it's time to change our ways. Or perhaps not.

There are a lot of people out there that will tell you that HTML is evil and XHTML is the *only* solution. I think that's silly. XHTML is a great improvement over HTML. It's stronger, more flexible, more powerful, more likely to be supported in the future, and can be expanded to fit any need. But I'll tell you something. Sometimes you don't need to fill every need. Sometimes, you just want to publish a simple page without stressing over every last quotation mark.

Luckily there is a lot of middle ground. There are actually three standard flavors of both HTML and XHTML. The first, called *transitional*, allows the use of the deprecated tags. The second, called *strict*, prohibits the use of any of the deprecated tags. The third flavor, *frameset*, allows both the use of deprecated tags and the use of frames, which have fallen into such disfavor that I've moved the chapter that describes them out of the book and onto my Web site (*see page 25*). You can combine each of these flavors in varying degrees with CSS. Which combination you choose may depend on several factors. (Keep reading.)

Deciding between HTML, XHTML, and CSS:

While I don't recommend using proprietary extensions—since they leave out part of your audience—there are a lot of other options. Here are some guidelines.

- The bigger the site, the more important it is that you use CSS and XHTML. The former makes it easy to apply, edit, and update formatting across the entire site; the latter gives your page the structure it needs to make sure it lasts into the future.
- Many companies and government agencies, including the U.S. government, require that your Web page fulfill specific *accessibility* requirements in order to make their sites available to people with disabilities. In these cases, you should adhere as closely as possible to XHTML strict, with CSS for formatting. And be sure to check the company's or agency's pertinent guidelines for details in your particular case.
- Large commercial sites that want to reach the widest audience may opt for transitional XHTML, taking advantage of some deprecated tags' practically universal support, while banking on XHTML's rock-solid stability. These kinds of sites will very likely shift to the more powerful CSS as their comfort level with it grows.
- Small or personal sites may want to take advantage of HTML's easy-going syntax along with CSS's powerful formatting and an occasional deprecated tag where necessary.
- My personal choice is to use XHTML and CSS and a bare minimum of deprecated tags.

XHTML considered dangerous?

There are some who question the move to XHTML. The problem stems from the fact that in order for XHTML to be backwards compatible and work in older browsers, one small concession had to be made: it had to be sent from the server in a way that browsers already understood: labeled as *html*. The idea was that as browsers evolved, they would eventually be able to understand XHTML pages served as *xhtml*.

Unfortunately, that just hasn't happened. As of mid 2006, Internet Explorer 7, which will most likely assume the #1 browser mantle from IE 6 once it comes out of beta, still cannot understand XHTML files served as *xhtml*. That means that designers can still not take advantage of XML's strength and even worse, according to Ian Hickson, in <http://bixie.ch/advocacy/xhtml/>, that pages written in XHTML and served as *html* can be more of a hindrance to the push toward standards than a help. He suggests that we should stick with HTML until browsers can serve *xhtml*.

But then, of course, we're stuck with the snake who devours her own tail. Personally, I favor moving towards XHTML and its promise of standardization and power rather than sticking with HTML until some mythical future when browsers will lead the way toward standards. If we all write in XHTML now, it will be in the browser manufacturers' interest to support XHTML. And then we will all reap the benefits that it promises.

```

<fieldset id="personal">
<label>Name:</label><input type="text"
name="name" size="30" /> <br />
<label>Address:</label><input type="text"
name="address" size="30" /> <br />
<label>Town/City:</label><input type="text"
name="city" size="30" /> <br />
<label>State:</label><input type="text"
name="state" size="2" maxlength="2" /><br />
<label>Zipcode:</label><input type="text"
name="zip" size="5" maxlength="5" /> <br />
<label>Customer ID:</label><input
type="password" name="code" size="8" />
</fieldset>

```

Figure i.1 On many pages, you'll find a snippet of XHTML code, with the pertinent sections highlighted in blue.

```

#personal label {position:absolute; left: 20px; font-
size: 90%; padding-top: .2em;}
input {margin-left: 9em; margin-bottom:.2em;
line-height: 1.4em; }

```

Figure i.2 If the CSS code is relevant to the example, it is shown in its own box, again with the pertinent sections highlighted in blue.

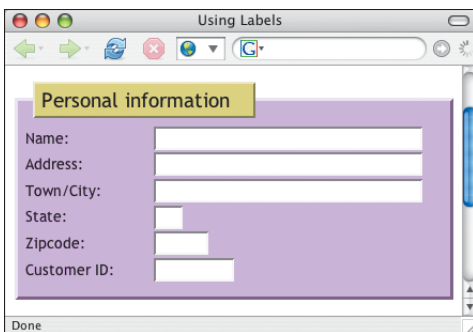


Figure i.3 The XHTML and CSS are then displayed in one or more browsers so you can see how it looks in real life. (This example is from page 264.)

How This Book Works

If you've ever been to a different part of your country than where you're from, you've probably noticed how the folks there talk, well, a little funny. They use different words or they say them with a different accent. And yet, you understand them just fine even if you chuckle about it in the car afterwards. That's the way it is with HTML and XHTML. In their case, they share *precisely* the same vocabulary (to the letter) but have a slightly different syntax.

Since they are so similar, I'll teach you HTML and XHTML at the same time. I'll start by explaining the syntax differences that distinguish them. And then throughout the book I will explain the vocabulary that they share. In those explanations, I use the stricter XHTML syntax (**Figure i.1**). You can either use it as is (to write XHTML), or opt for the looser HTML syntax (to write HTML). It's up to you.

It would be tiresome to have to refer to *HTML* and *XHTML* all the time, so I have chosen to use the abbreviated (X)HTML to refer to both at once. In the few instances I use one of the individual names, you'll know that the information pertains to that language only and not to the other.

CSS is incorporated into the descriptions of (X)HTML—again, that means, *both* HTML and XHTML—as a natural extension and yet a separate tool. While the information about CSS is concentrated in Chapters 7–14, you'll find bits and pieces throughout the book, next to the part of (X)HTML to which it is most applicable (**Figure i.2**).

In this book, I have included illustrations from the major browsers on both Windows and Mac (**Figure i.3**). While you may stick with one browser, there's no telling what your visitors will use. I recommend getting used to how other browsers show (X)HTML.

What's Changed in the Sixth Edition

The first edition of this book, published in 1996, had 11 chapters, 2 appendices, and just 176 pages. The sixth edition in your hands has 25 chapters, 6 appendices and more than 450 pages. This book has expanded and adapted as (X)HTML and CSS have grown and changed.

What's new

The curious thing this time around is that (X)HTML and CSS have not changed considerably since the last edition. That does not mean, however that the book is just a rehashing of that earlier edition, because indeed the Web itself has changed dramatically. In the three short years since the Fifth Edition, we have seen a maturing of CSS layout techniques, which we'll explore with brand new code examples in particular in Chapter 11 but throughout the CSS chapters in general, a surge in Web pages being rewritten with CSS in order to be viewed in handhelds and mobile telephones (Chapter 13), the move away from Perl/CGI in favor of PHP along with CSS formatting of form elements (Chapter 17), a veritable explosion of audio and video, which I'll help you deal with in Chapter 18, and a move toward frequently updated blogs which has led to syndication, RSS feeds, and podcasting, which we'll tackle in Chapter 25.

And although many of the other chapters have the same titles, all have been completely updated to reflect the latest browsers, the most standard XHTML and CSS techniques, and the fact that the book is now printed in glorious full color!

Internet Explorer 7

Internet Explorer 7 was still in beta as this book went to press, and so the illustrations that show Internet Explorer 7 are actually Internet Explorer 7 *beta 2* and *beta 3*. It is possible, though unlikely, that the display of (X)HTML and CSS will change slightly when the final program is released.

What's gone

Finally, I also made the difficult decision to completely remove four chapters. These are the chapters from the Fifth Edition on frames, WML (which has been supplanted by XHTML+CSS), and the two Old Way chapters on deprecated and little used formatting tags, and deprecated and even less used layout tags.

Although most of the elements described in those chapters are still considered valid though deprecated (X)HTML, they have fallen so far out of favor that few self-respecting Web page creators would touch them. I suggest you avoid them as well. However, for historical reasons as well as for completeness (perhaps you'll meet them somewhere and need to know how they work), I will make those chapters available on my Web site for download as a PDF. You can find them at <http://www.cookwood.com/html6ed/oldway>. The user name is *oldway*, and the password is *di7nosaur*.

The HTML VQS Web Site

With the Web constantly changing, it seemed most appropriate to add a dynamic element to this book: the HTML VQS 6th edition Web site (<http://www.cookwood.com/html/>).

On my site, you'll find the full source code for every one of the examples in this book, including the (X)HTML and the CSS (<http://www.cookwood.com/html6ed/examples/>), a list of errata, updates, articles, reviews and comments, and even the full table of contents and index.

There are also several resources available on my site that I hope you'll enjoy, including color tables, symbol and character tables, hexadecimal tables, and complete lists of both (X)HTML elements and attributes and of CSS properties and values.

Next, as I was writing this book, I amassed a collection of lesser tips and tricks that simply didn't fit on the appropriate page. I've made them all available on the site.

Finally, you'll find a lively Question and Answer Forum (www.cookwood.com/html/qanda) where you can post your most vexing questions—and easy ones, too. While I hang out there and will do my best to answer, there is a dedicated team of Web designers who usually beat me to the punch. If you're so inclined, feel free to step in and answer questions yourself. Your help will be greatly appreciated.

See you on the Web!

BASIC (X)HTML FORMATTING

While it's a good idea to try to separate formatting from content and to use style sheets for controlling the appearance of your page, there are a few simple ways to format text in (X)HTML that are handy to know.

When should you use basic (X)HTML formatting instead of CSS? There are two main situations. First, most of the elements discussed in this chapter are *logical* elements, that is, they give structure to your document by describing what they contain. For example, the `code` element is specifically designed for formatting lines of code from a script or program. While it formats such content in a monospace font, it also more importantly *identifies* the text as code.

The second reason to use the basic formatting elements in this chapter is because CSS is sometimes too big a bazooka for the job. If you want to highlight a word or phrase on your page, instead of enclosing it in a `span` element with a particular class and then creating a style sheet for that class, you can just wrap it in a simple formatting element and be done with it.

There are a number of formatting elements—for changing the font, size, and color, for example—that, while still technically legal and widely supported, are being phased out of (X)HTML in favor of style sheets. You can find more information about them in Appendix A, *(X)HTML Reference* as well as on my Web site (*see page 25*).

Making Text Bold or Italic

One way to make text stand out is to format it in boldface or italics.

To make text bold:

1. Type ``.
2. Type the text that you want to make bold.
3. Type ``.

To make text italic:

1. Type `<i>`.
2. Type the text that you want to make italic.
3. Type `</i>`.

✓ Tips

- You can also use the less common `em` and `strong` tags to format text (**Figures 4.2**). These are *logical* formatting tags for “emphasizing” text or marking it as “strong”. In most browsers, `em` is displayed in italics and `strong` in bold. On some browsers, though, they’re displayed differently. For example, some handhelds can’t display italics and use underlining instead.
- You may also use `cite` (for citations), `dfn` (for definitions), and `var` (for variables) to make text italic while giving information about the content.
- The `address` tag—old-fashioned but still valid—is another logical tag for making text italic. It’s usually only used to format the Web page designer’s email address.
- For more control over bold and italics, try style sheets. For details, consult *Creating Italics* on page 154 and *Applying Bold Formatting* on page 155.

```
<h1>Barcelona Night Life</h1>
```

```
<p>Barcelona is such a great place to live. People there really put a premium on <b>socializing</b>. Imagine it being more important to go out with your friends than to get that big promotion. Even when you're, gasp, <i>pushing 30</i>. They say there are more bars in Barcelona than in the rest of the European community <i>combined</i>.</p>
```

Figure 4.1 You may use bold or italic formatting anywhere in your (X)HTML document, except in the title.

```
<h1>Barcelona Night Life</h1>
```

```
<p>Barcelona is such a great place to live. People there really put a premium on <strong>socializing</strong>. Imagine it being more important to go out with your friends than to get that big promotion. Even when you're, gasp, <em>pushing 30</em>. They say there are more bars in Barcelona than in the rest of the European community <em>combined</em>.</p>
```

Figure 4.2 If you prefer, you can use the logical `em` and `strong` tags to add both structure and formatting at the same time.

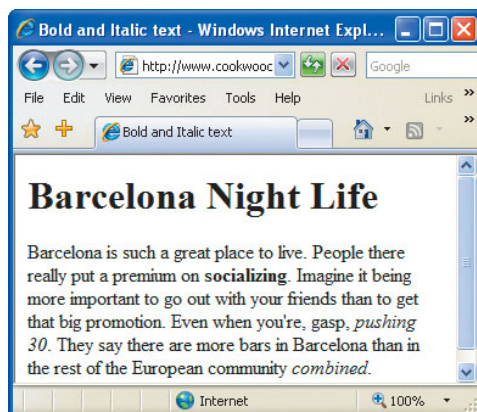


Figure 4.3 Bold and italic formatting are the simplest and most effective ways to make your text stand out.


```

<h1>Barcelona Night Life</h1>

<p>Barcelona is such a great place to live. People
there really put a premium on <strong>socializing
</strong>. Imagine it being more important to go
out with your friends than to get that big promotion.
Even when you're, gasp, <em>pushing 30</em>.
They say there are more bars in Barcelona than in
the rest of the European community
<em>combined</em>.</p>

<p><big>Don't get me wrong.</big> I don't mean
that everyone gets drunk all the time--bars are for
hanging out and talking or for having a cup of
coffee (espresso, of course).</p>

<p><small>The opinions expressed on this page
are mine and mine alone. </small></p>

```

Figure 4.4 The `big` and `small` tags are a fast and easy way to make text stand out.

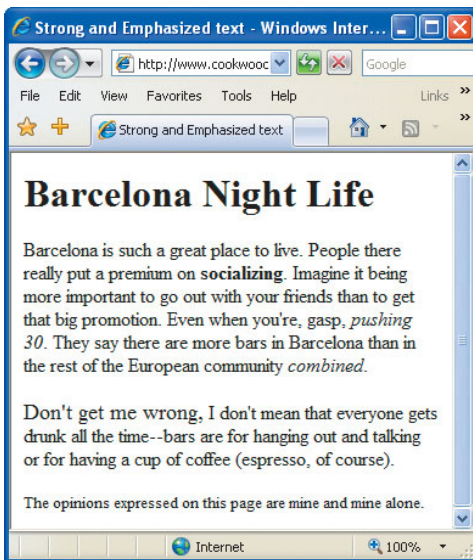


Figure 4.5 The `big` and `small` elements enjoy wide support. They have identical effects in most browsers. (This is Internet Explorer 7 for Windows.)

Changing the Size of Text

The `big` and `small` tags change the relative size of a given word or phrase with respect to the surrounding text.

To make the text bigger or smaller than the surrounding text:

1. Type `<big>` or `<small>`.
2. Type the text that should be bigger or smaller.
3. Type `</big>` or `</small>` depending on the tag used in step 1.

✓ Tips

- Of course, “big” and “small” are relative, and the specifications do not dictate just how much bigger or smaller browsers are supposed to make the text. In general, they stick to typical font sizes, like 8, 9, 10, 12, 14, 16, 18, 24, 36, and 48, moving one step up or down the ladder depending on the element used. The default size for most browsers is 16px.
- Although the `big` and `small` tags have not been deprecated in (X)HTML, you may still want to use style sheets in order to have more control over the size of the text. For more information, consult *Setting the Font Size* on page 156.
- Both the `big` and `small` tags have a cumulative effect if used more than once. So `<small><small>teensy text</small></small>` would be *two* sizes smaller than surrounding text.

Using a Monospaced Font

Every visitor to your page has two fonts specified in their browser's preferences: one regular, proportionally spaced one, and the other monospaced, like a typewriter's text. These are usually Times and Courier, respectively. If you are displaying computer codes, URLs, or other information that you wish to offset from the main text, you might want to format the text with the monospaced font.

Use `code` for formatting computer *code* in languages like C or Perl. The `tt` element (it stands for *typewriter text*) is for general monospaced text. Use `kbd` for formatting *key-board* instructions. And `samp` is for displaying *sample* text. None of these tags is used very often. The truth is that monospaced text is kind of ugly.

To format text with a monospaced font:

1. Type `<code>`, `<tt>`, `<kbd>`, or `<samp>`.
2. Type the text that you want to display in a monospaced font.
3. Type `</code>`, `</tt>`, `</kbd>`, or `</samp>`. Use the tag that matches the one you chose in step 1.

✓ Tips

- Remember that the monospaced font tags will not have a very dramatic effect in browsers that display all their text in monospaced fonts (like Lynx: <http://www.delorie.com/web/lynxview.html>).
- You can also format several lines of monospaced text with the `pre` tag (see page 73).
- You can apply *any* font (that your visitor has installed) to your text with styles (see page 152).

```
<h2>Perl Tutorial, Lesson 1</h2>
```

```
<p>If you're on a UNIX server, every Perl script should start with a shebang line that describes the path to the Perl interpreter on your server. The shebang line might look like this:</p>
```

```
<p><code>#!/usr/local/bin/perl</code></p>
```

Figure 4.6 The `code` element not only formats its contents with a monospaced font but also indicates that the contents are computer code. It's a logical tag.



Figure 4.7 Monospaced text is perfect for URLs and computer code.

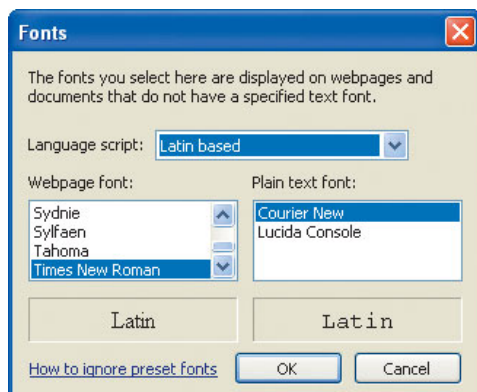


Figure 4.8 Text tagged with `code`, `kbd`, `samp`, or `tt` will be displayed in the font that your visitors have chosen for monospaced text in their browser. The Fonts box from IE 7 (shown) comes up when you choose Tools > Internet Options and then click the Fonts button in the General tab.

```
<body>

<p>Here's the first part of the Cat and Otter Bistro
script (see the WAP/WML chapter), where the
variables are declared, and the $number variable
is screened to make sure it's actually a number:

<pre>my $number = param('number');
my $smoke = param('smoke');
my $dinner_index = param('dinner_index');

$number =~ /[0-9]*/ ;
$number = $1;

</pre>
```

Figure 4.9 The `pre` element is ideal for text that contains important spaces and line breaks, like the chunk of Perl CGI code shown above.

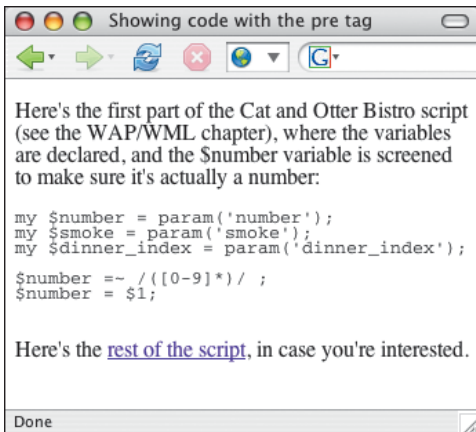


Figure 4.10 Notice how the line breaks, including the extra return between the third and fourth lines of code, are maintained.

Using Preformatted Text

Usually, browsers collapse all extra returns and spaces and automatically break lines according to the size of the window. Preformatted text lets you maintain the original line breaks and spacing that you've inserted in the text. It is ideal for computer code examples.

To use preformatted text:

1. Type `<pre>`.
2. Type or copy the text that you wish to display as is, with all the necessary spaces, returns, and line breaks.
3. Type `</pre>`.

✓ Tips

- Preformatted text is generally displayed with a monospaced font like Courier. You can use styles to change the font, if you like (*see page 152*).
- If what you want to display contains (X)HTML elements, you'll have to substitute the appropriate character entities for the greater than and less than signs (namely `>` and `<`, respectively). Otherwise the browser will try to display those elements; the `pre` tag works no magic on them. For more information, consult *Adding Characters from Outside the Encoding* on page 336.
- You can also use styles to maintain line breaks and spaces (*see page 164*).
- Note that `pre` is block-level while the tags on page 72 are all inline.

Quoting Text

There are two special tags for marking quoted text so that you can identify its author, origin, and language. Block-level quotes are generally indented by browsers. Inline quotes are supposed to be automatically enclosed in quotation marks but are not widely supported.

To quote block-level text:

1. Type `<blockquote>` to begin a block-level quote.
2. If desired, type `cite="url"`, where *url* is the address of the source of the quote.
3. Type `>` to complete the opening tag.
4. Type the text that you wish to appear set off from the preceding and following text, including any desired (X)HTML tags.
5. Type `</blockquote>` to complete the element.

To quote inline text:

1. Type `<q>` to begin.
2. If desired, type `xml:lang="xx" lang="xx"`, where *xx* is the two-letter code for the language the quote will be in. This code is supposed to determine the type of quote marks that will be used (“” for English, “” for many European languages, etc.).
3. Type `>` to complete the opening tag.
4. Type the text that should be quoted.
5. Type `</q>`.
6. If desired, in the `html` tag, add `xml:lang="xx" lang="xx"`, where *xx* is the two-letter code for the language that most of your Web page is in.

```
<p>Sometimes I get to the point where I'm not sure anything matters at all. Then I read something like this and I am inspired: </p>
```

```
<blockquote cite="http://www.kingsolver.com">
```

```
<p>It's not hard to figure out what's good for kids, but amid the noise of an increasingly antichild political climate, it can be hard to remember just to go ahead and do it: for example, to vote to raise your school district's budget, even though you'll pay higher taxes. ... To volunteer time and skills at your neighborhood school and also the school across town. To decide to notice, rather than ignore it, when a neighbor is losing it with her kids, and offer to babysit twice a week. This is not interference. Getting between a ball player and a ball is interference. The ball is inanimate.</p>
```

```
</blockquote>
```

```
<p>This is from Barbara Kingsolver's brilliant collection of essays, <cite>High Tide in Tucson</cite> (1995, HarperCollins)</p>
```

Figure 4.11 A block quote can be as short or as long as you need. You can even divide it into various paragraphs by adding `p` tags as necessary.

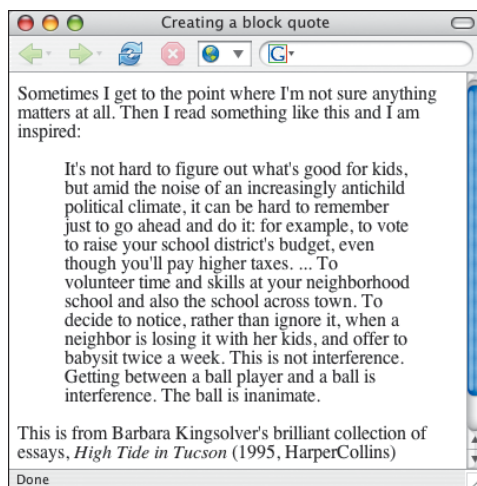


Figure 4.12 Block quotes are generally indented from both sides. The `cite` attribute is not yet recognized by any browser I've seen.

```
<html xmlns="http://www.w3.org/1999/xhtml"
xml:lang="en lang="en">
[snip]
<p>And then she said <q>Have you read
Kingsolver's <q>High Tide in Tucson</q>? It's
inspiring.</q></p>
<p>She tried again, this time in French: <q
lang="fr">Avez-vous lu le livre <q lang="fr">High
Tide in Tucson</q> de Kingsolver? C'est
inspirational.</q></p>
```

Figure 4.13 The `lang` attribute in the `html` tag is supposed to be a default for the other tags. In my tests, it has no effect on the `q` element.

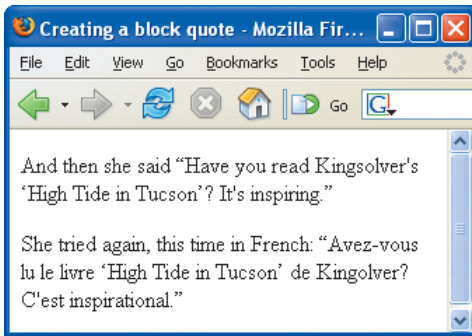


Figure 4.14 Firefox adds curly double quotes around `q` elements and curly single quotes around nested `q` elements.

✓ Tips

- Text and inline elements should not be placed directly between the opening and closing `blockquote` tags. Instead, enclose the text and inline elements in a block-level tag—like `p`, for example—*within* the `blockquote` tags.
- You can nest both `blockquote` and `q` elements. Nested `q` elements should automatically have the appropriate quotation marks—in English the outer quotes should be double and the inner ones should be single.
- Proper support for `q` varies widely from one browser to the next. Firefox offers double and single curly quotes on Mac and Windows. Opera 8 uses straight double quotes for everything, including nested `q` elements. Explorer for Windows (up to and including version 7) ignores the `q` element completely.
- No current browsers pay any attention to the `lang` attribute, either in the `q` element or in the `html` element. Internet Explorer 5 for Macintosh was the only browser I've found that ever did.
- The `c i t e` attribute may also be used with the `q` element, although it makes less sense. I haven't seen a browser that does anything with it in either element.
- For more details on the `xml:lang` and `lang` attributes, consult *Specifying A Page's Language* on page 338.
- You can find a complete list of language codes at <http://www.w3.org/WAI/ER/IG/ert/iso639.htm>.

Creating Superscripts and Subscripts

Letters or numbers that are raised or lowered slightly relative to the main body text are called *superscripts* and *subscripts*, respectively. (X)HTML includes tags for defining both kinds of offset text.

To create superscripts or subscripts:

1. Type `<sub>` to create a subscript or `<sup>` to create a superscript.
2. Type the characters or symbols that you wish to offset relative to the main text.
3. Type `</sub>` or `</sup>`, depending on what you used in step 1, to complete the offset text.

✓ Tips

- Most browsers automatically reduce the font size of a sub- or superscripted character by a few points.
- Superscripts are the ideal way to format certain foreign language abbreviations like M^{lle} for *Mademoiselle* in French or 3^a for *Tercera* in Spanish.
- Subscripts are perfect for writing out chemical molecules like H₂O.
- Superscripts are also handy for creating footnotes. You can combine superscripts and links to make active footnotes (the visitor jumps to the footnote when they click the number or asterisk). For more information, see Chapter 6, *Links*.
- Super- and subscripted characters gently spoil the even spacing between lines (Figure 4.16). You can remedy this by changing the size of the sub or sup text (see pages 71 and 156) and adjusting its line height (see page 158).

```
<body>
<h1>Famous Catalans</h1>

<p>When I was in the sixth grade, I played the
cello. There was a teacher at school who always
used to ask me if I knew who "Pablo Casals" was. I
didn't at the time (although I had met Rostropovich
once at a concert). Actually, Pablo Casals' real
name was <i>Pau</i> Casals, Pau being the
Catalan equivalent of Pablo<sup>1</sup></sup>.</p>

<p>In addition to being an amazing cellist, Pau
Casals is remembered in this country for his
empassioned speech against nuclear proliferation
at the United Nations<sup>2</sup></sup> which he
began by saying "I am a Catalan. Catalonia is an
oppressed nation."</p>

<p><sup>1</sup></sup>It means Paul in English.<br />
<sup>2</sup></sup>In 1963, I believe.</p>
</body>
```

Figure 4.15 The opening sup or sub tag precedes the text to be affected.

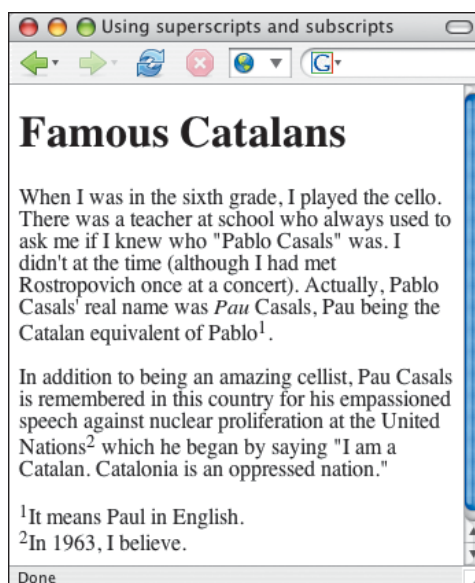


Figure 4.16 Unfortunately, the sub and sup elements spoil the line spacing. Notice that there is more space between lines 6 and 7 of the first paragraph and lines 4 and 5 of the second than between the other lines.


```
<body>
<p><big>I promise to do all of my homework,<ins>
all of my chores,</ins> clean the cat litter, and not
watch more than <del>six</del> a half
hour<del>s</del> of tv.</big></p>

<p>signed</p>
```

Figure 4.17 You have to be a little bit careful to include the associated punctuation with the `ins` and `del` elements.

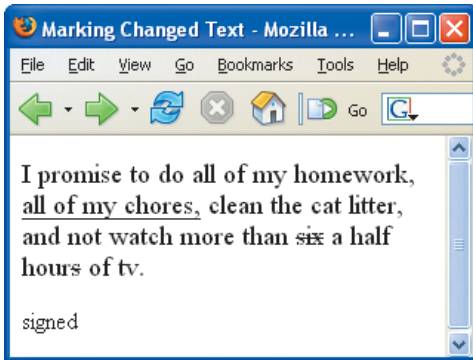


Figure 4.18 The changes become quite apparent.

Marking Changed Text

Another set of logical tags that you might find useful are the ones for marking text that has changed from one version to the next. Lawyers and bloggers do this all the time.

To mark newly inserted text:

1. Type `<ins>`.
2. Type the new text.
3. Type `</ins>`.

To mark deleted text:

1. Place the cursor before the text you wish to mark as deleted.
2. Type ``.
3. Place the cursor after the text you wish to mark as deleted.
4. Type ``.

✓ Tips

- Text marked with the `ins` tag is generally underlined. Since links are often underlined as well (if not in your site, in many others), this may be confusing to visitors. You may want to add an explanation at the beginning of your page and/or use styles to change how inserted passages (or links) are displayed (*see page 151*).
- Text marked with the `del` element is generally stricken out. Why not just erase it and be done with it? Striking it out makes it easy for others to see what has changed.
- You can also use styles to underline and strike out text (*see page 168*). The advantage of the `ins` and `del` elements is that they identify the text as being *inserted* or *deleted*, and not just underlined or stricken.

Explaining Abbreviations

Abbreviations and acronyms (an abbreviation that can be pronounced as a word) abound. Unfortunately, people use them so often that they sometimes forget that not everyone knows what they mean. You can use the `abbr` and `acronym` elements to add meaning to the abbreviation or acronym in question without breaking the flow of your Web page or distracting your readers with extra links.

To explain abbreviations:

1. Type `<abbr>`.
Or type `<acronym>` if the abbreviation can be pronounced as a word.
2. Next type `title="explanation"`, where *explanation* gives more details about the abbreviation.
3. Type `>`.
4. Then type the abbreviation itself.
5. Finally, finish up with `</abbr>` or `</acronym>` depending on what you used in step 1.

✓ Tips

- Firefox (on both platforms) supports both `abbr` and `acronym`, highlighting both elements with a dotted underline and providing the `title` attribute's contents as a tool tip (**Figure 4.20**).
- Internet Explorer for Windows (up to version 7) doesn't change the display of acronym or `abbr` elements, but does show titles as tool tips. IE 6 and earlier didn't support `abbr`.

```
<p><abbr title="Lyndon Baynes Johnson">LBJ
</abbr> took the <abbr title="Interborough Rapid
Transit">IRT</abbr> down to 4th Street <abbr
title="United States of America">USA</abbr>.

<br />When he got there, what did he see?

<br />The youth of America on <abbr title="d-
Lysergic Acid Diethylamide">LSD</abbr>.</p>

<p>--Hair, the Musical, 1967</p>

<p>Or perhaps you'd rather talk about something
slightly less political, like <acronym title="Light
Amplification By Stimulated Emission of Radiation">
laser</acronym>, or <acronym title="Radio
Detection And Ranging">radar</acronym>, or
<acronym title="Self-Contained Underwater
Breathing Apparatus">scuba</acronym>? </p>
```

Figure 4.19 It seems an awful lot of code for just a few words. Still, it can be very helpful to get immediate information about an abbreviation, at least the first time it is used.

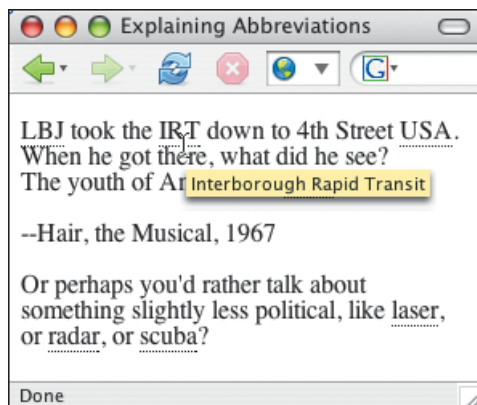


Figure 4.20 Firefox Mac/Win highlights abbreviations and acronyms with a dotted underline and when your visitors hover, the contents of the element's `title` attribute are shown in a tool tip.

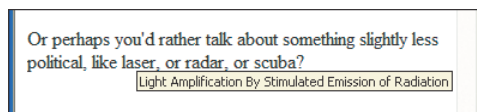


Figure 4.21 Explorer displays the title of abbreviations as a tool tip, but doesn't display the abbreviation itself any differently.


```

<body>
<center>
<h2>The Earth's Core</h2>
<p>At the center of the earth, more than 6000
kilometers from the surface, the temperature is a
toasty 6500 degrees Kelvin.</p>
</center>
<p>Not bad for a little planet.</p>
</body>

```

Figure 4.22 The center element acts like a div element with the align attribute set to center.

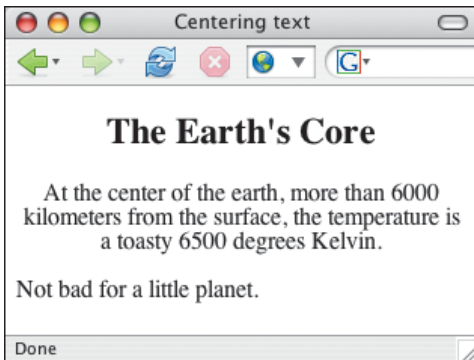


Figure 4.23 The center element continues to be well supported.

Centering Elements on a Page

The center tag is one of those elements that has been deprecated but not forgotten. When centering with CSS seems like too much of a hassle, many continue to use the center tag. It remains well supported and can be used with virtually any element on your page.

To center elements on a page:

1. Type `<center>`.
2. Create the element that you wish to center.
3. Type `</center>`.

✓ Tips

- The center element is nothing more than an abbreviation of `<div align="center"> ... </div>`.
- For more details on dividing your document into sections that you can then align, consult *Breaking up a Page into Divisions* on page 64.
- For more on the (deprecated) align attribute, in particular with paragraphs and headers, see the last tips on pages 61–62.
- If you use the center element, you should make sure to also declare a *transitional* DOCTYPE (see page 56).
- For information on using styles to center text, consult *Aligning Text* on page 165.
- For information on aligning images with text, consult *Aligning Images* on page 100.

This page intentionally left blank

INDEX

Symbols

- & (ampersand) 32
 - and XHTML 337
- * (asterisk)
 - comments in CSS 121
 - for wild card selector 139
- @charset rules 128
- @import rules 132
 - and image replacement 196
 - for handhelds 204
 - for print 210
 - media specific 133
- @media rules 133, 210
- \ (backslash)
 - for escaping characters in style sheets 332
 - in JavaScript 315
- [] (brackets), for selecting based on attributes 147
- , (comma), for grouping selectors 148
- (dashes), in comments 67, 131
- !-- (exclamation point and dashes) 67, 131
- / (forward slash)
 - comments in CSS 121
 - for (X)HTML 28
 - HTML vs. XHTML 38
 - in URL 35
- > (greater than symbol)
 - character reference for 337
 - for child selector 142
 - (X)HTML elements 28
- < (less than symbol)
 - character reference for 337
 - (X)HTML elements 28
- (non-breaking space) 62, 237, 251
 - and empty cells in tables 237
- . (period), for class selector 140
- | (pipe symbol), in selectors 147
- + (plus), for adjacent sibling selectors 143
- # (sharp)
 - anchors 107
 - for id selector 140
- 3D positioning 183
- A**
- a element 104–115
 - for multimedia files 285
- A List Apart, Web magazine 39, 192, 305
 - switching style sheets 130
- abbr element 78
- abbreviations
 - describing 78
 - for languages 338
 - hexadecimal colors 126
- absolute positioning 179
 - See also page layout
- absolute units 125
- absolute URLs 36
 - See also URLs
- accented characters. *See* characters, special
- accessibility 22
 - importance of 39
- accesskey attribute
 - in a element 112
 - in area element 118
 - in form elements 278
- Acrobat 282
- acronym element 78
- acronyms, describing 78
- action attribute, in form element 254
- active images 276
- active links, color of 146
- :active pseudo-class 146
- ActiveX controls
 - and background play 293
 - and object element 282
- Address bar, icon in 102
- address element 70
- Adobe GoLive 435
- Adobe Photoshop and Photoshop Elements 87, 88–89, 436
 - adding space around images 99
 - and image size 92, 95
 - for examples 87
- Adobe Systems, membership in W3C 17
- aggregators
 - description 375, 377
 - subscribing to RSS feeds with 396
- Aladdin Systems. *See* Allume Systems
- align attribute
 - in caption element 229
 - in div element 79
 - in hn tag 61
 - in hr element 101
 - in img element
 - for aligning 100
 - for floating images 96
 - in legend element 260
 - in p element 62
 - in table element
 - for centering 234
 - for text wrap 235
 - in table elements 238–239
- aligning
 - cells 238–239
 - with column groups 246–247
 - header cells 247
 - headers 61
 - images 100
 - paragraphs 62
 - text 61, 62, 165
 - vertically 188

Allume Systems 111
 alt attribute
 in area element 117
 in img element 91
 alternate fonts 153
 alternate style sheets 130
 alternate text 91, 117
 ampersand. *See* Symbols section
 anchors 106–107
 and tables of contents 106
 creating 106
 in headers 61
 links to 107
 with id attribute 63, 106
 animated GIFs 85
 AOL
 free Web hosting services for members 354
 membership in W3C 17
 Apache servers, and character encodings 331
 Apple Computer
 Link Maker 111
 membership in W3C 17
 See also iTunes
 applet element 308
 archived pages, avoiding 367
 area element 117
 arrows, entities for 429
 Artist column, in iTunes 390
 ASCII 32
 limitations of 328
 mode for transferring files 357, 359
 See also text-only format
 Atom, feed format 376
 attaching
 backgrounds to browser window 172
 elements to browser window 180
 attributes 29
 case 38
 selectors based on 147
 audio, embedding on page 300–301
 author
 specifying for podcast 387
 specifying for podcast episode 388
 specifying for RSS feed 382
 specifying for Web page 365
 author element in RSS 382

B

b element 70
 background 161, 172–173
 and deciphering others' tables 228
 for handhelds 203
 hiding for print 211
 of movies, changing 295
 sound 301
 background attribute, in table elements 240–241
 background properties 161, 172–173
 for handhelds 203
 in tables 240–241
 backslash. *See* Symbols section
 banners, hiding in print style sheets 211
 Bare Bones Software 435
 base element 109
 BBEdit 45, 435
 character encodings 333, 335
 syntax checking 345
 Berners-Lee, Tim 14
 bgcolor attribute
 and priority 241
 in table elements 240–241
 big element 71
 blinking text 168
 block-level
 description 30, 171
 vs inline elements 171
 blockquote element 74
 blocks of text, in forms 269
 blogs
 links to site 373
 marking changes 77
 BMP format 86, 349
 body element 58
 HTML vs. XHTML 38
 bold formatting 70, 155, 159
 and handhelds 203
 bookmarks, and title 60
 border attribute
 in img element 90, 114
 in table element 230–231
 border properties 184–185
 for handhelds 203
 in tables 230–231
 border-spacing property 242–243
 borders 184–185
 in handhelds 203, 208
 of images in links 114
 of tables 230–231
 displaying 249
 external 249
 hiding 249
 internal 250
 bottom property 178, 179, 180
 Boutin, Paul 361
 Bowman, Douglas 192, 197
 box model
 description 171
 IE 5.x for Windows bug 174–175
 br element 66
 and lists 217
 and white-space property 164
 clear attribute 98
 in cells 251
 stopping text wrap around images 98
 brackets. *See* Symbols section
 broken image icon 349
 browser wars 16
 in handhelds 199
 browsers 15
 and image size 92
 compatibility 350
 default display 42
 description 15
 hiding scripts from 318
 opening pages in 52, 346
 problems with 348, 350
 standards vs. quirks mode 41
 See also Firefox, Internet Explorer, Opera, Safari
 bugs
 IE 5.x for Windows
 and width 174–175, 233
 IE 6 for Windows
 XML declaration triggers quirks mode 57

Internet Explorer
 background and legend element 261
 borders 185
 borders in tables 231
 Netscape 4.x
 solving with `@import` rules 132

bullets
 and password boxes in forms 263
 in lists 218
 default order 225

button element
 for reset button 275
 for scripts 316
 for submit button 273

buttons
 controlling for embedded MP3s 300
 for scripts 316
 navigational 114
 radio (in forms) 265
 reset (in forms) 274
 with image 275
 submit (in forms) 272
 with image 273

Byte-Order Mark, for Unicode encoding 333

C

cache, loading images into 326
 cached pages, avoiding 367
 capitalize, with styles 166
 caption element 229
 captions
 in forms 260
 in tables 229
 cascade 122–123
See also CSS

case (upper/lower)
 and file names 34
 changing with styles 166
 HTML vs. XHTML 38

categories
 adding to podcast 388
 in RSS 381
 official list for podcasts 389
See also `itunes:category` element

category element in RSS 381
 in `item` element 382

CDATA sections
 for hiding scripts 319
 for hiding style sheets 131
 for inserting (X)HTML in RSS feed 383

Cederholm, Dan 44

cellpadding attribute, in table element 242–243

cells
 aligning 238–239
 background images 240–241
 changing color 240–241
 eliminating space between 242–243
 height 233
 line breaks 251
 padding and spacing 242–243
 spanning across columns 244
 spanning across rows 245
 width 232–233

cellspacing attribute, in table element 242–243

center element 79

centering 79
 headers 61
 in handhelds 208
 tables 234
 text 61, 62, 165
 with `margin` property 234

centimeters, in CSS 125

Chami.com 435

changing pages automatically 310

channel element in RSS 379

character encodings. *See* encodings, character

character references 32, 336
 description 328
 in titles 60

characters, special 327–338
 adding 336–337
 complete list of entities 419–429
 for (X)HTML symbols 420
 in file names 51
 in Web pages 32
 mathematical and technical 425–428
 punctuation 423–424
 shapes and arrows 429
 Western European 420–422

charset attribute, in script element 313

checkboxes 268

checked attribute, in input element
 for checkboxes 268
 for radio buttons 265

child element 31, 171

circled i, in iTunes 391

cite attribute, in blockquote and `q` 74

cite element 70

Clark, Joe 197

class
 assigning to element 63
 choosing by selector 140

class attribute 63
 in `div` element 64
 in `span` element 65

classid attribute, in object element 286, 302

clear attribute, in `br` element, around images 98

clear property 182

clickable images
 client-side maps 117
 in links 114

clip art for Web pages 436

clip property 191

closing tags 28
 HTML vs. XHTML 38

cloud element in RSS 381

code element 72

col element 246–247

colgroup element 246–247

collapsing margins 176

color property 160
 and handhelds 203

- colors
 - and compression 84
 - background 172–173
 - changing 160
 - changing for print 211
 - changing rows and columns 240–241
 - defining
 - in CSS 126
 - in (X)HTML 29
 - in handhelds 203
 - in images 82
 - of borders 184–185
 - of tables 230
 - of links 146
 - See also* inside back cover
 - cols attribute
 - in textarea element 269
 - colspan attribute, in td element 244
 - column groups 246–247
 - for speeding up display 252
 - columns
 - changing color 240–241
 - in tables 244, 246–247
 - spanning cells across 244
 - of text, creating
 - with tables 244–245
 - Comcast, free Web hosting for members 354
 - comma. *See* Symbols section
 - comments
 - adding to page 67
 - conditional (Internet Explorer) 290–291
 - CSS 121
 - documenting structure 170
 - for hiding scripts 318
 - for hiding style sheets 131
 - getting 346
 - comments element in RSS 383
 - compatibility
 - and standards 16
 - browser wars 16
 - tester pages 350
 - compression
 - and colors 84
 - and download time 84
 - of files for download 111
 - of photographs 84
 - computed value 158
 - and inheritance 157, 162, 163
 - conditional comments (Internet Explorer) 290–291
 - containing block, description 175
 - content attribute, in meta element 320, 330, 363, 364, 365, 366
 - content property 213
 - context, and selectors 141–143
 - conventions used in this book 23
 - coords attribute, in area element 117
 - copying other pages 53
 - copyright element in RSS 380
 - copyrights 86
 - See also* Creative Commons
 - specifying for Web page 365
 - Corel
 - Paint Shop Pro 87, 436
 - WordPerfect, and saving (X)HTML file 47
 - Courier 72, 73
 - crawler page. *See* site map 368
 - Creative Commons 86
 - cropping images 95
 - cross-browser compatibility 15, 350
 - See also* individual browsers, standards, bugs
 - CSS
 - and Mobile Web 199
 - beginnings of 18
 - box model 171
 - browser support 20
 - cascade 122–123
 - colors 126
 - comments 121
 - complete list of properties and values 409–416
 - drop-down menus 194–195
 - dynamic effects with 189–197
 - file extension 128
 - for handhelds 202–203
 - positioning 171, 178–183
 - standards vs quirks mode 41
 - URLs 132
 - See also* styles and style sheets
 - cursor property 186
 - cursors, changing with styles 186
 - custom markers, in lists 220, 222
- ## D
- dashes. *See* Symbols section
 - dates
 - adding to page with JavaScript 322
 - formatting in RSS 381
 - dd element 223
 - debugging. *See* problem solving
 - declarations
 - CSS 120
 - DOCTYPE. *See* DOCTYPE declarations
 - deep links 105
 - default display, of (X)HTML 42
 - default pages 49
 - default.htm, in Microsoft servers 49
 - definition lists 223
 - del element 77
 - deprecated, definition 40
 - descendants 31
 - and inheritance 122
 - selectors for 141
 - Description column, in iTunes 390
 - description element in RSS
 - in channel element 380
 - in item element 382
 - description of page, and improving ranking 364
 - designing site 44
 - books on 44
 - dfn element 70
 - DHTML 311
 - digital cameras 86
 - disabled attribute, in form elements 279
 - disabled JavaScript 317
 - disabling form elements 279
 - display property 190, 192, 196
 - for handhelds 205
 - displaying and hiding elements 190–191, 196
 - for handhelds 205
 - div element 64
 - structuring page with 170
 - divisions
 - and structure 170
 - creating 64

dl element 223
 docs element in RSS 381
 DOCTYPE declarations
 and proprietary elements 57
 and URLs 57
 and validators 57, 345
 choosing 40
 description 40
 for transitional (X)HTML 56
 HTML vs. XHTML 57
 on Web site 57
 requirements 38
 standards mode vs. quirks mode 41
 documenting code 67
 domain name, getting yours 355
 downloading
 making files available for 110
 time, keeping to minimum 84
 Dreamweaver. *See* Macromedia Dreamweaver
 drop-down menus, creating with CSS 194–195
 dt element 223
 duration, specifying for podcast episode 390
 dynamic effects with CSS 189–197
 drop-down menus 194–195
 image replacement 196–197
 pop-ups 193
 rollover buttons 192
 dynamically generated pages, and search engines 372

E

EarthLink, free Web hosting services for members 354
 editing Web pages 50
 with proper encoding 334
 editors, (X)HTML 434
 elements 28
 block-level vs. inline 30
 case 38
 choosing by name with selector 139
 naming 63
 nested 31
 parents and children 31
 em element 70
 em unit 124, 157
 and line height 158
 and spacing 162
 and text indents 163
 importance for handhelds 202
 email
 addresses, in URLs 35
 data from forms 258–259
 formatting address 70
 embed vs. object 282
 embedding
 audio 300–301
 Flash movies 305–307
 Google Video and YouTube 306–307
 Java applets 308
 MP3s 300–301
 other files 309
 QuickTime movies 286–299
 Windows Media Player files 302–304
 standards-compliant way 304
 empty elements 28
 HTML vs. XHTML 38

enclosure element in RSS 384
 enclosures, adding to RSS feed 384–385
 Encoded Text, option in Microsoft Word 333
 encodings, character 32, 47, 59, 328–329
 adding characters from outside 336–337
 and Fetch 359
 Apache servers 331
 common 333
 declaring
 for style sheet 332
 for Web page 59, 330
 default 47
 editing pages with 334
 for style sheets 128, 152, 332
 htaccess files 331
 of RSS feed 379
 saving with proper 333
 enctype attribute, in form element 270
 entities 32, 336
 complete list 419–429
 for (X)HTML symbols 420
 mathematical and technical 425–428
 punctuation 423–424
 shapes and arrows 429
 Western European 420–422
 enumerated values 29
 episodes, of podcast, describing 390
 events, intrinsic
 complete list of 417–418
 defined 314
 examples 324–326
 using 314–315
 ex unit 124, 157
 examples from this book 26
 Excel files, links to 110
 exclamation point. *See* Symbols section
 explicit
 marking podcast as 388
 marking podcast episodes as 391
 Explorer. *See* Internet Explorer
 extensions, file. *See* file extensions
 external style sheets 128, 132

F

Fahrner's Image Replacement 196–197
 favicons 102
 Favorites
 and title 60
 crawler page 368
 icon in 102
 feed, RSS. *See* RSS
 Feedburner.com 394
 FeedDemon 377
 Feeder 379
 Fetch (FTP program for Mac)
 and character encodings 359
 transferring files 358–359
 fieldset element 260
 file extensions
 and opening pages 52
 and viewing Web pages 52
 for RSS feeds 379
 for style sheets 128
 for Web pages 34, 46
 problems with in Windows 47, 348

- file names 34
 - and uppercase letters and symbols 51
 - problems with 349
 - See also* file extensions
- File scheme, in URLs 35
- files
 - compressing 111
 - creating new 45
 - making available for download 110
 - organizing 51
 - transferring to server 356–359
 - uploading 270
 - working with 43–53
- FIR. *See* Fahrner's Image Replacement
- Firefox
 - as aggregator 377
 - cross-browser compatibility 15
 - CSS support 20
- Fireworks. *See* Macromedia Fireworks
- :first-child pseudo-class 143
- :first-letter pseudo-element 144–145
- :first-line pseudo-element 144–145
- fixed positioning 180
- Flash movies, embedding 305–307
 - Google Video and YouTube 306–307
- Flash player 282
 - See also* Macromedia Flash
- Flash Satay 305
- flavors of (X)HTML 40
- Flickr
 - and Creative Commons licenses 86
 - and RSS Media Module, for photo feeds 383, 386
 - finding images 436
- float property 181–182
- floating elements
 - with styles 181–182
 - and width 175
 - with (X)HTML 96
- flow, and CSS 171
- fluid layouts. *See* liquid layouts
- :focus pseudo-class 146
- font-family property 152
 - specifying alternate fonts 153
- font property 159
- font-size property 156
- font-style property 154
- font-variant property 167
- font-weight property 155
- fonts
 - alternate 153
 - and handhelds 203
 - choosing 152, 159
 - common 152
 - generic names 153
 - monospaced 72
 - size 156–157
 - small caps 167
 - variants 167
- footnotes 76
- for attribute, in label element 264
- foreign language Web pages 327–338
- form element 254
- formats
 - choosing appropriate for images 88
 - for images 82–85
 - Plain Text 46
 - Text Document 46
 - text-only 46
 - Web pages 46
 - See also* encodings, character
- formatting
 - aligning with styles 165
 - background color 161
 - blinking text 168
 - bold 70, 155, 159
 - case 166
 - color 160, 161
 - fonts 72, 73, 152, 159
 - indents 163
 - italics 70, 154, 159
 - Kerning 162
 - line height 158, 159
 - link color 146
 - link labels 105
 - logical 70
 - monospaced fonts 72, 73
 - overlining 168
 - physical 70
 - preformatted text 73
 - size 71, 156–157, 159
 - small caps 167
 - strike out 77
 - subscripts 76
 - superscripts 76
 - testing 346
 - text 69–79, 151–168
 - tracking 162
 - underlining 77, 168
 - white space 164
 - See also* page layout
- forms 253–280
 - active images 276
 - and tables 255
 - bullets 263
 - buttons 272
 - with image 273
 - centering 79
 - checkboxes 268
 - default values 274
 - disabling elements 279
 - free-form text 262
 - grouping options 267
 - hidden fields 271
 - keeping elements from being changed 280
 - keyboard shortcuts 278
 - labeling areas 260
 - labeling parts 264
 - menus 266
 - organizing elements 260
 - password boxes 263
 - PHP scripts 256–259
 - processing 256–259
 - radio buttons 265
 - reset button 274–275
 - sending data via email 258–259
 - structure of 254
 - submit buttons 272–273
 - submitting with image 276
 - tab order 277
 - text areas 269
 - text blocks 269
 - text boxes 262
 - uploading files 270

forward slash. *See* Symbols section
 foundation of Web page, creating 58
 frame attribute, in table element 249
 frames
 and search engines 372
 See also Web site
 frameset flavor of (X)HTML 21, 40
 free-form text, in forms 262
 FTP
 in URLs 35
 links to 110
 links to private 111
 transferring files to server 356–359
 Macintosh 358–359
 Windows 356–357
 Fusion. *See* NetObjects Fusion

G

generated HTML pages, and base element 109
 generator element in RSS 381
 generator, specifying for Web page 365
 get method 255
 GIF images
 about 82–85
 disadvantages 84
 for favicons 102
 in forms 276
 LZW compression 84
 glossaries 223
 GoLive. *See* Adobe GoLive
 Google
 and keywords in meta element 363
 archived pages 367
 directory listing 44
 finding images 86, 436
 finding links to your site 371
 getting listed in 370
 Google Video 306–307
 membership in W3C 17
 restricting search according to usage rights 86
 Sitemap Generator 369
 googlebombing 371
 Gopher, links to 111
 GraphicConverter 87, 436
 graphics. *See* images
 greater than symbol. *See* Symbols section
 Greek letters 425–428
 Griffiths, Patrick 195
 grouping options, in forms 267
 grouping selectors 148
 guid element in RSS 382

H

handhelds
 adjusting style sheets for 208
 creating style sheet for 204
 extra links 207
 hiding elements 205
 image headers 206
 mobilize vs. miniaturize 200
 optimizing pages for 170
 points to consider 202–203
 simulators 201
 style sheets for 199–208
 testing site on 201

hanging indents 163
 hash symbol (#). *See* Symbols section
 head element 58
 HTML vs. XHTML 38
 header cells
 aligning 247
 and column groups 247
 description 229
 headers 61
 aligning 61
 and anchors 61
 and keywords 362
 and structure 170
 centering 79
 from images 206
 height attribute
 in img element 93, 94
 in object element 286–291, 302, 308
 in table elements 233
 height property 174, 193
 helper applications, description 33, 282–283
 helping visitors find page 363
 hexadecimal
 and colors 126, 160
 and special symbols 336
 description 431
 table 432
 Hickson, Ian 22
 hidden fields, adding to form 271
 hiding
 directory's contents 49
 elements from handhelds 205
 scripts 318, 319
 sounds 301
 style sheets
 from buggy browsers 132
 from XML parsers 131
 text 67
 hierarchy, of page 61
 History lists, and title 60
 hn element 61
 Holzschlag, Molly 197
 horizontal rules 101
 host, finding 354
 :hover pseudo-class 146
 hr element 101
 href attribute
 definition 104
 in a element 104
 in area element 118
 in base element 109
 in link element 102, 129
 hspace attribute
 in img element 99
 htaccess files
 and character encodings 331
 HTML 14
 and CSS 18
 closing tags 28
 identifying 56
 quotation marks 29
 similarities with XHTML 23
 See also XHTML, (X)HTML, and
 HTML vs. XHTML

- html element
 - and specifying languages 74
 - HTML vs. XHTML 38, 57
 - starting pages 56
- HTML vs. XHTML 38, 38–39
 - deciding between 21–22, 56
 - DOCTYPE declaration 57
 - head and body elements 58
 - html element 38, 57
- HTML-Kit for Windows 45, 435
- HTTP (Hypertext Transfer Protocol) 35
- http-equiv attribute, in meta element 310, 320, 330
- Hunt, Lachlan 291
- hypertext, definition 14
- i element 70
- .ico format 102
- icons 102
- id attribute 63
 - and anchors 63, 106
 - and form labels 264
 - and JavaScript 63
 - assigning to element 63
 - choosing by selector 140
 - in div element 64
 - in headers 61
 - in map element 117
 - in span element 65
- ideas from the Web 53
- image element in RSS 381
- image maps
 - and search engines 372
 - client-side 117
- image replacement 196–197
- Image Size command 95
- images 81–102
 - active (in forms) 276
 - adding to podcast 388
 - aligning 100
 - alternate text 91
 - and getting visitors 362
 - background 172–173
 - BMP format 86, 349
 - borders 90, 184–185
 - in handhelds 208
 - centering 79, 100
 - changing size of 94
 - changing when visitor points 324–326
 - clickable 116
 - clip art for Web pages 436
 - color 82
 - compression 84
 - cropping 95
 - display speed 90
 - dividing into clickable regions 116
 - favicons 102
 - floating 96
 - for headers 206
 - for labeling links 114
 - for submitting data 276
 - format 82
 - free use 86, 436
 - getting 86
 - GIF format 82
 - hiding tooltips 91
 - important factors 81–85
 - in Address Bar 102
 - in reset button 275
 - in submit buttons 273
 - in tables 240–241
 - inline 82
 - JPEG format 82
 - labeling with alternate text 91
 - loading into cache 326
 - making smaller 95
 - maps 117–118
 - placing on page 90
 - PNG format 82
 - pop-ups 193
 - positioning vertically with styles 188
 - problems with 349
 - replacement 196–197
 - resolution 83
 - rollovers 193, 324–326
 - saving 88–89
 - scaling 94
 - showing series automatically 310
 - size 83, 92–95, 174
 - figuring out 92
 - space around 99
 - speed of display 84, 90, 92
 - stopping text wrap 98
 - with styles 182
 - testing 346
 - that don't appear 349
 - thumbnails 115
 - tooltips 91
 - transparency 85
 - using 81–102
 - wrapping text around 96
 - with styles 181–182
 - wrapping text between 97
- img element 90
- !important value, in CSS 123
- importing style sheets 132
- inches, in CSS 125
- indents 163
- index.html 49
- inherit value, in CSS 122
- inheritance 122–123
 - and computed value 157, 158, 162, 163
 - and width 175
- inline elements, description 30, 171
- input element
 - for active images 276
 - for checkboxes 268
 - for hidden fields 271
 - for password boxes 263
 - for radio buttons 265
 - for reset button 274
 - for submit button 272
 - for text boxes 262
 - for uploading files 270
- ins element 77
- inserted text 77
- inspiration of others 53, 136

Internet Explorer
 5.x for Windows, width bug 174–175
 in tables 233
 6 for Windows
 standards vs. quirks mode 41
 XML declaration triggers quirks mode bug 57
 7 for Windows
 as aggregator 377
 browser wars 16
 bugs
 background and legend element 261
 borders 185
 borders in tables 231
 pseudo-classes and a element 146
 : hover pseudo-class 195
 conditional comments 290–291
 cross-browser compatibility 15
 CSS support 20
 problems with 350
 Internet Service Provider. *See* ISP
 intrinsic events
 complete list of 417–418
 declaring scripting language 320
 defined 314
 using 314–315
 Ishida, Richard 327
 ISO 8859-1
 and Fetch 359
 See also encodings, character
 ISP, as Web host 354
 italics 70, 154, 159
 and handhelds 203
 with address 70
 item element in RSS 382
 items, adding to RSS feed 382–391
 itpc, protocol, for subscribing to podcasts in iTunes 395
 iTunes 377
 Artist column 390
 creating links to 111
 description 375
 Description column 390
 preparing podcasts for 386–391
 Release Date column 391
 submitting podcast to 393
 subscribing to podcasts with 397
 itunes:author element in RSS 388, 390
 itunes:block element in RSS 389, 391
 itunes:category element in RSS 388
 itunes:duration element in RSS 390
 itunes:email element in RSS 387
 itunes:explicit element in RSS 388, 391
 itunes:image element in RSS 388
 itunes:keywords element in RSS 388, 390
 itunes:name element in RSS 387
 itunes:owner element in RSS 387
 itunes:subtitle element in RSS 390
 itunes:summary element in RSS 388

J

Jasc Software 87, 436
 Java, inserting applets 308

JavaScript
 adding date and time to page 322
 and id 63
 controlling Window size 323
 description 311
 disabled 317
 embedding multimedia with 292–293
 enabling disabled form elements 279
 loading images into cache 326
 rollovers 324–326
 scripts for your use 321–326
 switching style sheets 130
 using in page 311–326
 JAWS, screen reader 91, 197
 jEdit, text editor 435
 JPEG images
 about 82–85
 compression 84
 disadvantages 84
 in forms 276
 saving 88–89
 justifying text 165

K

kbd element 72
 keeping search engines out 366
 kerning 162
 keyboard shortcuts
 for form elements 278
 for links 112
 keywords 362
 adding for search engines 363
 adding to podcast 388
 for podcast episodes 390
 tips for improving ranking 362
 Krug, Steve 44

L

label attribute, in optgroup element 267
 label element 264
 labeling
 form parts 264
 forms 260
 links
 with images 114
 with text 104
 with tooltips 68, 91
 lang attribute 338
 in q and html 74
 language
 specifying for page 338
 specifying for script 312
 language attribute, in script element 312
 language codes 75, 338
 language element in RSS 380
 lastBuildDate element in RSS 380
 layout. *See* page layout
 left property 178, 179, 180
 legend element 260
 Lemke Software 87, 436
 lengths in CSS 124
 less than symbol. *See* Symbols section
 letter-spacing property 162
 li element 216

- line breaks 66
 - creating 66
 - in cells 251
 - maintaining original 73, 164
 - line height 158, 159
 - line-height property 158
 - lines, horizontal. *See* horizontal rules
 - link element 129–130
 - for favicons 102
 - media attribute 133
 - link element in RSS
 - in channel element 380
 - in item element 382
 - :link pseudo-class 146
 - links 14, 103–118
 - and pseudo-classes 146
 - clickable images 117
 - color of 146
 - creating other kinds 110
 - deep 105
 - eliminating image border 114
 - exchanging to get visitors 373
 - formatting labels 105
 - image maps 117
 - images as labels 114
 - keyboard shortcuts 112
 - labeling 105
 - with images 114
 - lowercase letters 104
 - relative URLs 104
 - removing from tab sequence 113
 - removing underlining 168
 - rollovers 324–326
 - RSS feed 394–395
 - setting tab order 113
 - tables of contents 106
 - testing 346
 - to anchors 106, 107
 - to any file 110
 - to external images 115
 - to FTP 110
 - to headers 61
 - to other Web pages 104
 - to private FTP 111
 - to specific sections of page 106
 - to specific windows 108, 323
 - to Yahoo Groups and messages 111
 - to your site, finding 371
 - underlining 168
 - window size 323
 - :hover pseudo-class 146
 - liquid layouts 169, 227
 - and tables 228
 - importance for handhelds 202
 - text size 157
 - list-style-image property 220
 - list-style-position property 221
 - list-style property 222
 - list-style-type property 218
 - lists 215–225
 - and br element 217
 - bullets 218
 - definition 223
 - drop-down menus with 194–195
 - in handhelds 203
 - markers
 - choosing 218
 - custom 220
 - positioning 221
 - nested 224
 - ordered 216
 - setting all styles at once 222
 - unordered 216
 - location, of style sheets 122–123, 135
 - logical formatting 70
 - looking at pages in browser 52
 - loose (X)HTML. *See* transitional (X)HTML
 - lowercase letters
 - and file names 34
 - and links 104
 - with styles 166
 - LView Pro 436
 - Lynx browser 72
 - LZW compression 84
- ## M
- Macintosh
 - and file extensions 34
 - default character encoding 59
 - transferring files to server 358–359
 - Macromedia Dreamweaver 45, 435
 - Macromedia Fireworks 436
 - and PNG images 89
 - Macromedia Flash 436
 - mailto URLs 35
 - managingEditor element in RSS feed 380
 - map element 117
 - margin property 176
 - and centering 234
 - margins 176
 - and width 175
 - in handhelds 202, 208
 - in print style sheets 211
 - markers
 - custom 220
 - in lists 218
 - positioning 221
 - setting all properties at once 222
 - marking changed text 77
 - markup, definition 28
 - mathematical symbols 425–428
 - max-height property 174
 - max-width property 174
 - maxlength attribute, in input element
 - for password boxes 263
 - for text boxes 262
 - McLellan, Drew 305
 - media attribute
 - for handheld style sheets 204
 - for print style sheets 210
 - in link and style elements 133
 - menus 266
 - drop-down 194–195
 - meta element
 - character encodings 59, 330
 - for adding extra information 365
 - for search engines 363, 364, 366
 - for setting default scripting language 320
 - for slide shows 310
 - method attribute, in form element 254
 - Meyer, Eric 193, 225

Microsoft Corporation, membership in W3C 17
 Microsoft Word 45
 and saving (X)HTML 47
 and Web pages 48
 editing with proper encoding 335
 Save as HTML command 348
 millimeters, in CSS 125
 MIME types
 Flash player files 305
 for favicons 102
 for MP3 files 300
 list of 385
 of enclosures 384
 QuickTime movies 288
 Windows Media Player files 304
 min-height property 174
 min-width property 174
 miniatures 115
 Mobile Web, description 199
 See also handhelds
 mobilizing vs. miniaturizing Web pages for
 handhelds 200
 monitors
 adapting designs to 169, 228
 and resolution 83, 124
 monospaced fonts 72
 and preformatted text 73
 mouseovers 324–326
 Mozilla Corporation, member of W3C 17
 MP3s, embedding 300–301
 multimedia 281–310
 adding to RSS feed 384–385
 applets 308
 background sounds 301
 embedding other files 309
 file size 281
 Flash movies 305–307
 Google Video and YouTube 306–307
 getting files 284
 Google Video and YouTube 306–307
 linking to 285
 plugins and players 282–283
 QuickTime movies 286–299
 for all 290–291
 for Windows 286–287
 standards-compliant way 288–289
 with JavaScript 292–293
 slide shows 310
 Windows Media Player files 302–304
 standards-compliant way 304
 See also podcasts
 multiple attribute, in select element 266

N

name attribute
 in a element 106
 in button element
 for reset button 275
 for scripts 316
 for submit button 273
 in input element
 for active images 276
 for checkboxes 268
 for hidden fields 271
 for menus 266
 for radio buttons 265

 for text areas 269
 for text boxes 262, 263
 for uploading files 270
 omitting for reset button 274
 omitting for submit button 272
 in map element 117
 in meta element 363, 364, 365, 366
 namespace
 adding iTunes to RSS feed 386
 declaring in html element 56–57
 RSS Media Module (for Flickr photo feeds) 386
 naming elements 63
 navigational buttons for links 114
 navigational drop-down menus 194–195
 nesting
 elements 31
 lists 224
 tables 236–237
 NetNewsWire 377, 396
 NetObjects Fusion 435
 Netscape 4.x, and @import rules 132
 Netscape 6
 CSS support 20
 standards vs. quirks mode 41
 switching style sheets 130
 Netscape Communications
 absorbed by AOL 17
 and browser wars 16
 Netscape Communicator
 problems with 350
 new files, creating 45
 new windows, controlling size 323
 newsgroups, and publicizing site 373
 nohref attribute, in area element 118
 Nokia, testing pages for handhelds 201
 non-breaking spaces 62
 in table cells 237, 251
 See also in Symbols section
 non-empty elements 28
 HTML vs. XHTML 38
 non-standard elements. *See* proprietary elements
 non-structural column groups 246–247
 noscript element 317
 noshade attribute, in hr element 101
 nowrap attribute, in td or th elements 251
 number symbol (#). *See* Symbols section
 numbers, in CSS 125
 numerals, in ordered lists 218
 numeric references 336

O

object element
 Flash movies 305–307
 Java applets 308
 MP3 audio 300–301
 other kinds of files 309
 QuickTime movies 286–291
 vs embed element 282
 Windows Media Player files 302–304
 ol element 216
 onblur attribute 315
 onchange attribute 315
 onclick attribute 314
 ondblclick attribute 314
 onfocus attribute 315
 onkeydown attribute 315

onkeypress attribute 315
 onkeyup attribute 315
 onload attribute, in body elements 314
 onmousedown attribute 314
 onmousemove attribute 314
 onmouseout attribute 314
 onmouseover attribute 314
 onmouseup attribute 314
 onreset attribute 315
 onselect attribute 314
 onsubmit attribute 315
 onunload attribute, in body element 314
 Open With, command 50
 opening pages
 in browser 52
 in new window 108
 Openwave, testing pages for handhelds 201
 Opera
 and keyboard shortcuts 278
 as aggregator 377
 cross-browser compatibility 15
 CSS support 20
 no quirks mode 41
 Small Screen command 201
 testing pages for handhelds 201
 Opera for Mobile 201
 optgroup element 267
 option element 266
 ordered lists 216
 organizing
 files 51
 forms 260
 pages 61
 site 44
 tables 228
 with divisions 64
 with inline spans 65
 orphans 214
 See also page-break properties
 overflow, controlling 187
 overflow property 187
 overlining text 168
 Oxton, John 17

P

p element 62
 padding 177
 cells 242–243
 in handhelds 208
 padding property 177
 for handhelds 202
 in tables 242–243
 page-break properties 212
 page breaks 212
 page layout 169–188, 227–245
 3D positioning 183
 absolute positioning 179
 aligning text 165
 backgrounds 172–173
 borders 184–185
 centering 79, 165, 175, 234
 clearing floats 182
 color 172–173
 cursors 186
 fixed positioning 180
 floating elements 181–182
 for handhelds 199–208
 height 174
 line breaks 164
 margins 176
 and width 175
 overflow 187
 padding 177
 relative positioning 178
 vertical alignment 188
 width 174–175
 and floating elements 181
 with styles 169–188
 with tables 227–245
 Page Source, command 53
 Paint Shop Pro for Windows 87, 436
 paragraphs
 aligning 62
 centering 62, 79
 space between 62, 176, 177
 starting new 62
 param element 286–291, 294, 295, 296, 297, 299,
 301, 302, 308
 parentheses, in URLs in CSS 125
 parents and children 31, 171
 and height and width 174
 and inheritance 122
 selectors 142
 parsers, XML 19
 password boxes 263
 paths, in URLs 36
 percentages
 and font size 157
 and line height 158
 and spacing 162
 and text indents 163
 importance for handhelds 202
 in CSS 125
 period. *See* Symbols section 140
 Perl/CGI scripts
 and base element 109
 and forms 253
 Photoshop. *See* Adobe Photoshop and Photoshop
 Elements
 PHP, and forms 253, 256–257
*PHP for the World Wide Web: Visual QuickStart
 Guide, Second Edition* 256
 physical formatting 70
 picas, in CSS 125
 pipe symbol. *See* Symbols section
 pixel shims, in table cells 233
 pixels
 and print style sheets 211
 font size 156
 in CSS 124
 in (X)HTML 29
 size vs resolution 83
 placing images on page 90
 Plain Text, format 46, 48
 See also text-only format
 players 282–283
 getting for visitors 283
 plugins 33, 282–283
 plus sign. *See* Symbols section

- PNG format
 - about 82–85
 - compression 84, 85
 - disadvantages 85
 - for favicons 102
 - podcasts 375–397
 - adding image to 388
 - adding information about 388–389
 - categories, official list 389
 - creating direct links to 395
 - describing individual episodes 390
 - description 375
 - links to subscribe 395
 - preparation for 378
 - preparing for iTunes 386–391
 - submitting to iTunes 393
 - podcatchers, description 377
 - pointer, changing with CSS 186
 - points
 - in CSS 125
 - in print style sheets 211
 - pop-ups, creating with CSS 193
 - portfolio, showing automatically 310
 - position property
 - with absolute value 179
 - with fixed value 180
 - with relative value 178
 - positioning 171, 178–183
 - in 3D 183
 - markers 221, 222
 - post method 254, 270
 - poster movies, in QuickTime 297
 - pound symbol (#). *See* Symbols section
 - pre element 73
 - predefined colors 126
 - predefined values 29
 - case of 38
 - in CSS, and quotation marks 124
 - preformatted text 73
 - preview movies in QuickTime 297
 - Print Preview, command 211
 - print style sheets 209–214
 - page breaks 212
 - printing link URLs 213
 - specifying 133
 - suggestions for 211
 - privacy, maintaining 366
 - problem solving 339–351
 - borders 185
 - browser displays code 348
 - browser incompatibilities 350
 - CSS checklist 344
 - first steps 341
 - getting help 351
 - handhelds 201
 - HTML checklist 342
 - images that don't appear 349
 - style sheets 121
 - techniques for 340
 - validating (X)HTML and CSS code 345
 - validating RSS feeds 392
 - XHTML checklist 343
 - processing forms 256–257
 - via email 258–259
 - programs for writing (X)HTML 434
 - properties (in CSS)
 - complete list 409–416
 - definition 119
 - possible values for 124–126
 - proprietary elements, and DOCTYPE declarations 57
 - pseudo-classes 138
 - definition 146
 - pop-ups 193
 - rollover buttons 192
 - :active 146
 - :first-child 143
 - :focus 146
 - :hover 146, 195
 - :link 146
 - :visited 146
 - pseudo-elements 138
 - :first-letter 144–145
 - :first-line 144–145
 - pubDate element in RSS 380
 - in item element 382
 - publicizing your site. *See* Web pages, getting visitors
 - publishing 353–359
 - RSS feed 394–395
 - punctuation symbols 423–424
- Q**
- q element 74
 - Question and Answer Forum 26, 351
 - QuickTime audio. *See* MP3s, embedding
 - QuickTime movies
 - adding attributes to secondary movies 298
 - and background play 293
 - embedding 286–299
 - for Windows 286–287
 - standards-compliant way 288–289
 - with JavaScript 292–293
 - looping 296
 - parameters for 289
 - poster movies 297
 - previews 297
 - scaling 294
 - sequels 299
 - URL extensions 298
 - QuickTime Player 282
 - quirks vs standard mode
 - IE 6 for Windows with XML declaration 57
 - quirks vs. standard mode 41
 - quotation marks
 - and font names 152
 - and scripts 315
 - and slide shows 310
 - and URLs in CSS 125, 132
 - and :first-letter pseudo-element 145
 - generated for q element 75
 - HTML vs. XHTML 38
 - predefined values in CSS 124
 - quoting text 74–75

R

radio buttons 265
 ranking, improving in search results 362, 363, 371
 rating element in RSS 381
 readonly attribute, in form elements 280
 red x (broken image icon) 349
 references, character 32, 336
 refreshing pages 310
 rel attribute, in link element 102, 129, 130
 relative positioning 178
 See also page layout
 relative sizes, of text 71
 relative units, in CSS 124
 relative URLs 37
 and base element 109
 in CSS 125, 129
 See also URLs
 Release Date column, in iTunes 391
 replaced elements, description 175
 reset button 274–275
 with image 275
 resetting forms 274
 resolution
 font size 156
 images 83, 95
 monitor 83, 124
 returns (carriage)
 and preformatted text 73
 controlling with style sheets 164
 right property 178, 179, 180
 robots 366
 rollovers 324–326
 buttons 192
 loading images into cache 326
 rows
 changing color 240–241
 creating new 229
 spanning cells across 245
 rows attribute, in textarea element 269
 rowspan attribute, in td element 245
 RSS
 adding enclosure 384–385
 adding items 382–391
 adding namespaces 386
 and XSLT 394
 categories 381
 in iTunes 388
 character encoding 379
 describing site with 380
 description 376–377
 formatting dates properly 381
 generators 379
 in Firefox 377
 in IE 7 377
 in Opera 377
 logo 394
 preparing for iTunes 386–391
 publishing feed 394–395
 specification 381
 starting a feed 379
 subscribing to feed 396
 validating feeds 392
 rss element in RSS 379
 RSS Media Module 383

rules

 horizontal 101
 in tables 250

rules attribute, in table element 250

S

Safari 15
 samp element 72
 Save as HTML command, in Word 47, 348
 Save as Web Page command, in Word 47, 48, 348
 Save For Web command, in Photoshop 95
 Save for Web command, in Photoshop 88–89
 saving
 images 88–89
 other people's Web pages 53
 Web pages 46
 character encoding 59
 scaling images 94
 scanners 86
 scheme, in URL 35
 screen size of handhelds 202
 screen style sheets 133, 204
 See also handhelds
 script element 312
 scripts 256–259, 311–326
 adding alternate information 317
 adding “automatic” 312
 and quotation marks 315
 and search engines 372
 calling external automatic 313
 controlling window size 323
 for processing forms 256–257
 for sending form data via email 258–259
 hiding from older browsers 318
 hiding from XML parsers 319
 JavaScript for your use 321–326
 loading images into cache 326
 location on page 312
 rollovers 324–326
 setting default language 320
 triggering with intrinsic event 314
 with buttons 316
 See also JavaScript, PHP
 search engines
 adding information for 363
 controlling summary 364
 improving ranking
 with description 364
 with keywords 362–363
 with links 371
 with title 60, 362
 keeping out 366
 keeping pages from being archived 367
 optimizing design for 170, 372
 specialized 370
 submitting your site to 370
searchenginewatch.com 361
 select element 266
 selected attribute, in option element 266
 selectors 137–149
 :active pseudo-class 146
 adjacent sibling 143
 attributes 147
 by class or id 140
 by context 141–143

- by name 139
- child selector 142
- combining 149
- definition 120
- descendant 141
- description 138
- :first-child pseudo-class 143
- :focus pseudo-class 146
- grouping 148
- :hover pseudo-class 146
- :link pseudo-class 146
- link states 146
- :visited pseudo-class 146
- wild card selector 139
- :first-line and :first-letter 144–145
- semicolons, and style rules 120
- sequels, in QuickTime movies 299
- server name in URL 35, 355
- shading, of horizontal rules 101
- shape attribute, in area element 117
- shapes, entities for 429
- sharp symbol (#). *See* Symbols section
- Shea, Dave 44, 197
- Shockwave player 282
- sidebars, hiding in print style sheets 211
- simulators for handhelds 201
- site, designing 44
- site map
 - creating 368–369
 - description 368
 - submitting 370
- Sitemap Generator, Google 369
- sitemap protocol 369
- size
 - and viewing speed 92–95
 - of cells in table 232–233
 - of images 83, 92–95
 - changing 94
 - figuring out 92
 - reducing 95
 - specifying 94
 - of multimedia files 281
 - of new windows 323
 - of pages, with tables 228
 - of tables 232–233
 - of text 71, 156–157
- size attribute
 - in input element
 - for password boxes 263
 - for text boxes 262
 - for uploading files 270
- size attribute
 - in hr element 101
 - in select element 266
- skipDays element in RSS 381
- skipHours element in RSS 381
- slide shows 310
- small caps 159, 167
- small element 71
- Small Screen command, in Opera 201
- Small Screen Rendering technology, in Opera 201
- sounds
 - background 301
 - embedding 300, 301
 - getting 284
- Source, command 53
- source element in RSS 383
- spaces
 - and preformatted text 73
 - in lengths 124
 - in selectors 141
 - in Web pages 32
 - maintaining original 164
- spacing
 - and sub/superscripted characters 76
 - around elements 176–177
 - around images 99
 - between lines and paragraphs 66
 - between paragraphs 62, 66
 - line breaks 66
 - maintaining original 73
 - of text (with styles) 162
- spamming 362
- span attribute
 - in col element 246–247
 - in colgroup element 246–247
- span element 65
- spanning cells
 - across columns 244
 - across rows 245
- special techniques
 - automatic page navigating 310
 - dynamic effects with CSS 189–197
 - JavaScript for your use 321–326
- specificity 122–123
- speed
 - and images 84, 90, 92–95
 - loading images into cache 326
 - of table display 252
 - and nested tables 236
- spiders 366
- src attribute
 - in img element 90
 - in input element 276
 - in script element 313
- standards
 - browser wars 16
 - importance of 17, 39
- standards vs. quirks mode 41
 - and URLs in DOCTYPE declarations 57
- start attribute, in ol element 219
- starting
 - new lines 66
 - new paragraph 62
 - Web pages 56
- Stearns, Geoff 293
- stopping floating 98, 182
- strict flavor of (X)HTML 21, 40
- strict mode. *See* standards mode
- strike out formatting
 - with (X)HTML 77
 - with styles 168
- strong element 70
- structure
 - importance of 42, 170
 - of tables 228
 - of (X)HTML 31
- style attribute
 - in any tag 134
 - in button element 316
- style element 131
 - media attribute 133

- style rules
 - and semicolons 120
 - constructing 120
 - definition 120
 - See also* style sheets, styles, CSS
- style sheets 119–214
 - adjusting for handhelds 208
 - alternate 130
 - and URLs 129
 - applying locally 134
 - browser problems 350
 - building blocks 119–126
 - cascade 122–123
 - character encoding 332
 - comments 121
 - creating for handheld 204
 - description 119
 - dynamic effects 189–197
 - external 128
 - file extension 128
 - importing 132
 - linking 129
 - file extension 128
 - for handhelds 199–208
 - for print 209–214
 - formatting text 151–168
 - hiding 131
 - hiding from buggy browsers 132
 - inheritance 122–123
 - internal 131
 - location, importance of 122–123, 135
 - media specific 133
 - selectors 137–149
 - specificity 122–123
 - switching 130
 - testing and debugging 121
 - text-only format 128
 - viewing other people's 136
 - working with files 127–136
 - See also* styles, CSS
- styles 119–214
 - aligning
 - text 165
 - vertically 188
 - background 161, 172–173
 - blinking text 168
 - bold formatting 155, 159
 - borders of elements 184–185
 - building blocks 119–126
 - case (upper/lower) 166
 - changing all font values 159
 - child element 171
 - choosing font 159
 - clear floating 182
 - color 160, 161, 172–173
 - comments 121
 - competing 122
 - cursors 186
 - defaults 159
 - displaying and hiding elements 190–191
 - drop-down menus 194–195
 - floating elements 181–182
 - font family 152, 159
 - font size 156–157, 159
 - formatting text 151–168
 - height of elements 174
 - indents 163
 - italics 154, 159
 - kerning 162
 - layout 169–188
 - line height 158, 159
 - list markers
 - all at once 222
 - choosing 218
 - custom 220
 - positioning 221
 - margins 176
 - and width 175
 - nested lists 224
 - offsetting elements within flow 178
 - overflow 187
 - overlining text 168
 - padding 177
 - page layout 169–188
 - parent element 171
 - positioning elements 178–183
 - absolutely 179
 - fixed 180
 - in 3D 183
 - relatively 178
 - vertically 188
 - properties and values 409–416
 - selectors 137–149
 - small cap fonts 167
 - stopping floating 182
 - strike out text 168
 - tracking 162
 - underlining text 168
 - unexpected results 159
 - white space 164
 - width 174–175
 - and floating elements 181
 - word spacing 162
 - See also* CSS, style sheets
- sub element 76
- submit buttons 272
 - creating 272–273
 - multiple 272
 - vs. active images 276
 - with image 273
- subscribing
 - to podcasts with iTunes 397
 - to RSS feed 396
- subscripts 76
- Sullivan, Danny 361
- sup element 76
- superscripts 76
- support, technical 351
- SWFObject (for embedding multimedia) 293
- switching style sheets 130
- symbols 327–338
 - adding 336–337
 - entities for 419–429
 - for (X)HTML 420
 - in file names 51
 - in titles 60
 - in Web pages 32
 - mathematical and technical 425–428
 - punctuation 423–424
 - shapes and arrows 429
 - Western European 420–422

syndication 375–397
 feeds 376–377
 preparing for 378
 See also RSS, podcasts
 syntax checkers 45

T

tab order
 for forms 277
 for links 113
 tabIndex attribute
 in a element 113
 in form elements 277
 table element 229
 table-layout property 252
 tables 227–252
 aligning cell's contents 238–239
 and forms 255
 background image 240–241
 borders 230–231
 displaying external 249
 displaying internal 250
 cell spacing and padding 242–243
 cell width 232–233
 centering 234
 changing color 240–241
 column groups 246–247
 columns 244–247
 combining 236–237
 creating 229
 empty cells with ` ` 237
 external borders 249
 height 232–233
 horizontal sections 248
 IE 5.x for Windows width bug 233
 internal borders 248, 250
 line breaks 251
 liquid layout 228
 mapping out 228
 nesting 236–237
 page layout 227–245
 pixel shims 233
 rows 245
 simple 229
 size of 228, 232–233
 spanning cells
 across columns 244
 across rows 245
 speeding up display 252
 structure 228
 width 232–233
 wrapping text around 235
 tables of contents, and anchors 106
 target attribute
 in a element
 for windows 108
 target attribute
 in area element 118
 in base element 109
 targets
 setting default 109
 to specific windows 108
 tbody element 248
 td element 229

technical support 351
 technical symbols 425–428
 Telegraphics, plugin for .ico format 102
 telnet, links to 111
 templates, (X)HTML 56
 testing Web pages 52, 346
 See also problem solving
 text
 aligning 61, 62, 165
 alternate for images 91
 blinking 168
 bold 70
 case, with styles 166
 centering 61, 62, 165
 color 160
 font 152, 159
 hiding (comments) 67
 in monospaced fonts 72
 in Web pages 32
 italics 70
 justifying 165
 line breaks 66
 line height 158
 marking changed 77
 multicolumn 244–245
 overlining 168
 preformatted 73
 quoting 74–75
 size 71, 156–157, 159
 in handhelds 208
 in print style sheets 211
 stopping wrap 98
 strike out 77
 subscripts 76
 superscripts 76
 underlining 77, 168
 wrapping
 around images 96
 around tables 235
 between images 97
 text-align property 165
 for table cells 238–239
 text areas 269
 text boxes 262
 text-decoration property 168
 Text Document, format 46
 See also text-only format
 text editors 434
 text-indent property 163
 Text Size command 157
 text-transform property 166
 textarea element 269
 TextEdit 45, 434
 textInput element in RSS 381
 text-only format 46, 348
 and external scripts 313
 and HTML 14
 and opening pages 52
 and style sheets 128
 tfoot element 248
 th element 229
 thead element 248
 3D positioning 183
 thumbnails 115

time, adding to page with JavaScript 322

title attribute

- in abbr and acronym elements 78
- in any tag 68
- in link element 130
- to hide tooltips for images 91

title element 60

title element in RSS

- in channel element 380
- in item element 382

titles

- and search engines 60
- creating 60
- improving ranking with 362
- keywords 362

tools, (X)HTML 433–436

tooltips

- for any element 68
- for images 91
- hiding 91

top property 178, 179, 180

tr element 229

tracking 162

traffic, increasing. *See* Web pages, getting visitors

transferring files 356–359

- Macintosh 358–359
- Windows 356–357

transforming text. *See* case (upper/lower), changing with styles

transitional flavor of (X)HTML 21, 40

- DOCTYPE declarations 56

Translate ISO characters, option in Fetch 359

transparency 85

tt element 72

ttd element in RSS 381

type attribute

- in button element
 - for reset button 275
 - for scripts 316
 - for submit button 273
- in input element
 - for active images 276
 - for checkboxes 268
 - for password boxes 263
 - for radio buttons 265
 - for reset button 274
 - for submit button 272
 - for text boxes 262
 - for uploading files 270
- in ol tag
 - proper nesting order 225

type attribute

- in link element 102, 129
- in ol and ul elements 218
- in script element 312
- in style element 131

U

ul element 216

Ullman, Larry 256

underlining 168

- inserted text 77
- removing 168

Unicode 32, 328

Unicode Resources site (Alan Wood) 329

Uniform Resource Locator. *See* URLs

units

- in CSS 124, 125
- in HTML 29

universality 14

unordered lists 216

uploading files, with forms 270

uppercase letters

- in file names 51
- with styles 166

URL extensions (in QuickTime) 298

URLs 35–37

- absolute vs. relative 36
- and DOCTYPE declarations 57
- and file names 34
- creating your own 355
- definition 35
- formatting as monospaced font 72
- in CSS 125, 132
- in external style sheets 129
- in links 104
- in printed output 213
- problems with 349
- relative 37

usemap attribute, in image element 118

Usenet news groups, links to 111

UTF-16, and XHTML 59

UTF-8 32, 329

- and XHTML 59

See also encodings, character

V

validating

- and DOCTYPE declarations 57
- RSS feeds 392
- (X)HTML and CSS 345

valign attribute, in table elements 238–239

value attribute

- in button element
 - for reset button 275
 - for submit button 273
- in input element
 - and active images 276
 - for checkboxes 268
 - for hidden fields 271
 - for radio buttons 265
 - for reset button 274
 - for submit button 272
 - for text boxes 262

value attribute

- and textarea element 269
- in li element 219
- in option element 266

values

- CSS list 409–416
- in CSS, definition 119
- predefined 29
- selectors based on 147
- (X)HTML 29

var element 70

Verizon, free Web hosting services for members 354

versions of (X)HTML 40, 56

vertical-align property 188

- for table cells 238–239

vi (Unix program) 434
 video
 embedding QuickTime 286–299
 for Windows 286–287
 standards-compliant way 288–289
 with JavaScript 292–293
 embedding Windows Media Player 302–304
 standards-compliant way 304
 getting 284
 Google and YouTube 306–307
 View Source, command 53
 viewing
 (X)HTML code 53
 other designers' CSS 136
 Web pages 15, 52
 visibility property 190
 visited links, color of 146
 :visited pseudo-class 146
 vspace attribute
 in img element 99

W

W3C 17
 International section 329
 validators 345
 WAI. *See* Web Accessibility Initiative
 WAIS servers, links to 111
 WAP (Wireless Application Protocol) 199
 WaSP. *See* Web Standards Project, The
 Web Accessibility Initiative 39
 Web design, books on 44
 Web host, finding 354
 Web pages
 adding author, generator, copyright information 365
 character encoding 59, 328–329
 creating new 45
 default page 49
 describing for search engines 364
 description of 13
 designing site 44
 editing 50
 file extensions 46
 file names 34
 foreign languages 327–338
 format 46
 foundation 58
 getting visitors 361–373
 by getting linked 371
 hierarchy 61
 home page 49
 hosting 354
 identifying 56
 inserting images 90
 keeping private 366
 non-text content 33
 on handhelds 199–208
 opening in browser 52
 opening in new window 108
 organizing 61
 printing 209–214
 problems. *See* problem solving
 programs for writing 434
 publishing 353–359
 saving 46
 starting 56
 structure 170
 testing 52, 346
 text content 32
 transferring to server 356–359
 typical width 228
 validating 345
 viewing 15
 Web rings 371, 373
 Web site for this book 26
 Web Standards Project, The 16, 39, 350
 webMaster element in RSS 380
 WebMonkey 361
 weights, of fonts in styles 155
 Western European characters and symbols 420–422
 white-space property 164
 white space. *See also* spaces and spacing
 widows property 214
 See also page-break properties
 width 174–175
 and floating elements 181
 of cells in table 232–233
 of tables 232–233
 width attribute
 in hr element 101
 in img element 93, 94
 in object element 286–291, 302, 308
 in td or table elements 232–233
 width property 174–175, 193
 and floating elements 181
 in tables 232–233
 Windows
 and file extensions 34
 problems 348
 default character encoding 59
 transferring files to server 356–357
 windows
 controlling size 323
 opening new 108
 typical size 228
 See also monitors
 Windows Media Player 282
 embedding 302–304
 standards-compliant way 304
 windows-1252 59
 See also encodings, character 59
 WML (Wireless Markup Language) 199
 See also Web site
 Wood, Alan 327
 Word. *See* Microsoft Word
 WordPad 45, 434
 word-spacing property 162
 World Wide Web 14
 World Wide Web Consortium. *See* W3C
 wrapping text
 around
 anything 181
 images 96
 tables 235
 between images 97
 stopping around images 98, 182
 WS_FTP (FTP program for Windows) 356–357
 WYSIWYG display 434

X

- x, red (broken image icon) 349
- (X)HTML
 - abbreviation, meaning of 23, 27
 - and Mobile Web 199
 - building blocks 27–42
 - code displayed 348
 - copying other's 53
 - creating new pages 45
 - default display 42
 - displaying code on Web page 337
 - editors 434
 - elements and attributes 28–29
 - complete list of 399–408
 - entities
 - complete list 419–429
 - for (X)HTML 420
 - mathematical and technical 425–428
 - punctuation 423–424
 - shapes and arrows 429
 - Western European 420–422
 - for handhelds 202–203
 - formatting 69–79
 - frameset flavor 21
 - in RSS feed 383
 - specifying version 56
 - strict flavor 21
 - structural elements 55–68
 - structure 31
 - tools 433–436
 - transitional flavor 21
 - UTF-8 329
 - values 29
 - versions, flavors, and DOCTYPE 40
 - viewing other people's 53
 - writing with Microsoft Word 48
 - See also* HTML vs. XHTML
- XHTML
 - advantages of 39
 - and & (ampersand) 32, 337
 - and character encodings 59
 - and CSS 39
 - and required namespace declaration 56–57
 - attribute values 29
 - CDATA sections 131, 319
 - closing tags 28
 - considered dangerous 22
 - hiding scripts and styles 319
 - history 19
 - id attribute in map element 118
 - quotation marks 29
 - similarities with HTML 23
 - tbody element 248
 - See also* HTML, (X)HTML, and HTML vs. XHTML
- x-mac-roman, character encoding 59
- XML 39
 - and RSS 379
 - declaration
 - bug in IE 6 for Windows 57
 - history 19
 - syntax 38
- XML parsers 19
 - and scripts 319
 - and style sheets 131

- xmlns attribute, for declaring XML
 - namespace 56–57
- xml:lang attribute 338
 - in q and html 74
- XSL 128, 131
- XSLT, and RSS feeds 394

Y

- Yahoo
 - and exploring Web 44
 - getting listed in 370
 - links to Groups 111
 - searching for images 436
- YouTube Video 306–307

Z

- z-index property 183
- Zeldman, Jeffrey 17, 39, 44
- Zen of CSS Design* 197

Colophon:

I wrote and laid out this book entirely in Adobe FrameMaker 6 (on a G4 Mac running Classic!). Although I find Frame increasingly awkward, I could never have done all of the cross references, figure numbering, and especially the index without it.

I viewed the examples on all major browsers on both platforms. I used Parallels WorkStation to run Windows right in my MacBook. I took screen captures with Snapz Pro X and then cleaned them up with Adobe Photoshop CS. I used Adobe Illustrator to create the line drawings. The font faces in this book are various weights of Garamond and Futura.

Except for the ones in other people's Web sites (obviously), and the nice illustration in Figure 16.2 on page 228 by Andreu Cabré, the videos, photos, and drawings in this book are of my own creation, though I'm sometimes embarrassed to admit it. You can see more of my photographs at <http://www.flickr.com/photos/cookwood>