

Foundation

HTML5 with CSS3

A Modern Guide and Reference

Craig Cook
Jason Garber

friendsof 
an Apress® company

Foundation HTML5 with CSS3

Craig Cook
Jason Garber



Foundation HTML5 with CSS3

Copyright © 2012 by Craig Cook and Jason Garber

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

ISBN 978-1-4302-3876-8

ISBN 978-1-4302-3877-5 (eBook)

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logos, or image we use the names, logos, or images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com.

For information on translations, please e-mail rights@apress.com or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales—eBook Licensing web page at www.apress.com/bulk-sales.

Any source code or other supplementary materials referenced by the author in this text is available to readers at www.apress.com. For detailed information about how to locate your book's source code, go to www.apress.com/source-code.

Credits

President and Publisher: Paul Manning
Coordinating Editor: Jennifer Blackwell

Lead Editor: Ben Renow-Clarke
Copy Editor: Lori Cavanaugh

Technical Reviewer: Karen Tegtmeier
Compositor: Bytheway Publishing Services

Editorial Board: Steve Anglin, Ewan Buckingham, Gary Cornell,
Louise Corrigan, Morgan Ertel, Jonathan Gennick,
Jonathan Hassell, Robert Hutchinson, Michelle Lowman,
James Markham, Matthew Moodie,
Jeff Olson, Jeffrey Pepper, Douglas Pundick,
Ben Renow-Clarke, Dominic Shakeshaft,
Gwenan Spearing, Matt Wade, Tom Welsh
Indexer: SPi Global
Artist: SPi Global
Cover Designer: Anna Ishchenko

Contents at a Glance

About the Authors	xi
About the Technical Reviewer.....	xii
Acknowledgments	xiii
Introduction	xiv
Chapter 1: Getting Started	1
Chapter 2: HTML and CSS Basics	17
Chapter 3: The Document	37
Chapter 4: Constructing Content	69
Chapter 5: Embedding Media	139
Chapter 6: Linking the Web	185
Chapter 7: Building Tables	207
Chapter 8: Assembling Forms and Applications.....	239
Chapter 9: Page Layout with CSS	299
Chapter 10: Putting it All Together.....	331
Index.....	403

Contents

About the Authors	xi
About the Technical Reviewer	xii
Acknowledgments	xiii
Introduction	xiv
Chapter 1: Getting Started	1
Introducing the Internet and the World Wide Web	1
What Is HTML?	2
The Evolution of HTML	3
What's in HTML5?	6
Separating Content from Presentation	7
The Next Level of CSS	8
Working with HTML and CSS	9
Choosing an Editor	9
Choosing a Web Browser	10
Validating Your Documents	10
Hosting Your Web Site	12
Introducing the URL	12
The Components of a URL	12
Absolute and Relative URLs	14
Summary	14
Chapter 2: HTML and CSS Basics	17
The Parts of Markup: Tags, Elements, and Attributes	17
Attributes	19
Content Models	20
Nesting Elements	22
Global Attributes	22
White Space	25
Adding Comments	26
CSS Fundamentals	27
Anatomy of a CSS Rule	27
CSS Selectors	28
Combining Selectors	31
Grouping Selectors	31
Advanced Selectors	31
Specificity and the Cascade	32

Formatting CSS	34
Summary	36
Chapter 3: The Document	37
The Anatomy of an HTML Document.....	38
The Doctype	38
The Root Element: html.....	40
head.....	41
title	42
meta.....	43
link	46
base.....	50
style	52
script.....	54
body.....	57
The Document Tree	58
Connecting CSS.....	58
Inline Styling	59
Embedded Style Sheets.....	59
External Style Sheets	60
The Cascade Order	61
Style with Color.....	62
Specifying Colors in CSS	63
Summary	67
Chapter 4: Constructing Content	69
Content and Structure	69
Logical Sections	70
section	71
article	72
header	73
footer.....	74
aside	75
nav	77
Headings	78
h1, h2, h3, h4, h5, and h6	78
hgroup	80
Outlines and Sectioning Roots.....	82
Meaningful Portions.....	84
p.....	84
blockquote	85

address.....	86
pre.....	87
hr.....	88
div.....	89
figure.....	90
figcaption.....	92
iframe.....	94
Lists.....	97
ul.....	97
ol.....	98
li.....	100
dl.....	101
dt.....	102
dd.....	102
Phrasing Elements.....	103
Emphasis and Importance: em and strong.....	104
time.....	105
mark.....	106
cite.....	108
q.....	109
dfn.....	110
abbr.....	110
small.....	112
span.....	113
Programming: code, kbd, samp, and var.....	113
Revisions: del and ins.....	115
Offset Text: i, b, u, and s.....	116
Superscript and Subscript: sup and sub.....	118
Text Annotations: ruby, rp, and rt.....	118
Bidirectional Text: bdi and bdo.....	120
Line breaks: br and wbr.....	122
Special Characters.....	124
Styling Content with CSS.....	125
Establishing New Elements.....	126
Declaring Base Font Styles.....	127
Shorthand for Fonts.....	132
Heading Hierarchy.....	133
Styling Lists.....	136
Changing Unordered List Markers.....	136

Changing the Style of Ordered Lists	138
Summary	138
Chapter 5: Embedding Media	139
How Digital Images Work	140
Web-Friendly Image Formats	141
Embedding Images	145
img	145
Embedding Audio and Video	152
Media Codecs and Formats	152
audio	153
video	157
source	160
track	162
Embedding Plug-ins and Other Content	164
object	164
param	165
embed	166
Dynamic Drawings	167
canvas	167
Images with Style, Styling with Images	173
Wrapping Text Around an Image	173
Background Images	175
Positioning a Background Image	179
Shorthand for Backgrounds	182
Summary	183
Chapter 6: Linking the Web	185
Anchors	185
a	186
Inline vs. Block-level Links	191
Image Maps	192
map	192
area	193
Styling Anchors and Image Maps with CSS	196
Styling Anchors	196
Button-making	200
Styling Image Maps	203
Summary	203
Chapter 7: Building Tables	207
The Table's Lament	207

table	208
tr	209
td	210
th	213
Table Row Groups	216
thead	216
tbody	218
tfoot	219
Columns	222
colgroup	222
col	224
One more thing	225
caption	227
Styling Tables with CSS	227
Styling Rows and Cells	227
Styling Columns	231
Styling Captions	234
Summary	238
Chapter 8: Assembling Forms and Applications	239
How Forms Work	240
The form Element	240
The input Element	243
input type="text"	247
input type="search"	248
More Form Controls	263
select	263
option	266
optgroup	267
textarea	269
datalist	272
button	273
keygen	275
Structuring Forms	276
fieldset	276
legend	277
label	279
Displaying Output	281
output	282
meter	283

progress.....	286
Interactive Elements.....	287
details and summary	288
menu.....	289
command.....	290
Styling Forms with CSS.....	291
Removing the Border from fieldsets	293
Aligning Labels	294
Changing Fonts in Form Controls	295
Summary	297
Chapter 9: Page Layout with CSS	299
The Box Model	300
Margins and Padding.....	301
Borders	302
Page Width: Fixed vs. Liquid.....	304
Fixed Width.....	304
Liquid Width.....	306
Floating Columns	308
Clearing Floats	310
Positioning Elements.....	316
Relative Positioning	316
Absolute Positioning	319
Fixed Positioning	323
Summary	329
Chapter 10: Putting it All Together.....	331
A Web Design Process	332
Stage 1: Plan	332
Stage 2: Design	334
Stage 3: Build	337
Stage 4: Test	337
Building Power Outfitters.....	338
Setting Up the Document	339
Assembling the Template	341
Fleshing Out the Home Page	347
Designing Power Outfitters with CSS.....	351
Resetting Browser Defaults	353
Multiple Background Images	354
The Body Boundary.....	357
Styling the Masthead	357

Styling the Navigation.....	362
Laying Out the Content.....	365
Reusable Boxes	367
The New Item Box.....	370
Styling the Footer	374
Taking It Further	376
Vendor Prefixes	377
Box Shadows.....	378
Embedded Fonts	381
Text Shadows.....	386
Rounded Corners	388
Color Gradients	392
Media Queries	396
And Even Further.....	401
Summary	401
Index.....	403

About the Authors



Craig Cook has been designing and building websites since 1998, though he still silently harbors the aspiration to draw comic books. His background is in traditional graphic design, and although he spent years learning how to make ink stick to paper, he fell in love with the Web and the affair continues to this day. In addition to his passions for design and technology, Craig has an affinity for science-fiction novels, zombie movies, and black T-shirts. He occasionally muses on these subjects and others at his personal website, focalcurve.com. Craig lives and works near San Francisco.



Jason Garber is a web developer living and working near Washington, DC. He got his start building websites in 1996, long before turning a hobby into a career in 2003. Jason founded Refresh DC, a community of web designers, developers, and other new media professionals. He remains active in the community and continues to help organize monthly Refresh DC meetings. Jason also plays guitar in a DC post-rock band, The Orchid, and obsessively collects vinyl records. He sometimes writes on his personal site, sixtwothree.org.

About the Technical Reviewer



Karen Tegtmeyer is an independent consultant and software developer with more than 10 years of experience. She has worked in a variety of roles, including design, development, training, and architecture. She also is an Adjunct Computer Science Instructor at Des Moines Area Community College. Her website is www.KarenTegtmeyer.com

Acknowledgments

I must first thank all of the authors, artists, designers, coders, bloggers, evangelists, and gurus on whose shoulders I stand. I've been inspired and guided by the work and teachings of Dan Cederholm, Tantek Çelik, Joe Clark, Chris Coyier, Derek Featherstone, Mark Norman Francis, Aaron Gustafson, Christian Heilmann, Molly Holzschlag, Jeremy Keith, Bruce Lawson, Ethan Marcotte, Scott McCloud, Eric Meyer, Chris Mills, Dave Shea, Greg Storey, Jeffrey Zeldman... and many others equally deserving of a name-drop but I'm trying to keep this short.

I should also extend gratitude to all the daily-grinders, cubicle-dwellers, and impassioned community members who strive to build a better Web—not for riches and adoration but simply because they love what they do and care about doing it right. You're making the Web a better place to live. Take a bow.

Many kind thanks to my co-author, Jason, for jumping in feet first. What with the glacial pace of my writing, your two chapters easily shaved months off the schedule. Thanks also to everyone at Apress/Friends of ED for your patience with my aforementioned glacial pace throughout this entire process. It's done now.

Thank you Bill, Jannyce, and Jolene for your continued friendship and support, and big thanks to Stephanie Hobson, Cindy Li, and Matt Harris for being my super heroes.

I'm endlessly grateful for my parents, R.L. and Beverly, who instilled me with a desire to learn, a passion to create, and a compulsion to instruct.

Craig Cook

First and foremost, thanks to Craig for asking me to pitch in on the book and for your willingness to help a novice writer. Your guidance and feedback were incredibly helpful. Thank you as well to everyone at Apress for answering my many, many questions about the whole book-writing process.

Special thanks to Cindy Li and Kevin Lawver for your support and mentorship through the years. You've both opened so many doors for me and I don't think I could ever fully repay either of you.

Since 2006, the Refresh DC community has been a constant source of inspiration and knowledge. I absolutely believe that Washington, DC has the best web design and development community out there. Thank you to everyone who has presented, helped organize, and attended one of our meetings. You keep me professionally energized year-round.

Mom, Dad, Kim, Bayleigh, Heather, Phillip, Logan, Grandma, and Granny: You're the best family I've ever had. I love you all and promise to call and write more!

And lastly, thank you to Lauren for your patience and love.

Jason Garber

Introduction

The World Wide Web has come a long way in a relatively short time. Since its debut in the early 1990s, the Web has quickly developed from an esoteric collection of academic papers into a full-fledged and pervasive medium, an equal to print, radio, and television. It's a vast repository of information on every subject imaginable, from astrophysics and ancient history to the care and feeding of hermit crabs. The Web has become an integral part of many people's daily lives and is the platform for many aspects of modern business and commerce. But at its heart, the Web is still just a way to share documents.

This book will show you how to create your own documents so you can share them on the Web. You'll become intimately familiar with the rules and constructs of HyperText Markup Language (HTML), the computer language on which the Web is built. It's a simple language, and the basic rules are easy to pick up and put to use. HTML is a tool, and once you know how to use it, you're limited only by your imagination.

Like any other language, HTML has changed and evolved over time. New features are added, existing features are redefined, and outdated features are removed. The latest iteration of HTML is HTML5—ostensibly the fifth version of the language—and it's quite an evolutionary leap indeed. HTML5 incorporates a wide range of new features and introduces new types of functionality far beyond the humble text document.

We won't be going quite that far in this book. Instead of diving into all the shiny new advancements in Web technology, this book offers the true foundation of HTML5, the supporting structure beneath all that innovation. You'll learn how to use HTML effectively and responsibly, to build web pages with clean, meaningful code, and to make them accessible to as many people and devices as possible. In the chapters that follow we'll show you the heart of HTML5 and set the stage for your next leap into the Web's frontier. This book is a starting point, and there's no limit to where you can go from here.

Along the way, we'll also introduce Cascading Style Sheets (CSS). CSS is a language that describes the presentation of web documents, declaring what colors and fonts to use, and the size, shape, and placement of elements on the page. It's very powerful and flexible, and is also really cool. But CSS is a fairly complex language in its own right, and we can't possibly cover every facet of it in a single book.

However, as you'll soon see, CSS relates directly to HTML, and you'll first need to understand HTML before you can put CSS to good use. This book will show you the basics of CSS and offer practical examples of how you can use it. We'll give you the solid grounding in HTML you'll need as a platform to delve deeper into the art and craft of designing web pages with CSS.

The CSS language also continues to evolve. CSS3 is the next generation of CSS, bringing with it exciting new features and effects, offering new ways to design web pages that weren't possible before. We'll introduce just a few of the new advancements in CSS3 and show you how to apply them to your content.

Who This Book Is For

This book is for anyone interested in learning how to build web pages from the ground up using modern best practices. We assume you're familiar with the Internet and the World Wide Web, and you probably

wouldn't pick up a book with "HTML" in the title unless you'd at least heard of it. Beyond that, we don't assume any prior knowledge of web design or computer programming. As you advance through this book, the topics get a little more advanced as well. But fear not: this is a book for beginners, and we'll walk you through the tough parts.

Even if you're not a beginner, this book may still be well worth your time. Quite a bit has changed in HTML5, so if you're a more experienced web developer looking to get back to basics and see what all these new-fangled elements are about, this is the book for you. Although we've put the book together following a general narrative from beginning to end, it's also pretty handy as a reference book so you can turn to any section and find the information you need.

Technical books can be intimidating and hard to read, with a lot of prickly jargon to slog through. We've done our best to keep this book simple and easy to follow, and we define the technical terms as they appear. If you make it through this introduction and find yourself excited rather than bored, you should definitely keep reading.

How This Book Is Structured

The first two chapters lead you through the bare essentials you'll need to start making web pages. Throughout the bulk of this book, Chapters 3 through 8, you'll dig into different subject areas within HTML, becoming familiar with all of the different elements and attributes at your disposal. Along the way, you'll see examples of some of the many ways you can use CSS to style your content, and we've dedicated Chapter 9 to showing you just a few techniques to lay out pages with CSS. We finish up with Chapter 10, where you'll see a case study that takes much of what you've learned throughout the previous chapters and puts it together into a functional website, built from scratch with HTML and CSS.

Here's a brief road map of where this book is going to take you:

- Chapter 1, "Getting Started," takes a high-level view of how the Web works and what you'll need in order to create your own websites.
- Chapter 2, "HTML and CSS Basics," presents the fundamental rules to follow when you assemble web documents and style sheets, laying the groundwork for the rest of the book.
- Chapter 3, "The Document," explains the structure of an HTML document, establishing a framework that you can fill with content.
- Chapter 4, "Structuring Content," explores how you'll add content to your documents and support your text with a stable, meaningful structure.
- Chapter 5, "Embedding Media," describes how you can add pictures, audio, and video to your web pages for meaningful communication as well as decoration and entertainment.
- Chapter 6, "Linking the Web," looks at the all-important hyperlink, the very cornerstone of the Web. Hyperlinks are the vital connections between web pages.

- Chapter 7, “Constructing Tables,” shows you how to structure complex data in tables, organizing related information in sets of connected rows and columns.
- Chapter 8, “Building Forms and Applications,” will show you how to create forms that allow your visitors to input their own information and interact with your website.
- Chapter 9, “Page Layout with CSS,” dives deeper into the craft of using CSS to arrange content on the page, covering a few of the simple techniques you’ll use to design for the Web.
- Chapter 10, “Putting It All Together,” puts the topics discussed throughout the book to practical use, taking you step-by-step through the process of building a website from top to bottom.

Conventions

Throughout this book we’ll show you many examples of HTML and CSS coding. Most of these examples appear in numbered listings, separated from the regular text, like you see in Listing 1. In most cases we’ll only show you a short snippet of code that you can assume resides alongside other code in a complete document or style sheet.

Listing 1. An example of a code listing

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Document Title</title>
  </head>
  <body>
    <p>Hello, world!</p>
  </body>
</html>
```

Sometimes a line of code is too long to fit within the limited width of a printed page and we’re forced to wrap it to a second line. When that happens, we’ll use the symbol ↪ to let you know a line is wrapped only to fit the page layout; the real code would appear on a single line.

We’ll occasionally add notes, tips, and cautions that relate to the section you’ve just read. They appear distinct from the main text, like so:

Don’t overlook these extra tidbits. They’re relevant to the current topic and deserve some attention.

We may also sometimes wander off on a slight tangent that isn’t really part of the topic at hand but is still important information you should know. To keep things flowing smoothly, we’ll place such supplemental information in sidebars, which look like this:

SIDEBARS

Sidebars offer extra information, exploring a related topic in more depth without derailing the main topic. The term *sidebar* comes from magazine and newspaper publishing, where these sorts of accompanying stories are often printed in another column alongside the main article.

Downloading the code

All of the HTML and CSS examples you'll see in this book are available for download from the Apress website (apress.com). Once you've downloaded and unzipped the file, you'll find each chapter's source code in a separate folder; you can pick it apart and refer to it at your leisure. You can also find the source code and other information at this book's companion website, foundationhtml.com.



Chapter 1

Getting Started

We're sure you're champing at the bit to start building web pages, but we'd like to set the stage first and cover some general information about the Internet and World Wide Web, as well as some background on HTML and CSS. This chapter isn't a comprehensive overview by any means, but it will get you up to speed on some of the terminology and concepts you'll need to be familiar with throughout the rest of this book. If you're already pretty web-savvy, and if you've used and worked with websites for some time, feel free to skip ahead to Chapter 2 and start getting your hands dirty.

Introducing the Internet and the World Wide Web

"The Internet" is simply a catchall name for the vast, globe-spanning network of computers that are connected to each other and can transmit and receive data, shuttling information back and forth around the world at nearly the speed of light. It's been around in some form for nearly half a century now, ever since a few smart people figured out how to make one computer talk to another computer. The Internet has since become so ubiquitous and pervasive, impacting so many aspects of modern life, that it's hard to imagine a world without it.

The World Wide Web is one facet of the Internet, like a bustling neighborhood in a much larger city (other Internet "neighborhoods" include e-mail, news groups, and chat rooms). The Web is made up of millions of files and documents residing on different computers across the Internet, all interconnected to weave a web of information around the world, which is how it gets its name. In its relatively short history, the Web has grown and evolved far beyond the simple text documents it began with, carrying other types of information through the same channels: images, video, audio, and fully immersive interactive experiences. But at its

core, the Web is fundamentally a text-based medium, and that text is usually encoded in HTML (more on that in a minute).

Many different devices can access the Web: desktop and laptop computers, tablets and PDAs, mobile phones, game consoles, and even some household appliances. Whatever the device, it in turn operates software that interprets HTML. These programs are technically known as *user-agents*, but the more familiar term is *web browsers*. A web browser is specifically a program intended to visually render web documents, whereas some user-agents interpret HTML but don't display it.

In this book we'll generally use the word *browser* to mean any user-agent capable of handling and rendering HTML documents, and we may use the term *graphical browser* when we're specifically referring to one that renders the document in a visually enhanced format, in full color, and with styled text and images. It's important to make this distinction because some web browsers are *not* graphical and only render plain, unstyled text without any images.

A browser or user-agent is also known as a *client*, because it is the thing requesting and receiving service. The computer that serves data to the client is called, not surprisingly, a *server*. The Internet is riddled with servers, all storing and processing data and delivering it in response to client requests. The client and the server are two ends of the chain, connected to each other through the Internet.

What Is HTML?

The World Wide Web originated as a purely textual medium, built upon the written word. Pictures were soon added to the mix, and eventually sound, animation, and video made the Web the rich multimedia tapestry it is today. But the overwhelming bulk of Web content still takes the form of written text, and that's not likely to change any time soon. Most of the time you spend surfing the Web is probably spent reading.

The Web, for all its multimedia richness, is still essentially a textual medium. It's a weave of documents, cross-referenced and interconnected by the humble *hyperlink*, wherein a bit of text in one document is linked directly to another document somewhere else on the Web. And just like that, what would otherwise be ordinary text becomes the much more exciting and dynamic *hypertext*, and hypertext needs to be encoded in a whole new language: HyperText Markup Language (HTML).

HTML is the computer coding language that describes the structure of a web page. It converts ordinary text into active text for display and use on the Web, and also gives plain, unstructured text the sort of structure human beings rely on to read it. As you read this book, you're looking for visual cues to help you organize the words into smaller portions that you can process and comprehend. You recognize the significance of things like punctuation, capitalization, spacing, and font sizes. You know just by looking at it that this paragraph ends after this sentence.

Computers don't read text the same way humans do—they can't interpret a string of words and grasp the concept behind them, they don't see the visual cues we use to separate one group of words from another, and they can't automatically group related sentences into meaningful paragraphs. Instead of visual cues, a computer requires a structure composed of clear markers that designate the nature of each portion of text. That's the essence of a markup language: embedded instructions that a computer can follow in order to make content readable and usable by humans.

HTML consists of encoded markers called *tags* that surround and differentiate portions of text, indicating the function and purpose of the content those tags “mark up.” Tags are embedded directly in a plain-text document where they can be interpreted by a browser. They’re called *tags* because, well, that’s what they are. Just as a price tag displays the cost of an item and a toe tag identifies a cadaver, so too does an HTML tag indicate the nature of a portion of content and provide vital information about it. Listing 1-1 is a very simple bit of HTML, just a heading and a paragraph.

Listing 1-1. An example of text marked up with HTML. The tags are highlighted in bold.

```
<h1>This is a Level One Heading</h1>  
<p>This is a paragraph.</p>
```

A browser doesn’t display the tags themselves; tags only tell the browser how to treat the content between them. A matched pair of start and end tags (the end tag has a slash) forms an *element*, comprising the tags and everything in between them. You’ll learn a lot more about tags and elements in Chapter 2, and you’ll learn about the full range of HTML elements throughout the rest of this book.

From its inception, HTML has been carefully designed to be a simple and flexible language. It’s a free, open standard, not owned or controlled by any company or individual. There is no license to purchase or specialized software required to author your own HTML documents. Anyone can create and publish web pages, and it’s that very openness that makes the Web the powerful, far-reaching medium it is. HTML exists so that we can all share information freely and easily.

However, you do need to follow certain rules when you author documents in HTML—there are certain ways they should be assembled to make certain they’ll work properly. The Web runs on agreement, with all the different authors and programmers and clients and servers agreeing to abide by the same basic rules, collectively referred to as *web standards*. Standardizing web languages ensures that the Web can work consistently and reliably for everyone—users and authors alike. Sticking to the agreed-upon rules makes communication possible, like the rules of grammar and punctuation that help you understand this sentence.

Of course, it follows that someone needs to write down the rules to which we should agree. The technical specifications for many of the core languages (including HTML) that make up the Web are overseen and maintained by the World Wide Web Consortium (W3C), an international, non-profit organization founded in 1994 for just this purpose—to standardize the languages and map a clear path for the Web of the future.

You can learn more about the W3C and read all of their public specifications, past and present, at their website, w3.org. The specifications can be difficult to read because they’re extremely technical in nature, written primarily for computer scientists and software vendors who program web user-agents. But this kind of standardization is essential for the widespread adoption of the Web, ensuring that websites function properly across different browsers and operating systems. The Web is meant to be “platform independent” and “device independent,” and adherence to web standards makes that possible.

The Evolution of HTML

HTML first appeared in 1990—built upon the pre-existing Standard Generalized Markup Language (SGML)—as the foundational language for the newborn World Wide Web, but it wasn’t formally defined

until 1993. It was further refined and extended with HTML 2.0, the first official HTML standard, in 1995. Version 3.2 arrived in early 1997 with a slew of new features, and HTML 4.0 came shortly thereafter near the end of the same year.

In those early years of the Web, the language specifications weren't always followed as closely as they should have been. Different browsers supported different features of HTML, and introduced their own nonstandard features just to get a leg up on the competition. Given the unruly landscape of the time, authors didn't follow the standards any better than the browsers did. The early web was a tangle of bloated, convoluted markup and proprietary, browser-specific functionality. Developers often resorted to making multiple versions of their sites targeted to different browsers, or even worse, they built websites that worked properly in only one browser and failed utterly in others. Ask an old timer about the Browser Wars of the mid-90s and they'll regale you with frightening tales of forked scripts, nested tables, and pixel shims. Those were dark days indeed.

Thankfully, this is no longer the case. The web browsers of today follow the standardized specs much more consistently than in previous generations, encouraging authors to do the same, and thus advancing the Web toward the ultimate goal of a truly universal medium.

As the Web really took off in the late 1990s, a few minor (but significant) changes to HTML 4.0 were released in 1999 as HTML 4.01. After a decade of rapid innovation, HTML 4.01 was expected to be the last complete specification of the HTML language. A new kid called XHTML had joined the class, and it was praised as the wave of the future.

The Age of X

Around the turn of the century (way back in the year 2000), the W3C was convinced that the future of the Web lay in eXtensible Markup Language (XML), a powerful language that allows authors to create customized elements rather than relying only on the elements predefined by the language itself. Extensible HTML (XHTML) is a reformulation of HTML following the more stringent syntax of XML. It was meant to bridge the gap between HTML and XML, preparing web authors for this bright XML future everyone expected to arrive any day now.

Whereas XML is extensible, XHTML offers a finite set of predefined elements to choose from—all the same elements that were available in HTML 4.01, in fact. The only real differences between HTML 4.01 and XHTML 1.0 are stylistic, with just a few more rules dictating how XHTML must be written. HTML is a lax language designed to be tolerant of minor transgressions in syntax, whereas XML is fussy and demands strict adherence to its rules. XHTML simply applies the strictness of XML to HTML, resulting in a hardened set of rules for authoring a document. An XHTML document is essentially just an HTML document written to a more exacting standard.

It was also right around the time XHTML came on the scene that web designers and developers began a serious campaign to improve the state of the Web, encouraging their clients and colleagues to develop in accordance with web standards, and pressuring browser makers to correctly support those same standards in their products. XHTML, with its stricter rules of conformance, was the darling of the web standards movement because it encouraged authors to pay closer attention to how they constructed their documents.

The Web Standards Project (WaSP) was founded in 1998 in reaction to the inconsistent browser behaviors and unsustainable development practices of the era. This group led the charge in what became “the web standards movement,” promoting a new set of best practices for web designers and developers, ultimately changing the way web sites are made and improving the state of the web, for authors and users alike. WaSP continues to work with web authors, educators, browser vendors, and standards bodies to advance and promote web standards. Their website is webstandards.org.

Meanwhile, the W3C immediately began work on XHTML 2.0. No simple reformulation of existing standards, this was going to be a radical overhaul of the language from the ground up, a whole new approach to authoring documents for the Web. That was over a decade ago. The XHTML 2.0 specification stagnated and eventually stalled, while the Web continued to move inexorably forward, innovating on top of a foundation that was beginning to show its age. By the mid-2000s it became clear to some that XHTML 2.0 was perhaps not the best way forward after all, and it was time to re-examine and refresh good old HTML.

Out with the X, in with the 5

A splinter group formed within the W3C in 2004 and began to craft new addendums to HTML. They called themselves the Web Hypertext Application Technology Working Group (WHATWG, whatwg.org) and their side projects were dubbed Web Apps 1.0 and Web Forms 2.0, both meant to be extensions of the stale HTML 4.01 spec. Eventually these two projects were united in a new fledgling specification: HTML5.

In due time the W3C also came to accept that XHTML 2.0 wasn't working out as planned, and recognized that this new HTML5 business was something worth paying attention to. The W3C started the process of adopting and formalizing the work produced by WHATWG. And so HTML5 gained official status as the next HTML standard.

As all versions of HTML have done, HTML5 builds on what came before, always refining and extending and improving. In fact, HTML5 is still taking shape as we write this in the summer of 2011, though they're aiming for the spec to be completed in 2012. But, although the specification is incomplete at the moment, it's relatively stable at the time of this writing (knock on wood) and there's nothing preventing you from using the fundamentals of HTML5 on the Web today.

Two groups—WHATWG and the W3C—are working on HTML5 in tandem. Although the specification is still taking shape, you can read the work in progress at their respective websites: WHATWG's version is at whatwg.org/html and the W3C's is at w3.org/TR/html5/. Depending on when each was last updated, there may be some differences between the two versions of the spec, and both are works in progress and subject to change. Generally speaking, the WHATWG version includes the very latest changes, and the W3C version is a bit more refined and finalized.

One of the tenets of HTML5 is to maintain backward compatibility (something XHTML 2 would have broken); existing content must continue to function under HTML5. In that sense, any document marked up

in any version or variant of HTML is already an HTML5 document, and any browser that interprets HTML already supports most of HTML5. What really matters is browser support for the few specific features that are brand new.

HTML5 introduces a number of new tags and attributes that didn't exist in any prior HTML version. Current versions of most popular browsers already support many of these new features, whereas some other advanced features aren't fully developed and aren't yet supported by browsers, but that tide is changing at a breakneck pace. All the major browser makers—Mozilla, Microsoft, Apple, Google, and Opera—are releasing frequent updates to their browsers, improving support for the finer points of HTML5 with each new version.

What's in HTML5?

As often happens with any advance in technology, "HTML5" was quickly seized upon as a buzzword to make things sound bleeding edge and cool, even if what was being discussed wasn't part of HTML5 at all. A broad range of technologies and techniques were soon lumped together under the banner of "HTML5," leading to a great deal of confusion about just what was and wasn't, in actuality, HTML5.

HTML5 is simply the next iteration of HTML, the language that gives web content its necessary structure. As you read earlier in this chapter, HTML tags form structural elements in a document, allowing readers (and programs) to differentiate a headline from a paragraph, or a paragraph from a list, or a list from a quotation, and so on. Content without structure is content without meaning. This latest version of HTML introduces a number of new, meaningful elements that were lacking in HTML 4 and XHTML. In addition to the usual headings, paragraphs, tables, and lists, there are new elements for things like navigation, menus, articles, summaries, dates and times, figures with captions, and a heap of new interactive form elements. All the useful elements from previous versions of HTML have been kept, but HTML5 eliminates some legacy elements that have outlived their usefulness. You'll learn all about the elements of HTML5, both old and new, in detail throughout the rest of this book.

Also new in HTML5 are elements for embedding rich media in documents. Images have been on the Web almost from the beginning, but for years authors had to rely on third-party plug-in applications—such as Adobe's Flash or Apple's QuickTime—to play sound and video over the Web. HTML5 makes it possible to play sound and video natively in the browser, without plug-ins. HTML5 also brings the `canvas` element, an area in a document where scripts and programs can draw live graphics. You'll learn more about embedding media in Chapter 5.

After all our "the Web is made of documents" talk, we shouldn't gloss over the prevalence of *web applications*. A web application might be similar to other computer applications you're familiar with—like an email program, a word processor, or the spreadsheet shown in Figure 1-1—but it works directly in a web browser. Under the surface, a great many web apps are actually nothing more than enhanced documents, using sophisticated code to manipulate HTML right before your eyes, yet still built on that same HTML foundation. HTML5 is being written with web apps in mind, offering new abilities and frameworks to enhance the applications built on top of it.

- [download online At Home in Stone Creek](#)
- [read A Reunion of Ghosts: A Novel pdf, azw \(kindle\), epub, doc, mobi](#)
- [download The Seed Underground: A Growing Revolution to Save Food here](#)
- [*Broca's Brain: Reflections on the Romance of Science here*](#)

- <http://ramazotti.ru/library/At-Home-in-Stone-Creek.pdf>
- <http://transtrade.cz/?ebooks/Autodesk-3ds-Max-2016-Essentials.pdf>
- <http://damianfoster.com/books/The-Seed-Underground--A-Growing-Revolution-to-Save-Food.pdf>
- <http://cambridgebrass.com/?freebooks/Broca-s-Brain--Reflections-on-the-Romance-of-Science.pdf>